Cadabra

Generated by Doxygen 1.8.11

Contents

1 Cadabra **1**

2 Module Index **3**

2.1 Modules 3

3 Namespace Index **5**

3.1 Namespace List 5

4 Hierarchical Index **7**

4.1 Class Hierarchy 7

5 Class Index **13**

5.1 Class List 13

6 File Index **19**

6.1 File List 19

7 Module Documentation **25**

7.1 Core 25

7.1.1 Module Description 25

7.1.2 Function Documentation 26

7.1.2.1 do_list(const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f) 26

7.1.2.2 do_subtree(const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f) 27

7.1.2.3 find_in_list(const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f) 27

7.1.2.4 find_in_subtree(const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f, bool including_head=true) 27

7.1.2.5 is_in(const T &val, const std::initializer_list< T > &list) 27

7.1.2.6 make_list(Ex el) . . . 27
7.1.2.7 multiply(rset_t::iterator &, multiplier_t) . . . 27
7.2 Cleanup . . . 28
7.2.1 Module Description . . . 28
7.3 Expression comparison . . . 29
7.3.1 Module Description . . . 29
7.3.2 Function Documentation . . . 29
7.3.2.1 subtree_compare(const Properties *, Ex::iterator one, Ex::iterator two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2, bool literal_↩ wildcards=false) . . . 29
7.4 Python Core . . . 31
7.4.1 Module Description . . . 31
7.4.2 Function Documentation . . . 31
7.4.2.1 __eq__Ex_Ex(const cadabra::Ex &, const cadabra::Ex &) . . . 31
7.4.2.2 __eq__Ex_int(const cadabra::Ex &, int) . . . 32
7.4.2.3 call_post_process(cadabra::Kernel &, cadabra::Ex &ex) . . . 32
7.4.2.4 create_scope() . . . 32
7.4.2.5 Ex_latex_(const cadabra::Ex &) . . . 32
7.4.2.6 Ex_str_(const cadabra::Ex &) . . . 33
7.4.2.7 get_kernel_from_scope() . . . 33
7.4.2.8 inject_defaults(cadabra::Kernel *) . . . 33
7.4.2.9 operator+(const cadabra::Ex &ex1, const cadabra::Ex &ex2) . . . 33
7.4.2.10 operator-(const cadabra::Ex &ex1, const cadabra::Ex &ex2) . . . 33
7.4.2.11 pull_in(std::shared_ptr< cadabra::Ex > ex) . . . 33
7.5 Scalar algebra . . . 34
7.5.1 Module Description . . . 34
7.5.2 Function Documentation . . . 34
7.5.2.1 apply(const cadabra::Kernel &, cadabra::Ex &, cadabra::Ex::iterator &, const std::vector< std::string > &wrap, const std::string &args, const std::string &method) 34
7.5.2.2 Ex_to_Sympy(const cadabra::Ex &) . . . 34
7.5.2.3 invert_matrix(const cadabra::Kernel &, cadabra::Ex &ex, cadabra::Ex &rules) . . . 34

7.5.2.4 map_sympy(const cadabra::Kernel &, cadabra::Ex &, const std::vector< std::string > &wrap, const std::string &args, const std::string &method) . . . 34

7.6 Algorithms . . . 35

7.6.1 Module Description . . . 35

7.7 Properties . . . 36

7.7.1 Module Description . . . 36

7.8 Client-Server . . . 37

7.8.1 Module Description . . . 37

7.9 Front-end . . . 38

7.9.1 Module Description . . . 38

7.10 Gtk+ . . . 39

7.10.1 Module Description . . . 39

7.11 OS-X . . . 40

7.11.1 Module Description . . . 40

8 Namespace Documentation **41**

8.1 cadabra Namespace Reference . . . 41

8.1.1 Detailed Description . . . 48

8.1.2 Typedef Documentation . . . 49

8.1.2.1 dispatcher_t . . . 49

8.1.2.2 DTree . . . 49

8.1.2.3 hashval_t . . . 49

8.1.2.4 multiplier_t . . . 49

8.1.2.5 nset_t . . . 49

8.1.2.6 rset_t . . . 49

8.1.3 Function Documentation . . . 49

8.1.3.1 add(rset_t::iterator &num, multiplier_t fac) . . . 49

8.1.3.2 cdb2python(const std::string &) . . . 49

8.1.3.3 check_index_consistency(const Kernel &k, Ex &tr, Ex::iterator it) . . . 49

8.1.3.4 cleanup_components(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 50

8.1.3.5 cleanup_derivative(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 50

8.1.3.6 cleanup_diagonal(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 50
8.1.3.7 cleanup_dispatch(const Kernel &k, Ex &, Ex::iterator &it) . . . 50
8.1.3.8 cleanup_dispatch_deep(const Kernel &k, Ex &, dispatcher_t disp=&cleanup_↩ dispatch) . . . 50
8.1.3.9 cleanup_dispatch_deep(const Kernel &k, Ex &tr, Ex::iterator &top, dispatcher_t dispatch) . . . 50
8.1.3.10 cleanup_exterior_derivative(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 50
8.1.3.11 cleanup_frac(const Kernel &k, Ex &tr, Ex::iterator &st) . . . 50
8.1.3.12 cleanup_indexbracket(const Kernel &k, Ex &, Ex::iterator &it) . . . 50
8.1.3.13 cleanup_kronecker(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 50
8.1.3.14 cleanup_numericalflat(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 50
8.1.3.15 cleanup_partialderivative(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 50
8.1.3.16 cleanup_productlike(const Kernel &k, Ex &, Ex::iterator &it) . . . 50
8.1.3.17 cleanup_rational(const Kernel &k, Ex &tr, Ex::iterator &st) . . . 51
8.1.3.18 cleanup_sub(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 51
8.1.3.19 cleanup_sumlike(const Kernel &k, Ex &tr, Ex::iterator &it) . . . 51
8.1.3.20 cleanup_updown(const Kernel &k, Ex &, Ex::iterator &it) . . . 51
8.1.3.21 convert_line(const std::string &, std::string &lhs, std::string &rhs, std::string &indent) 51
8.1.3.22 escape_quotes(const std::string &) . . . 51
8.1.3.23 export_as_HTML(const DTree &doc, bool for_embedding=false, std::string title="""") 51
8.1.3.24 export_as_LaTeX(const DTree &doc) . . . 51
8.1.3.25 export_as_python(const DTree &doc) . . . 51
8.1.3.26 flip_sign(rset_t::iterator &num) . . . 51
8.1.3.27 half(rset_t::iterator &num) . . . 51
8.1.3.28 HTML_recurse(const DTree &doc, DTree::iterator it, std::ostringstream &str, const std::string &preamble_string, bool for_embedding=false, std::string title="""") . . . 51
8.1.3.29 JSON_deserialise(const std::string &, DTree &) . . . 51
8.1.3.30 JSON_in_recurse(DTree &doc, DTree::iterator loc, const Json::Value &cells) . . . 52
8.1.3.31 JSON_recurse(const DTree &, DTree::iterator, Json::Value &) . . . 52
8.1.3.32 JSON_serialise(const DTree &) . . . 52
8.1.3.33 LaTeX_recurse(const DTree &doc, DTree::iterator it, std::ostringstream &str, const std::string &preamble_string) . . . 52

8.1.3.34 latex_to_html(const std::string &) . . . 52
8.1.3.35 list_size(const Ex &tr, Ex::iterator it) . . . 52
8.1.3.36 one(rset_t::iterator &num) . . . 52
8.1.3.37 operator<(const exchange::tensor_type_t &one, const exchange::tensor_type_t &two) . . . 52
8.1.3.38 pre_clean_dispatch(const Kernel &kernel, Ex &ex, Ex::iterator &it) . . . 52
8.1.3.39 pre_clean_dispatch_deep(const Kernel &k, Ex &tr) . . . 52
8.1.3.40 push_down_multiplier(const Kernel &k, Ex &tr, Ex::iterator it) . . . 52
8.1.3.41 python_recurse(const DTree &doc, DTree::iterator it, std::ostringstream &str) . . . 53
8.1.3.42 subtree_equal(const Properties *properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier) . . . 53
8.1.3.43 subtree_exact_equal(const Properties *properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards) . . . 53
8.1.3.44 subtree_exact_less(const Properties *properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards) . . . 53
8.1.3.45 subtree_less(const Properties *properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier) . . . 53
8.1.3.46 to_long(multiplier_t mul) . . . 53
8.1.3.47 to_string(long num) . . . 53
8.1.3.48 tree_equal(const Properties *properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier) . . . 53
8.1.3.49 tree_exact_equal(const Properties *properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards) . . . 53
8.1.3.50 tree_exact_less(const Properties *properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards) . . . 53
8.1.3.51 tree_less(const Properties *, const Ex &one, const Ex &two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2) . . . 53
8.1.3.52 unichar(kunichar c) . . . 53
8.1.3.53 zero(rset_t::iterator &num) . . . 53
8.1.4 Variable Documentation . . . 53
8.1.4.1 name_set . . . 53
8.1.4.2 rat_set . . . 53
8.2 cadabra2_defaults Namespace Reference . . . 53
8.2.1 Function Documentation . . . 54

8.2.1.1 _displayhook(arg) . . . 54
8.2.1.2 display(obj, delay_send=False) . . . 54
8.2.1.3 post_process(__cdbkernel__, ex) . . . 54
8.2.2 Variable Documentation . . . 54
8.2.2.1 __cdbkernel__ . . . 54
8.2.2.2 displayhook . . . 54
8.2.2.3 have_matplotlib . . . 54
8.2.2.4 mclose . . . 54
8.2.2.5 mopen . . . 54
8.2.2.6 remember_display_hook . . . 54
8.2.2.7 server . . . 54
8.2.2.8 sympy . . . 54
8.3 cadabra::symbols Namespace Reference . . . 54
8.3.1 Variable Documentation . . . 55
8.3.1.1 greek . . . 55
8.4 cadabra_osx Namespace Reference . . . 55
8.5 combin Namespace Reference . . . 55
8.5.1 Typedef Documentation . . . 56
8.5.1.1 range_t . . . 56
8.5.1.2 range_vector_t . . . 56
8.5.1.3 weights_t . . . 56
8.5.2 Function Documentation . . . 56
8.5.2.1 determine_intersection_ranges(const range_vector_t &prod, const range_↩ vector_t &indv, range_vector_t &target) . . . 56
8.5.2.2 fact(T x) . . . 56
8.5.2.3 factorial(unsigned int x) . . . 56
8.5.2.4 hash(const std::vector< unsigned int > &) . . . 56
8.5.2.5 operator<<(std::ostream &str, const symmetriser< T > &sym) . . . 56
8.5.2.6 operator==(const std::vector< unsigned int > &, const std::vector< unsigned int > &) . . . 56
8.5.2.7 ordersign(iterator1 b1, iterator1 e1, iterator2 b2, iterator2 e2, int stepsize=1) . . . 56

8.5.2.8 ordersign(iterator1 b1, iterator1 e1) . . . 56
8.5.2.9 vector_prod(const std::vector< unsigned int > &) . . . 56
8.5.2.10 vector_prod_fact(const std::vector< unsigned int > &) . . . 56
8.5.2.11 vector_sum(const std::vector< int > &) . . . 56
8.6 echokernel Namespace Reference . . . 57
8.6.1 Variable Documentation . . . 57
8.6.1.1 kernel_class . . . 57
8.7 snoop Namespace Reference . . . 57
8.7.1 Variable Documentation . . . 57
8.7.1.1 email . . . 57
8.7.1.2 error . . . 57
8.7.1.3 fatal . . . 57
8.7.1.4 flush . . . 57
8.7.1.5 info . . . 57
8.7.1.6 log . . . 58
8.7.1.7 warn . . . 58
8.8 sympy Namespace Reference . . . 58
8.9 yngtab Namespace Reference . . . 58
8.9.1 Detailed Description . . . 59
8.9.2 Typedef Documentation . . . 59
8.9.2.1 keeptrack_tab_t . . . 59
8.9.3 Function Documentation . . . 59
8.9.3.1 add_box(tableau &tab1, unsigned int row1, const tableau &tab2, unsigned int row2, unsigned int col2) . . . 59
8.9.3.2 add_box(T1 &tab1, unsigned int row1, const T2 &tab2, unsigned int row2, unsigned int col2) . . . 59
8.9.3.3 add_box(T1 &tab1, unsigned int row1, const tableau &tab2, unsigned int row2, unsigned int col2) . . . 59
8.9.3.4 legal_box(const std::vector< std::pair< int, int > > &prev, const std::vector< std::pair< int, int > > &ths, int colpos, int trypos) . . . 59
8.9.3.5 LR_add_box(const Tab &tab2, Tab &newtab, unsigned int currow2, unsigned int curcol2, unsigned int startrow, unsigned int maxrows, OutputIterator outit, keeptrack_tab_t &Ycurrent, bool alltabs) . . . 59
8.9.3.6 LR_tensor(const tableaux< T > &tabs1, const T &tab2, unsigned int maxrows, OutputIterator out, bool alltabs=false) . . . 59
8.9.3.7 LR_tensor(const Tab &tab1, const Tab &tab2, unsigned int maxrows, OutputIterator outit, bool alltabs=false) . . . 59
8.9.3.8 LR_tensor(const tableaux< T > &, bool symmetric, unsigned int maxrows, OutputIterator outit) . . . 59
8.9.3.9 operator<<(std::ostream &str, const tableau &tab) . . . 59
8.9.3.10 operator<<(std::ostream &, const tableaux< T > &) . . . 59
8.9.3.11 operator<<(std::ostream &, const filled_tableau< T > &) . . . 59

9 Class Documentation **61**

9.1 cadabra::Accent Class Reference . . . 61

9.1.1 Description . . . 61

9.1.2 Member Function Documentation . . . 61

9.1.2.1 name() const . . . 61

9.2 preprocessor::accu_t Class Reference . . . 62

9.2.1 Constructor & Destructor Documentation . . . 62

9.2.1.1 accu_t() . . . 62

9.2.2 Member Function Documentation . . . 62

9.2.2.1 erase() . . . 62

9.2.3 Member Data Documentation . . . 62

9.2.3.1 accu . . . 62

9.2.3.2 bracket . . . 62

9.2.3.3 head_is_generated . . . 62

9.2.3.4 is_index . . . 62

9.2.3.5 order . . . 62

9.2.3.6 parts . . . 62

9.3 cadabra::ActionAddCell Class Reference . . . 63

9.3.1 Description . . . 63

9.3.2 Member Enumeration Documentation . . . 63

9.3.2.1 Position . . . 63

9.3.3 Constructor & Destructor Documentation . . . 64

9.3.3.1 ActionAddCell(DataCell, DTree::iterator ref_, Position pos_) . . . 64

9.3.4 Member Function Documentation . . . 64

9.3.4.1 execute(DocumentThread &, GUIBase &) override . . . 64

9.3.4.2 revert(DocumentThread &, GUIBase &) override . . . 64

9.3.5 Member Data Documentation . . . 64

9.3.5.1 child_num . . . 64

9.3.5.2 newcell . . . 64

9.3.5.3 newref . . . 64

9.3.5.4 pos . . . 64
9.3.5.5 ref . . . 64
9.4 cadabra::ActionBase Class Reference . . . 65
9.4.1 Description . . . 65
9.4.2 Member Function Documentation . . . 65
9.4.2.1 execute(DocumentThread &, GUIBase &)=0 . . . 65
9.4.2.2 revert(DocumentThread &, GUIBase &)=0 . . . 65
9.4.2.3 undoable() const . . . 66
9.5 cadabra::ActionEraseText Class Reference . . . 66
9.5.1 Description . . . 66
9.5.2 Constructor & Destructor Documentation . . . 67
9.5.2.1 ActionEraseText(DTree::iterator, int, int) . . . 67
9.5.3 Member Function Documentation . . . 67
9.5.3.1 execute(DocumentThread &, GUIBase &) override . . . 67
9.5.3.2 revert(DocumentThread &, GUIBase &) override . . . 67
9.5.4 Member Data Documentation . . . 67
9.5.4.1 from_pos . . . 67
9.5.4.2 removed_text . . . 67
9.5.4.3 this_cell . . . 67
9.5.4.4 to_pos . . . 67
9.6 cadabra::ActionInsertText Class Reference . . . 67
9.6.1 Description . . . 67
9.6.2 Constructor & Destructor Documentation . . . 68
9.6.2.1 ActionInsertText(DTree::iterator, int pos, const std::string &) . . . 68
9.6.3 Member Function Documentation . . . 68
9.6.3.1 execute(DocumentThread &, GUIBase &) override . . . 68
9.6.3.2 revert(DocumentThread &, GUIBase &) override . . . 68
9.6.4 Member Data Documentation . . . 68
9.6.4.1 insert_pos . . . 68
9.6.4.2 text . . . 68

9.6.4.3 this_cell . . . 68
9.7 cadabra::ActionPositionCursor Class Reference . . . 69
9.7.1 Description . . . 69
9.7.2 Member Enumeration Documentation . . . 69
9.7.2.1 Position . . . 69
9.7.3 Constructor & Destructor Documentation . . . 70
9.7.3.1 ActionPositionCursor(DTree::iterator ref_, Position pos_) . . . 70
9.7.4 Member Function Documentation . . . 70
9.7.4.1 execute(DocumentThread &, GUIBase &) override . . . 70
9.7.4.2 revert(DocumentThread &, GUIBase &) override . . . 70
9.7.5 Member Data Documentation . . . 70
9.7.5.1 needed_new_cell . . . 70
9.7.5.2 newref . . . 70
9.7.5.3 pos . . . 70
9.7.5.4 ref . . . 70
9.8 cadabra::ActionRemoveCell Class Reference . . . 70
9.8.1 Description . . . 70
9.8.2 Constructor & Destructor Documentation . . . 71
9.8.2.1 ActionRemoveCell(DTree::iterator ref_) . . . 71
9.8.2.2 ∼ActionRemoveCell() . . . 71
9.8.3 Member Function Documentation . . . 71
9.8.3.1 execute(DocumentThread &, GUIBase &) override . . . 71
9.8.3.2 revert(DocumentThread &, GUIBase &) override . . . 71
9.8.4 Member Data Documentation . . . 71
9.8.4.1 reference_child_index . . . 71
9.8.4.2 reference_parent_cell . . . 71
9.8.4.3 removed_tree . . . 71
9.8.4.4 this_cell . . . 71
9.9 cadabra::ActionSetRunStatus Class Reference . . . 72
9.9.1 Description . . . 72

9.9.2 Constructor & Destructor Documentation . . . 72
9.9.2.1 ActionSetRunStatus(DTree::iterator ref_, bool running) . . . 72
9.9.3 Member Function Documentation . . . 72
9.9.3.1 execute(DocumentThread &, GUIBase &) override . . . 72
9.9.3.2 revert(DocumentThread &, GUIBase &) override . . . 73
9.9.3.3 undoable() const override . . . 73
9.9.4 Member Data Documentation . . . 73
9.9.4.1 new_running_ . . . 73
9.9.4.2 this_cell . . . 73
9.9.4.3 was_running_ . . . 73
9.10 cadabra::ActionSplitCell Class Reference . . . 73
9.10.1 Description . . . 73
9.10.2 Constructor & Destructor Documentation . . . 74
9.10.2.1 ActionSplitCell(DTree::iterator ref_) . . . 74
9.10.2.2 ∼ActionSplitCell() . . . 74
9.10.3 Member Function Documentation . . . 74
9.10.3.1 execute(DocumentThread &, GUIBase &) override . . . 74
9.10.3.2 revert(DocumentThread &, GUIBase &) override . . . 74
9.10.4 Member Data Documentation . . . 74
9.10.4.1 newref . . . 74
9.10.4.2 this_cell . . . 74
9.11 cadabra::Algorithm Class Reference . . . 75
9.11.1 Description . . . 75
9.11.2 Member Typedef Documentation . . . 78
9.11.2.1 index_map_t . . . 78
9.11.2.2 index_position_map_t . . . 78
9.11.2.3 iterator . . . 79
9.11.2.4 post_order_iterator . . . 79
9.11.2.5 range_t . . . 79
9.11.2.6 range_vector_t . . . 79

9.11.2.7 result_t . . . 79
9.11.2.8 sibling_iterator . . . 79
9.11.3 Constructor & Destructor Documentation . . . 79
9.11.3.1 Algorithm(const Kernel &, Ex &) . . . 79
9.11.3.2 ∼Algorithm() . . . 79
9.11.4 Member Function Documentation . . . 79
9.11.4.1 apply(iterator &)=0 . . . 79
9.11.4.2 apply_deep(Ex::iterator &it) . . . 79
9.11.4.3 apply_generic(bool deep=true, bool repeat=false, unsigned int depth=0) . . . 79
9.11.4.4 apply_generic(iterator &, bool deep, bool repeat, unsigned int depth) . . . 80
9.11.4.5 apply_once(Ex::iterator &it) . . . 80
9.11.4.6 apply_pre_order(bool repeat=false) . . . 80
9.11.4.7 begin_index(iterator it) const . . . 80
9.11.4.8 can_apply(iterator)=0 . . . 80
9.11.4.9 check_consistency(iterator) const . . . 80
9.11.4.10 check_degree_consistency(iterator) const . . . 80
9.11.4.11 check_index_consistency(iterator) const . . . 80
9.11.4.12 classify_add_index(iterator it, index_map_t &ind_free, index_map_t &ind_↩ dummy) const . . . 80
9.11.4.13 classify_indices(iterator, index_map_t &ind_free, index_map_t &ind_dummy) const . . . 80
9.11.4.14 classify_indices_up(iterator, index_map_t &ind_free, index_map_t &ind_dummy) const . . . 80
9.11.4.15 compare_(const str_node &, const str_node &) . . . 80
9.11.4.16 contains(sibling_iterator from, sibling_iterator to, sibling_iterator arg) . . . 81
9.11.4.17 determine_intersection(index_map_t &one, index_map_t &two, index_map_↩ t &target, bool move_out=false) const . . . 81
9.11.4.18 dumpmap(std::ostream &, const index_map_t &) const . . . 81
9.11.4.19 end_index(iterator it) const . . . 81
9.11.4.20 equal_without_numbers(nset_t::iterator, nset_t::iterator) . . . 81
9.11.4.21 fill_index_position_map(iterator, const index_map_t &, index_position_map_t &) const . . . 81
9.11.4.22 fill_map(index_map_t &, sibling_iterator, sibling_iterator) const . . . 81

9.11.4.23 find_arg_superset(range_vector_t &, Iter st, Iter nd) . . . 81

9.11.4.24 find_arg_superset(range_vector_t &, sibling_iterator it) . . . 81

9.11.4.25 find_argument_lists(range_vector_t &, bool only_comma_lists=true) const . . . 81

9.11.4.26 find_modulo_parent_rel(iterator it, index_map_t &imap) const . . . 81

9.11.4.27 force_prod_wrap(iterator &) . . . 81

9.11.4.28 get_dummy(const list_property *, const index_map_t *m1, const index_map_t *m2=0, const index_map_t *m3=0, const index_map_t *m4=0, const index_map_t *m5=0) const . . . 82

9.11.4.29 get_dummy(const list_property *, iterator) const . . . 82

9.11.4.30 get_dummy(const list_property *, iterator, iterator) const . . . 82

9.11.4.31 index_in_set(Ex, const index_map_t *) const . . . 82

9.11.4.32 index_parity(iterator) const . . . 82

9.11.4.33 intersection_number(sibling_iterator, sibling_iterator, sibling_iterator, sibling_iterator, BinaryPredicate) const . . . 82

9.11.4.34 is_factorlike(iterator) . . . 82

9.11.4.35 is_nonprod_factor_in_prod(iterator) . . . 82

9.11.4.36 is_single_term(iterator) . . . 82

9.11.4.37 is_termlike(iterator) . . . 82

9.11.4.38 less_without_numbers(nset_t::iterator, nset_t::iterator) . . . 82

9.11.4.39 locate_object_set(const Ex &objs, Ex::iterator st, Ex::iterator nd, std::vector< unsigned int > &store) . . . 82

9.11.4.40 locate_single_object(Ex::iterator obj_to_find, Ex::iterator st, Ex::iterator nd, std::vector< unsigned int > &store) . . . 82

9.11.4.41 max_numbered_name(const std::string &, const index_map_t *m1, const index_map_t *m2=0, const index_map_t *m3=0, const index_map_t *m4=0, const index_map_t *m5=0) const . . . 82

9.11.4.42 max_numbered_name_one(const std::string &nm, const index_map_t *one) const . . . 82

9.11.4.43 node_integer(iterator, int) . . . 83

9.11.4.44 node_one(iterator) . . . 83

9.11.4.45 node_zero(iterator) . . . 83

9.11.4.46 number_of_direct_indices(iterator it) . . . 83

9.11.4.47 number_of_indices(iterator it) . . . 83

9.11.4.48 number_of_indices(const Properties &, iterator it) . . . 83

9.11.4.49 print_classify_indices(std::ostream &, iterator) const . . . 83
9.11.4.50 prod_unwrap_single_term(iterator &) . . . 83
9.11.4.51 prod_wrap_single_term(iterator &) . . . 83
9.11.4.52 propagate_zeroes(post_order_iterator &, const iterator &) . . . 83
9.11.4.53 pushup_multiplier(iterator) . . . 83
9.11.4.54 rename_replacement_dummies(iterator, bool still_inside_algo=false) . . . 83
9.11.4.55 report_progress(const std::string &, int todo, int done, int count=2) . . . 83
9.11.4.56 separated_by_derivative(iterator, iterator, iterator check_dependence) const . . . 83
9.11.4.57 set_progress_monitor(ProgressMonitor *) . . . 83
9.11.5 Member Data Documentation . . . 84
9.11.5.1 discard_command_node . . . 84
9.11.5.2 get_dummy_sw . . . 84
9.11.5.3 index_sw . . . 84
9.11.5.4 interrupted . . . 84
9.11.5.5 kernel . . . 84
9.11.5.6 number_of_calls . . . 84
9.11.5.7 number_of_modifications . . . 84
9.11.5.8 pm . . . 84
9.11.5.9 report_progress_stopwatch . . . 84
9.11.5.10 suppress_normal_output . . . 84
9.11.5.11 tr . . . 84
9.12 cadabra::AntiCommuting Class Reference . . . 84
9.12.1 Member Function Documentation . . . 85
9.12.1.1 name() const . . . 85
9.12.1.2 sign() const . . . 85
9.13 cadabra::AntiSymmetric Class Reference . . . 85
9.13.1 Constructor & Destructor Documentation . . . 86
9.13.1.1 ~AntiSymmetric() . . . 86
9.13.2 Member Function Documentation . . . 86
9.13.2.1 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const override . . . 86

9.13.2.2 name() const override . . . 86
9.13.2.3 size(const Properties &, Ex &, Ex::iterator) const override . . . 86
9.14 AppDelegate Class Reference . . . 86
9.15 AppDelegate() Category Reference . . . 87
9.15.1 Property Documentation . . . 87
9.15.1.1 notebookController . . . 87
9.15.1.2 window . . . 87
9.16 snoop::Snoop::AppEntry Class Reference . . . 87
9.16.1 Description . . . 87
9.16.2 Constructor & Destructor Documentation . . . 88
9.16.2.1 AppEntry() . . . 88
9.16.2.2 AppEntry(const std::string &uuid_, uint64_t create_millis_, uint64_t receive_millis_, pid_t pid_, const std::string &ip_address_, const std::string &machine_id_, const std::string &app_name_, const std::string &app_version_, const std::string &user_id_, int server_status_) . . . 88
9.16.3 Member Function Documentation . . . 88
9.16.3.1 from_json(const Json::Value &) . . . 88
9.16.3.2 to_json(bool human_readable) const . . . 88
9.16.4 Member Data Documentation . . . 88
9.16.4.1 app_name . . . 88
9.16.4.2 app_version . . . 88
9.16.4.3 connected . . . 88
9.16.4.4 create_millis . . . 88
9.16.4.5 id . . . 88
9.16.4.6 ip_address . . . 88
9.16.4.7 machine_id . . . 88
9.16.4.8 pid . . . 88
9.16.4.9 receive_millis . . . 88
9.16.4.10 server_status . . . 88
9.16.4.11 user_id . . . 88
9.16.4.12 uuid . . . 88
9.17 ArgumentException Class Reference . . . 89

9.17.1 Description . . . 89
9.17.2 Constructor & Destructor Documentation . . . 89
9.17.2.1 ArgumentException(std::string="") . . . 89
9.17.3 Member Function Documentation . . . 89
9.17.3.1 py_what() const . . . 89
9.18 BaseProperty Class Reference . . . 89
9.18.1 Description . . . 89
9.19 ProgressMonitor::Block Class Reference . . . 90
9.19.1 Constructor & Destructor Documentation . . . 90
9.19.1.1 Block() . . . 90
9.19.2 Member Data Documentation . . . 90
9.19.2.1 name . . . 90
9.19.2.2 started . . . 90
9.19.2.3 step . . . 90
9.19.2.4 total_steps . . . 90
9.20 Server::Block Class Reference . . . 90
9.20.1 Constructor & Destructor Documentation . . . 91
9.20.1.1 Block(websocketpp::connection_hdl, const std::string &, uint64_t id) . . . 91
9.20.2 Member Data Documentation . . . 91
9.20.2.1 cell_id . . . 91
9.20.2.2 error . . . 91
9.20.2.3 hdl . . . 91
9.20.2.4 input . . . 91
9.20.2.5 output . . . 91
9.21 Cadabra Class Reference . . . 91
9.21.1 Description . . . 91
9.21.2 Constructor & Destructor Documentation . . . 92
9.21.2.1 Cadabra(int, char **) . . . 92
9.21.2.2 ~Cadabra() . . . 92
9.21.3 Member Function Documentation . . . 92

9.21.3.1 create(int, char **) . . . 92
9.21.3.2 on_activate() override . . . 92
9.21.3.3 on_open(const Gio::Application::type_vec_files &files, const Glib::ustring &hint) override . . . 92
9.21.3.4 open_help(const std::string &filename, const std::string &title) . . . 92
9.21.4 Member Data Documentation . . . 92
9.21.4.1 compute . . . 92
9.21.4.2 compute_thread . . . 92
9.22 CadabraException Class Reference . . . 93
9.22.1 Constructor & Destructor Documentation . . . 93
9.22.1.1 CadabraException(std::string) . . . 93
9.23 cadabra::canonicalise Class Reference . . . 94
9.23.1 Description . . . 94
9.23.2 Constructor & Destructor Documentation . . . 94
9.23.2.1 canonicalise(const Kernel &, Ex &) . . . 94
9.23.3 Member Function Documentation . . . 94
9.23.3.1 apply(iterator &) . . . 94
9.23.3.2 can_apply(iterator) . . . 95
9.23.3.3 get_index_set_name(iterator) const . . . 95
9.23.3.4 only_one_on_derivative(iterator index1, iterator index2) const . . . 95
9.23.3.5 position_type(iterator) const . . . 95
9.23.3.6 remove_traceless_traces(iterator &) . . . 95
9.23.3.7 remove_vanishing_numericals(iterator &) . . . 95
9.23.4 Member Data Documentation . . . 95
9.23.4.1 generating_set . . . 95
9.23.4.2 reuse_generating_set . . . 95
9.24 Server::CatchOutput Class Reference . . . 95
9.24.1 Description . . . 95
9.24.2 Constructor & Destructor Documentation . . . 96
9.24.2.1 CatchOutput() . . . 96
9.24.2.2 CatchOutput(const CatchOutput &) . . . 96

9.24.3 Member Function Documentation . . . 96
9.24.3.1 clear() . . . 96
9.24.3.2 str() const . . . 96
9.24.3.3 write(const std::string &txt) . . . 96
9.24.4 Member Data Documentation . . . 96
9.24.4.1 collect . . . 96
9.25 cadabra::CodeInput Class Reference . . . 96
9.25.1 Description . . . 96
9.25.2 Constructor & Destructor Documentation . . . 97
9.25.2.1 CodeInput(DTree::iterator, Glib::RefPtr< Gtk::TextBuffer >, double scale, int font_step) . . . 97
9.25.2.2 CodeInput(DTree::iterator, const std::string &, double scale, int font_step) . . . 97
9.25.3 Member Function Documentation . . . 98
9.25.3.1 handle_button_press(GdkEventButton *) . . . 98
9.25.3.2 handle_erase(const Gtk::TextIter &start, const Gtk::TextIter &end) . . . 98
9.25.3.3 handle_insert(const Gtk::TextIter &pos, const Glib::ustring &text, int bytes) . . . 98
9.25.3.4 init(int font_step) . . . 98
9.25.3.5 set_font_size(int num) . . . 98
9.25.3.6 slice_cell(std::string &before, std::string &after) . . . 98
9.25.3.7 update_buffer() . . . 98
9.25.4 Member Data Documentation . . . 98
9.25.4.1 buffer . . . 98
9.25.4.2 edit . . . 99
9.26 cadabra::collect_components Class Reference . . . 99
9.26.1 Description . . . 99
9.26.2 Constructor & Destructor Documentation . . . 99
9.26.2.1 collect_components(const Kernel &, Ex &) . . . 99
9.26.3 Member Function Documentation . . . 99
9.26.3.1 apply(iterator &) override . . . 99
9.26.3.2 can_apply(iterator) override . . . 99
9.27 cadabra::collect_factors Class Reference . . . 100

9.27.1 Description . . . 100
9.27.2 Member Typedef Documentation . . . 100
9.27.2.1 factor_hash_iterator_t . . . 100
9.27.2.2 factor_hash_t . . . 100
9.27.3 Constructor & Destructor Documentation . . . 100
9.27.3.1 collect_factors(const Kernel &, Ex &) . . . 100
9.27.4 Member Function Documentation . . . 100
9.27.4.1 apply(iterator &) . . . 100
9.27.4.2 can_apply(iterator) . . . 101
9.27.4.3 fill_hash_map(iterator) . . . 101
9.27.5 Member Data Documentation . . . 101
9.27.5.1 factor_hash . . . 101
9.28 cadabra::collect_terms Class Reference . . . 101
9.28.1 Description . . . 101
9.28.2 Member Typedef Documentation . . . 102
9.28.2.1 term_hash_iterator_t . . . 102
9.28.2.2 term_hash_t . . . 102
9.28.3 Constructor & Destructor Documentation . . . 102
9.28.3.1 collect_terms(const Kernel &, Ex &) . . . 102
9.28.4 Member Function Documentation . . . 102
9.28.4.1 apply(iterator &) . . . 102
9.28.4.2 can_apply(iterator) . . . 102
9.28.4.3 collect_from_hash_map() . . . 102
9.28.4.4 fill_hash_map(iterator) . . . 102
9.28.4.5 fill_hash_map(sibling_iterator, sibling_iterator) . . . 102
9.28.4.6 remove_zeroed_terms(sibling_iterator, sibling_iterator) . . . 102
9.28.5 Member Data Documentation . . . 102
9.28.5.1 term_hash . . . 102
9.29 combin::combinations< T > Class Template Reference . . . 103
9.29.1 Member Typedef Documentation . . . 104

9.29.1.1 const_iterator . . . 104
9.29.1.2 permuted_sets_t . . . 104
9.29.2 Constructor & Destructor Documentation . . . 104
9.29.2.1 combinations() . . . 104
9.29.2.2 combinations(const std::vector< T > &) . . . 104
9.29.2.3 ~combinations() . . . 104
9.29.3 Member Function Documentation . . . 104
9.29.3.1 clear() . . . 104
9.29.3.2 clear_results() . . . 104
9.29.3.3 multiplier(unsigned int) const . . . 104
9.29.3.4 operator[](unsigned int) const . . . 104
9.29.3.5 ordersign(unsigned int) const . . . 104
9.29.3.6 size() const . . . 104
9.29.3.7 vector_generated(const std::vector< unsigned int > &) . . . 104
9.29.4 Member Data Documentation . . . 104
9.29.4.1 storage . . . 104
9.30 combin::combinations_base< T > Class Template Reference . . . 105
9.30.1 Member Enumeration Documentation . . . 106
9.30.1.1 weight_cond . . . 106
9.30.2 Constructor & Destructor Documentation . . . 106
9.30.2.1 combinations_base() . . . 106
9.30.2.2 combinations_base(const std::vector< T > &) . . . 106
9.30.2.3 ~combinations_base() . . . 106
9.30.3 Member Function Documentation . . . 106
9.30.3.1 clear() . . . 106
9.30.3.2 clear_results() . . . 106
9.30.3.3 entry_accepted(unsigned int current) const . . . 107
9.30.3.4 final_weight_constraints_check() const . . . 107
9.30.3.5 is_allowed_by_weight_constraints(unsigned int i) . . . 107
9.30.3.6 multiplier(const std::vector< T > &) const . . . 107

9.30.3.7 nextstep(unsigned int current, unsigned int fromalgehad, unsigned int groupindex, std::vector< bool > algehad) . . . 107
9.30.3.8 permute(long start=-1, long end=-1) . . . 107
9.30.3.9 restore_weights(unsigned int i) . . . 107
9.30.3.10 set_unit_sublengths() . . . 107
9.30.3.11 sum_of_sublengths() const . . . 107
9.30.3.12 total_permutations() const . . . 107
9.30.3.13 update_weights(unsigned int i) . . . 107
9.30.3.14 vector_generated(const std::vector< unsigned int > &)=0 . . . 107
9.30.4 Member Data Documentation . . . 107
9.30.4.1 block_length . . . 107
9.30.4.2 current_weight . . . 107
9.30.4.3 end_ . . . 107
9.30.4.4 input_asym . . . 107
9.30.4.5 max_weights . . . 107
9.30.4.6 multiple_pick . . . 108
9.30.4.7 original . . . 108
9.30.4.8 start_ . . . 108
9.30.4.9 sub_problem_blocksize . . . 108
9.30.4.10 sublengths . . . 108
9.30.4.11 temparr . . . 108
9.30.4.12 vector_generated_called_ . . . 108
9.30.4.13 weight_conditions . . . 108
9.30.4.14 weights . . . 108
9.31 cadabra::combine Class Reference . . . 108
9.31.1 Member Typedef Documentation . . . 109
9.31.1.1 indexlocmap_t . . . 109
9.31.2 Constructor & Destructor Documentation . . . 109
9.31.2.1 combine(const Kernel &, Ex &) . . . 109
9.31.3 Member Function Documentation . . . 109
9.31.3.1 apply(iterator &) override . . . 109

9.31.3.2 can_apply(iterator) override . . . 109
9.31.4 Member Data Documentation . . . 109
9.31.4.1 iloc . . . 109
9.32 cadabra::Commuting Class Reference . . . 109
9.32.1 Member Function Documentation . . . 110
9.32.1.1 name() const . . . 110
9.32.1.2 sign() const . . . 110
9.33 cadabra::CommutingAsProduct Class Reference . . . 110
9.33.1 Member Function Documentation . . . 111
9.33.1.1 name() const . . . 111
9.34 cadabra::CommutingAsSum Class Reference . . . 111
9.34.1 Member Function Documentation . . . 111
9.34.1.1 name() const . . . 111
9.35 cadabra::CommutingBehaviour Class Reference . . . 112
9.35.1 Member Function Documentation . . . 112
9.35.1.1 equals(const property ∗) const . . . 112
9.35.1.2 sign() const =0 . . . 112
9.36 cadabra::complete Class Reference . . . 112
9.36.1 Description . . . 112
9.36.2 Constructor & Destructor Documentation . . . 113
9.36.2.1 complete(const Kernel &, Ex &, Ex &) . . . 113
9.36.3 Member Function Documentation . . . 113
9.36.3.1 apply(iterator &) override . . . 113
9.36.3.2 can_apply(iterator) override . . . 113
9.36.4 Member Data Documentation . . . 113
9.36.4.1 goal . . . 113
9.37 component Class Reference . . . 113
9.37.1 Constructor & Destructor Documentation . . . 114
9.37.1.1 component(const Kernel &, Ex &ex, Ex &rules) . . . 114
9.37.2 Member Function Documentation . . . 114

9.37.2.1 apply(iterator &) . . . 114
9.37.2.2 can_apply(iterator) . . . 114
9.38 cadabra::ComputeThread Class Reference . . . 114
9.38.1 Description . . . 114
9.38.2 Constructor & Destructor Documentation . . . 115
9.38.2.1 ComputeThread() . . . 115
9.38.2.2 ComputeThread(const ComputeThread &)=delete . . . 115
9.38.2.3 ∼ComputeThread() . . . 115
9.38.3 Member Function Documentation . . . 115
9.38.3.1 all_cells_nonrunning() . . . 115
9.38.3.2 execute_cell(DTree::iterator) . . . 116
9.38.3.3 find_cell_by_id(DataCell::id_t, bool remove) . . . 116
9.38.3.4 init() . . . 116
9.38.3.5 number_of_cells_executing(void) const . . . 116
9.38.3.6 on_close(websocketpp::connection_hdl hdl) . . . 116
9.38.3.7 on_fail(websocketpp::connection_hdl hdl) . . . 116
9.38.3.8 on_message(websocketpp::connection_hdl hdl, message_ptr msg) . . . 116
9.38.3.9 on_open(websocketpp::connection_hdl hdl) . . . 116
9.38.3.10 restart_kernel() . . . 116
9.38.3.11 run() . . . 116
9.38.3.12 set_master(GUIBase ∗, DocumentThread ∗) . . . 116
9.38.3.13 stop() . . . 116
9.38.3.14 terminate() . . . 116
9.38.3.15 try_connect() . . . 117
9.38.3.16 try_spawn_server() . . . 117
9.38.4 Member Data Documentation . . . 117
9.38.4.1 connection . . . 117
9.38.4.2 connection_is_open . . . 117
9.38.4.3 docthread . . . 117
9.38.4.4 gui . . . 117

9.38.4.5 gui_thread_id . . . 117
9.38.4.6 our_connection_hdl . . . 117
9.38.4.7 port . . . 117
9.38.4.8 restarting_kernel . . . 117
9.38.4.9 running_cells . . . 117
9.38.4.10 server_pid . . . 117
9.38.4.11 server_stderr . . . 117
9.38.4.12 server_stdout . . . 117
9.38.4.13 wsclient . . . 117
9.39 Server::Connection Class Reference . . . 117
9.39.1 Constructor & Destructor Documentation . . . 118
9.39.1.1 Connection() . . . 118
9.39.2 Member Data Documentation . . . 118
9.39.2.1 hdl . . . 118
9.39.2.2 uuid . . . 118
9.40 ConsistencyException Class Reference . . . 118
9.40.1 Constructor & Destructor Documentation . . . 118
9.40.1.1 ConsistencyException(std::string) . . . 118
9.41 cadabra::Coordinate Class Reference . . . 119
9.41.1 Member Function Documentation . . . 119
9.41.1.1 name() const . . . 119
9.42 cadabra::DAntiSymmetric Class Reference . . . 119
9.42.1 Constructor & Destructor Documentation . . . 120
9.42.1.1 ~DAntiSymmetric() . . . 120
9.42.2 Member Function Documentation . . . 120
9.42.2.1 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const override . . . 120
9.42.2.2 name() const override . . . 120
9.42.2.3 size(const Properties &, Ex &, Ex::iterator) const override . . . 120
9.43 cadabra::DataCell Class Reference . . . 120
9.43.1 Description . . . 120

9.43.2 Member Enumeration Documentation . . . 121
9.43.2.1 CellType . . . 121
9.43.3 Constructor & Destructor Documentation . . . 122
9.43.3.1 DataCell(CellType t=CellType::python, const std::string &str="""", bool hidden=false) 122
9.43.3.2 DataCell(id_t, CellType t=CellType::python, const std::string &str="""", bool hidden=false) . . . 122
9.43.3.3 DataCell(const DataCell &) . . . 122
9.43.4 Member Function Documentation . . . 122
9.43.4.1 id() const . . . 122
9.43.5 Member Data Documentation . . . 122
9.43.5.1 cell_type . . . 122
9.43.5.2 hidden . . . 122
9.43.5.3 max_serial_number . . . 122
9.43.5.4 running . . . 122
9.43.5.5 sensitive . . . 123
9.43.5.6 serial_mutex . . . 123
9.43.5.7 serial_number . . . 123
9.43.5.8 textbuf . . . 123
9.44 cadabra::decompose_product Class Reference . . . 123
9.44.1 Description . . . 123
9.44.2 Member Typedef Documentation . . . 124
9.44.2.1 numtab_t . . . 124
9.44.2.2 numtabs_t . . . 124
9.44.2.3 sibtab_t . . . 124
9.44.2.4 sibtabs_t . . . 124
9.44.3 Constructor & Destructor Documentation . . . 124
9.44.3.1 decompose_product(const Kernel &, Ex &tr) . . . 124
9.44.4 Member Function Documentation . . . 124
9.44.4.1 apply(iterator &) override . . . 124
9.44.4.2 can_apply(iterator) override . . . 125
9.44.4.3 fill_asym_ranges(TableauBase::tab_t &tab, int offset, combin::range_vector_t &) 125

9.44.4.4 indices_equivalent(iterator it) const . . . 125
9.44.4.5 project_onto_initial_symmetries(Ex &rep, iterator rr, young_project &yp, const TableauBase *tt, iterator ff, int offset, const TableauBase::tab_t &thetab, bool remove_traces) . . . 125
9.44.5 Member Data Documentation . . . 125
9.44.5.1 asym_ranges . . . 125
9.44.5.2 dim . . . 125
9.44.5.3 f1 . . . 125
9.44.5.4 f2 . . . 125
9.44.5.5 ind1 . . . 125
9.44.5.6 ind2 . . . 125
9.44.5.7 nt1 . . . 125
9.44.5.8 nt2 . . . 125
9.44.5.9 t1 . . . 125
9.44.5.10 t1tab . . . 125
9.44.5.11 t2 . . . 125
9.44.5.12 t2tab . . . 125
9.45 cadabra::Depends Class Reference . . . 126
9.45.1 Member Function Documentation . . . 126
9.45.1.1 dependencies(const Kernel &, Ex::iterator) const override . . . 126
9.45.1.2 name() const override . . . 126
9.45.1.3 parse(const Kernel &, keyval_t &) override . . . 126
9.45.1.4 unnamed_argument() const override . . . 127
9.45.2 Member Data Documentation . . . 127
9.45.2.1 dependencies_ . . . 127
9.46 cadabra::DependsBase Class Reference . . . 127
9.46.1 Member Function Documentation . . . 127
9.46.1.1 dependencies(const Kernel &, Ex::iterator) const =0 . . . 127
9.47 cadabra::DependsInherit Class Reference . . . 128
9.47.1 Member Function Documentation . . . 128
9.47.1.1 dependencies(const Kernel &, Ex::iterator) const override . . . 128

9.47.1.2 name() const override . . . 128
9.48 cadabra::Derivative Class Reference . . . 129
9.48.1 Constructor & Destructor Documentation . . . 129
9.48.1.1 ∼Derivative() . . . 129
9.48.2 Member Function Documentation . . . 129
9.48.2.1 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const override . . . 129
9.48.2.2 name() const override . . . 130
9.48.2.3 size(const Properties &, Ex &, Ex::iterator) const override . . . 130
9.48.2.4 value(const Kernel &, Ex::iterator, const std::string &forcedlabel) const override . . . 130
9.49 cadabra::Diagonal Class Reference . . . 130
9.49.1 Member Function Documentation . . . 131
9.49.1.1 name() const . . . 131
9.50 cadabra::DifferentialForm Class Reference . . . 131
9.50.1 Member Function Documentation . . . 131
9.50.1.1 degree(const Properties &, Ex::iterator) const override . . . 131
9.50.1.2 name() const override . . . 132
9.50.1.3 parse(const Kernel &, keyval_t &) override . . . 132
9.50.2 Member Data Documentation . . . 132
9.50.2.1 degree_ . . . 132
9.51 cadabra::DifferentialFormBase Class Reference . . . 132
9.51.1 Member Function Documentation . . . 132
9.51.1.1 degree(const Properties &, Ex::iterator) const =0 . . . 132
9.52 cadabra::DiracBar Class Reference . . . 133
9.52.1 Member Function Documentation . . . 133
9.52.1.1 name() const . . . 133
9.53 cadabra::DisplayBase Class Reference . . . 133
9.53.1 Description . . . 133
9.53.2 Constructor & Destructor Documentation . . . 134
9.53.2.1 DisplayBase(const Kernel &, const Ex &) . . . 134
9.53.3 Member Function Documentation . . . 134

9.53.3.1 dispatch(std::ostream &, Ex::iterator)=0 . . . 134
9.53.3.2 needs_brackets(Ex::iterator it)=0 . . . 134
9.53.3.3 output(std::ostream &) . . . 134
9.53.3.4 output(std::ostream &, Ex::iterator) . . . 134
9.53.4 Member Data Documentation . . . 134
9.53.4.1 kernel . . . 134
9.53.4.2 tree . . . 134
9.54 cadabra::DisplaySympy Class Reference . . . 135
9.54.1 Constructor & Destructor Documentation . . . 136
9.54.1.1 DisplaySympy(const Kernel &, const Ex &) . . . 136
9.54.2 Member Function Documentation . . . 136
9.54.2.1 children_have_brackets(Ex::iterator ch) const . . . 136
9.54.2.2 dispatch(std::ostream &, Ex::iterator) override . . . 136
9.54.2.3 import(Ex &) . . . 136
9.54.2.4 needs_brackets(Ex::iterator it) override . . . 136
9.54.2.5 print_arrowlike(std::ostream &, Ex::iterator) . . . 137
9.54.2.6 print_children(std::ostream &, Ex::iterator, int skip=0) . . . 137
9.54.2.7 print_closing_bracket(std::ostream &, str_node::bracket_t) . . . 137
9.54.2.8 print_commalike(std::ostream &, Ex::iterator) . . . 137
9.54.2.9 print_components(std::ostream &, Ex::iterator) . . . 137
9.54.2.10 print_equalitylike(std::ostream &, Ex::iterator) . . . 137
9.54.2.11 print_fraclike(std::ostream &, Ex::iterator) . . . 137
9.54.2.12 print_intlike(std::ostream &, Ex::iterator) . . . 137
9.54.2.13 print_matrix(std::ostream &str, Ex::iterator it) . . . 137
9.54.2.14 print_multiplier(std::ostream &, Ex::iterator) . . . 137
9.54.2.15 print_opening_bracket(std::ostream &, str_node::bracket_t) . . . 137
9.54.2.16 print_other(std::ostream &str, Ex::iterator it) . . . 137
9.54.2.17 print_parent_rel(std::ostream &, str_node::parent_rel_t, bool first) . . . 137
9.54.2.18 print_partial(std::ostream &str, Ex::iterator it) . . . 137
9.54.2.19 print_powlike(std::ostream &, Ex::iterator) . . . 137

9.54.2.20 print_productlike(std::ostream &, Ex::iterator, const std::string &inbetween) . . . 137
9.54.2.21 print_sumlike(std::ostream &, Ex::iterator) . . . 138
9.54.3 Member Data Documentation . . . 138
9.54.3.1 depsyms . . . 138
9.54.3.2 symmap . . . 138
9.55 cadabra::DisplayTerminal Class Reference . . . 138
9.55.1 Constructor & Destructor Documentation . . . 139
9.55.1.1 DisplayTerminal(const Kernel &, const Ex &) . . . 139
9.55.2 Member Function Documentation . . . 139
9.55.2.1 children_have_brackets(Ex::iterator ch) const . . . 139
9.55.2.2 dispatch(std::ostream &, Ex::iterator) override . . . 139
9.55.2.3 needs_brackets(Ex::iterator it) override . . . 139
9.55.2.4 print_arrowlike(std::ostream &, Ex::iterator) . . . 140
9.55.2.5 print_children(std::ostream &, Ex::iterator, int skip=0) . . . 140
9.55.2.6 print_closing_bracket(std::ostream &, str_node::bracket_t, str_node::parent_rel↩ _t) . . . 140
9.55.2.7 print_commalike(std::ostream &, Ex::iterator) . . . 140
9.55.2.8 print_commutator(std::ostream &str, Ex::iterator it, bool comm) . . . 140
9.55.2.9 print_components(std::ostream &, Ex::iterator) . . . 140
9.55.2.10 print_equalitylike(std::ostream &, Ex::iterator) . . . 140
9.55.2.11 print_fraclike(std::ostream &, Ex::iterator) . . . 140
9.55.2.12 print_intlike(std::ostream &, Ex::iterator) . . . 140
9.55.2.13 print_multiplier(std::ostream &, Ex::iterator, int mult=1) . . . 140
9.55.2.14 print_opening_bracket(std::ostream &, str_node::bracket_t, str_node::parent_↩ rel_t) . . . 140
9.55.2.15 print_other(std::ostream &, Ex::iterator) . . . 140
9.55.2.16 print_parent_rel(std::ostream &, str_node::parent_rel_t, bool first) . . . 140
9.55.2.17 print_powlike(std::ostream &, Ex::iterator) . . . 140
9.55.2.18 print_productlike(std::ostream &, Ex::iterator, const std::string &inbetween) . . . 140
9.55.2.19 print_sumlike(std::ostream &, Ex::iterator) . . . 140
9.55.3 Member Data Documentation . . . 140

9.55.3.1 bracket_level . . . 140
9.55.3.2 symmap . . . 140
9.56 cadabra::DisplayTeX Class Reference . . . 141
9.56.1 Constructor & Destructor Documentation . . . 142
9.56.1.1 DisplayTeX(const Kernel &, const Ex &) . . . 142
9.56.2 Member Function Documentation . . . 142
9.56.2.1 children_have_brackets(Ex::iterator ch) const . . . 142
9.56.2.2 dispatch(std::ostream &, Ex::iterator) override . . . 142
9.56.2.3 needs_brackets(Ex::iterator it) override . . . 142
9.56.2.4 print_arrowlike(std::ostream &, Ex::iterator) . . . 143
9.56.2.5 print_children(std::ostream &, Ex::iterator, int skip=0) . . . 143
9.56.2.6 print_closing_bracket(std::ostream &, str_node::bracket_t, str_node::parent_rel_t) . . . 143
9.56.2.7 print_commalike(std::ostream &, Ex::iterator) . . . 143
9.56.2.8 print_commutator(std::ostream &str, Ex::iterator it, bool comm) . . . 143
9.56.2.9 print_components(std::ostream &, Ex::iterator) . . . 143
9.56.2.10 print_conditional(std::ostream &, Ex::iterator) . . . 143
9.56.2.11 print_equalitylike(std::ostream &, Ex::iterator) . . . 143
9.56.2.12 print_fraclike(std::ostream &, Ex::iterator) . . . 143
9.56.2.13 print_ftableau(std::ostream &, Ex::iterator) . . . 143
9.56.2.14 print_indexbracket(std::ostream &, Ex::iterator) . . . 143
9.56.2.15 print_intlike(std::ostream &, Ex::iterator) . . . 143
9.56.2.16 print_multiplier(std::ostream &, Ex::iterator, int mult=1) . . . 143
9.56.2.17 print_opening_bracket(std::ostream &, str_node::bracket_t, str_node::parent_rel_t) . . . 143
9.56.2.18 print_other(std::ostream &str, Ex::iterator it) . . . 143
9.56.2.19 print_parent_rel(std::ostream &, str_node::parent_rel_t, bool first) . . . 143
9.56.2.20 print_powlike(std::ostream &, Ex::iterator) . . . 143
9.56.2.21 print_productlike(std::ostream &, Ex::iterator, const std::string &inbetween) . . . 143
9.56.2.22 print_relation(std::ostream &, Ex::iterator) . . . 144
9.56.2.23 print_sumlike(std::ostream &, Ex::iterator) . . . 144

9.56.2.24 print_tableau(std::ostream &, Ex::iterator) . . . 144
9.56.2.25 print_wedgeproduct(std::ostream &, Ex::iterator) . . . 144
9.56.2.26 reads_as_operator(Ex::iterator obj, Ex::iterator arg) const . . . 144
9.56.2.27 texify(std::string) const . . . 144
9.56.3 Member Data Documentation . . . 144
9.56.3.1 bracket_level . . . 144
9.56.3.2 curly_bracket_operators . . . 144
9.56.3.3 extra_brackets_for_symbols . . . 144
9.56.3.4 latex_linefeeds . . . 144
9.56.3.5 latex_spacing . . . 144
9.56.3.6 print_star . . . 144
9.56.3.7 symmap . . . 144
9.56.3.8 tight_plus . . . 144
9.56.3.9 tight_star . . . 144
9.56.3.10 utf8_output . . . 144
9.57 cadabra::Distributable Class Reference . . . 145
9.57.1 Constructor & Destructor Documentation . . . 145
9.57.1.1 ∼Distributable() . . . 145
9.57.2 Member Function Documentation . . . 145
9.57.2.1 name() const . . . 145
9.58 cadabra::distribute Class Reference . . . 146
9.58.1 Description . . . 146
9.58.2 Constructor & Destructor Documentation . . . 146
9.58.2.1 distribute(const Kernel &, Ex &) . . . 146
9.58.3 Member Function Documentation . . . 146
9.58.3.1 apply(iterator &) . . . 146
9.58.3.2 can_apply(iterator) . . . 146
9.59 cadabra::DocumentThread Class Reference . . . 147
9.59.1 Description . . . 147
9.59.2 Member Typedef Documentation . . . 148

9.59.2.1 ActionStack . . . 148
9.59.3 Member Enumeration Documentation . . . 148
9.59.3.1 help_t . . . 148
9.59.4 Constructor & Destructor Documentation . . . 149
9.59.4.1 DocumentThread(GUIBase *) . . . 149
9.59.4.2 DocumentThread(const DocumentThread &)=delete . . . 149
9.59.5 Member Function Documentation . . . 149
9.59.5.1 build_visual_representation() . . . 149
9.59.5.2 help_type_and_topic(const std::string &before, const std::string &after, help_↩ t &help_type, std::string &help_topic) const . . . 149
9.59.5.3 is_registered() const . . . 149
9.59.5.4 load_from_string(const std::string &) . . . 149
9.59.5.5 new_document() . . . 149
9.59.5.6 process_action_queue() . . . 149
9.59.5.7 queue_action(std::shared_ptr< ActionBase >) . . . 150
9.59.5.8 set_compute_thread(ComputeThread *) . . . 150
9.59.5.9 set_user_details(const std::string &name, const std::string &email, const std↩ ::string &affiliation) . . . 150
9.59.5.10 undo() . . . 150
9.59.6 Member Data Documentation . . . 150
9.59.6.1 ActionAddCell . . . 150
9.59.6.2 ActionEraseText . . . 150
9.59.6.3 ActionInsertText . . . 150
9.59.6.4 ActionPositionCursor . . . 150
9.59.6.5 ActionRemoveCell . . . 150
9.59.6.6 ActionSetRunStatus . . . 150
9.59.6.7 ActionSplitCell . . . 150
9.59.6.8 compute . . . 150
9.59.6.9 disable_stacks . . . 150
9.59.6.10 doc . . . 150
9.59.6.11 gui . . . 151

9.59.6.12 pending_actions . . . 151
9.59.6.13 redo_stack . . . 151
9.59.6.14 registered . . . 151
9.59.6.15 stack_mutex . . . 151
9.59.6.16 undo_stack . . . 151
9.60 cadabra::drop_keep_weight Class Reference . . . 151
9.60.1 Constructor & Destructor Documentation . . . 152
9.60.1.1 drop_keep_weight(const Kernel &, Ex &, Ex &) . . . 152
9.60.2 Member Function Documentation . . . 152
9.60.2.1 can_apply(iterator) override . . . 152
9.60.2.2 do_apply(iterator &, bool keepthem) . . . 152
9.60.3 Member Data Documentation . . . 152
9.60.3.1 arg . . . 152
9.60.3.2 gmn . . . 152
9.60.3.3 label . . . 152
9.60.3.4 weight . . . 152
9.60.3.5 wgh . . . 152
9.61 cadabra::drop_weight Class Reference . . . 153
9.61.1 Constructor & Destructor Documentation . . . 153
9.61.1.1 drop_weight(const Kernel &, Ex &, Ex &) . . . 153
9.61.2 Member Function Documentation . . . 153
9.61.2.1 apply(iterator &) override . . . 153
9.62 echokernel.EchoKernel Class Reference . . . 153
9.62.1 Member Function Documentation . . . 154
9.62.1.1 do_execute(self, code, silent, store_history=True, user_expressions=None, allow_stdin=False) . . . 154
9.62.2 Member Data Documentation . . . 154
9.62.2.1 banner . . . 154
9.62.2.2 implementation . . . 154
9.62.2.3 implementation_version . . . 154
9.62.2.4 language . . . 154

9.62.2.5 language_info . . . 154
9.62.2.6 language_version . . . 154
9.63 cadabra::einsteinify Class Reference . . . 154
9.63.1 Constructor & Destructor Documentation . . . 155
9.63.1.1 einsteinify(const Kernel &, Ex &, Ex &) . . . 155
9.63.2 Member Function Documentation . . . 155
9.63.2.1 apply(iterator &) override . . . 155
9.63.2.2 can_apply(iterator) override . . . 155
9.63.3 Member Data Documentation . . . 155
9.63.3.1 metric . . . 155
9.64 cadabra::eliminate_converter Class Reference . . . 155
9.64.1 Constructor & Destructor Documentation . . . 156
9.64.1.1 eliminate_converter(const Kernel &, Ex &, Ex &) . . . 156
9.64.2 Member Function Documentation . . . 156
9.64.2.1 apply(iterator &) override . . . 156
9.64.2.2 can_apply(iterator) override . . . 156
9.64.2.3 handle_one_index(iterator, iterator, iterator, sibling_iterator) . . . 156
9.64.2.4 is_conversion_object(iterator) const =0 . . . 156
9.64.3 Member Data Documentation . . . 157
9.64.3.1 ind_dummy . . . 157
9.64.3.2 ind_free . . . 157
9.64.3.3 preferred . . . 157
9.65 cadabra::eliminate_kronecker Class Reference . . . 157
9.65.1 Constructor & Destructor Documentation . . . 157
9.65.1.1 eliminate_kronecker(const Kernel &, Ex &) . . . 157
9.65.2 Member Function Documentation . . . 157
9.65.2.1 apply(iterator &) override . . . 157
9.65.2.2 can_apply(iterator) override . . . 158
9.66 cadabra::eliminate_metric Class Reference . . . 158
9.66.1 Constructor & Destructor Documentation . . . 158

9.66.1.1 eliminate_metric(const Kernel &, Ex &, Ex &) . . . 158
9.66.2 Member Function Documentation . . . 158
9.66.2.1 is_conversion_object(iterator) const override . . . 158
9.67 cadabra::epsilon_to_delta Class Reference . . . 159
9.67.1 Constructor & Destructor Documentation . . . 159
9.67.1.1 epsilon_to_delta(const Kernel &, Ex &, bool reduce_) . . . 159
9.67.2 Member Function Documentation . . . 159
9.67.2.1 apply(iterator &) override . . . 159
9.67.2.2 can_apply(iterator) override . . . 159
9.67.3 Member Data Documentation . . . 160
9.67.3.1 epsilons . . . 160
9.67.3.2 reduce . . . 160
9.67.3.3 repdelta . . . 160
9.67.3.4 signature . . . 160
9.68 cadabra::EpsilonTensor Class Reference . . . 160
9.68.1 Constructor & Destructor Documentation . . . 161
9.68.1.1 ∼EpsilonTensor() . . . 161
9.68.2 Member Function Documentation . . . 161
9.68.2.1 name() const override . . . 161
9.68.2.2 parse(const Kernel &, keyval_t &) override . . . 161
9.68.3 Member Data Documentation . . . 161
9.68.3.1 krdelta . . . 161
9.68.3.2 metric . . . 161
9.69 exec_stream_t::error_t Class Reference . . . 161
9.69.1 Constructor & Destructor Documentation . . . 162
9.69.1.1 error_t(std::string const &msg) . . . 162
9.69.1.2 error_t(std::string const &msg, error_code_t code) . . . 162
9.69.1.3 ∼error_t() . . . 162
9.69.1.4 error_t() . . . 162
9.69.2 Member Function Documentation . . . 162

9.69.2.1 compose(std::string const &msg, error_code_t code) . . . 162
9.69.2.2 what() const . . . 162
9.69.3 Member Data Documentation . . . 162
9.69.3.1 m_msg . . . 162
9.70 cadabra::evaluate Class Reference . . . 162
9.70.1 Constructor & Destructor Documentation . . . 163
9.70.1.1 evaluate(const Kernel &, Ex &, const Ex &component_values, bool rhs=false) . . . 163
9.70.2 Member Function Documentation . . . 163
9.70.2.1 apply(iterator &) override . . . 163
9.70.2.2 can_apply(iterator) override . . . 164
9.70.2.3 cleanup_components(iterator it1) . . . 164
9.70.2.4 dense_factor(iterator sib, const index_map_t &ind_free, const index_map_t &ind_dummy) . . . 164
9.70.2.5 dependencies(iterator it) . . . 164
9.70.2.6 handle_components(iterator it) . . . 164
9.70.2.7 handle_derivative(iterator it) . . . 164
9.70.2.8 handle_epsilon(iterator it) . . . 164
9.70.2.9 handle_factor(sibling_iterator sib, const index_map_t &full_ind_free) . . . 164
9.70.2.10 handle_prod(iterator it) . . . 164
9.70.2.11 handle_sum(iterator it) . . . 164
9.70.2.12 is_component(iterator it) const . . . 164
9.70.2.13 merge_component_children(iterator it) . . . 164
9.70.2.14 merge_components(iterator it1, iterator it2) . . . 165
9.70.2.15 simplify_components(iterator) . . . 165
9.70.2.16 wrap_scalar_in_components_node(iterator sib) . . . 165
9.70.3 Member Data Documentation . . . 165
9.70.3.1 components . . . 165
9.70.3.2 only_rhs . . . 165
9.71 cadabra::Ex Class Reference . . . 165
9.71.1 Description . . . 165
9.71.2 Member Enumeration Documentation . . . 167

9.71.2.1 result_t . . . 167
9.71.3 Constructor & Destructor Documentation . . . 167
9.71.3.1 Ex() . . . 167
9.71.3.2 Ex(tree< str_node >::iterator) . . . 167
9.71.3.3 Ex(const str_node &) . . . 167
9.71.3.4 Ex(const Ex &) . . . 167
9.71.3.5 Ex(const std::string &) . . . 167
9.71.3.6 Ex(int) . . . 168
9.71.4 Member Function Documentation . . . 168
9.71.4.1 arg(iterator, unsigned int) . . . 168
9.71.4.2 arg_size(sibling_iterator) . . . 168
9.71.4.3 arg_to_num(sibling_iterator, unsigned int) const . . . 168
9.71.4.4 calc_hash(iterator it) const . . . 168
9.71.4.5 changed_state() . . . 168
9.71.4.6 equation_by_name(nset_t::iterator it) const . . . 168
9.71.4.7 equation_by_name(nset_t::iterator it, unsigned int &) const . . . 168
9.71.4.8 equation_by_number(unsigned int i) const . . . 168
9.71.4.9 equation_by_number_or_name(iterator it, unsigned int last_used_equation) const 168
9.71.4.10 equation_by_number_or_name(iterator it, unsigned int last_used_equation, unsigned int &) const . . . 168
9.71.4.11 equation_label(iterator it) const . . . 168
9.71.4.12 equation_number(iterator it) const . . . 168
9.71.4.13 equation_number_or_name(iterator it, unsigned int last_used_equation) const . 168
9.71.4.14 erase_expression(iterator it) . . . 168
9.71.4.15 flatten_and_erase(iterator position) . . . 168
9.71.4.16 history_size() const . . . 169
9.71.4.17 is_rational() const . . . 169
9.71.4.18 list_unwrap_single_element(iterator &) . . . 169
9.71.4.19 list_wrap_single_element(iterator &) . . . 169
9.71.4.20 move_index(iterator position, const iterator &from) . . . 169
9.71.4.21 named_parent(iterator it, const std::string &) const . . . 169

9.71.4.22 number_of_equations() const . . . 169
9.71.4.23 number_of_steps(iterator it) const . . . 169
9.71.4.24 operator==(const Ex &other) const . . . 169
9.71.4.25 pop_history() . . . 169
9.71.4.26 print_entire_tree(std::ostream &str) const . . . 169
9.71.4.27 print_python(std::ostream &str, Ex::iterator it) . . . 170
9.71.4.28 print_recursive_treeform(std::ostream &str, Ex::iterator it) . . . 170
9.71.4.29 print_recursive_treeform(std::ostream &str, Ex::iterator it, unsigned int &number) 170
9.71.4.30 procedure_by_name(nset_t::iterator it) const . . . 170
9.71.4.31 push_history(Ex selector) . . . 170
9.71.4.32 replace_index(iterator position, const iterator &from, bool keep_parent_rel=false) 170
9.71.4.33 reset_state() . . . 170
9.71.4.34 state() const . . . 170
9.71.4.35 to_rational() const . . . 170
9.71.4.36 update_state(result_t) . . . 170
9.71.5 Member Data Documentation . . . 170
9.71.5.1 history . . . 170
9.71.5.2 selectors . . . 170
9.71.5.3 state_ . . . 171
9.72 cadabra::Ex_comparator Class Reference . . . 171
9.72.1 Description . . . 171
9.72.2 Member Typedef Documentation . . . 172
9.72.2.1 replacement_map_t . . . 172
9.72.2.2 subtree_replacement_map_t . . . 172
9.72.3 Member Enumeration Documentation . . . 173
9.72.3.1 match_t . . . 173
9.72.3.2 useprops_t . . . 173
9.72.4 Constructor & Destructor Documentation . . . 173
9.72.4.1 Ex_comparator(const Properties &) . . . 173
9.72.5 Member Function Documentation . . . 173

9.72.5.1 can_move_adjacent(Ex::iterator prod, Ex::sibling_iterator one, Ex::sibling_iterator two, bool fix_one=false) . . . 173

9.72.5.2 can_swap(Ex::iterator one, Ex::iterator two, match_t subtree_comparison, bool ignore_implicit_indices=false) . . . 173

9.72.5.3 can_swap_ilist_ilist(Ex::iterator obj1, Ex::iterator obj2) . . . 174

9.72.5.4 can_swap_prod_obj(Ex::iterator prod, Ex::iterator obj, bool) . . . 174

9.72.5.5 can_swap_prod_prod(Ex::iterator prod1, Ex::iterator prod2, bool) . . . 174

9.72.5.6 can_swap_prod_sum(Ex::iterator prod, Ex::iterator sum, bool) . . . 174

9.72.5.7 can_swap_sum_obj(Ex::iterator sum, Ex::iterator obj, bool) . . . 174

9.72.5.8 can_swap_sum_sum(Ex::iterator sum1, Ex::iterator sum2, bool) . . . 174

9.72.5.9 clear() . . . 174

9.72.5.10 compare(const Ex::iterator &, const Ex::iterator &, bool nobrackets=false, useprops_t use_props=useprops_t::always, bool ignore_parent_rel=false) . . . 174

9.72.5.11 equal_subtree(Ex::iterator i1, Ex::iterator i2, useprops_t use_props=useprops_t::always, bool ignore_parent_rel=false) . . . 174

9.72.5.12 match_subproduct(const Ex &, Ex::sibling_iterator lhs, Ex::sibling_iterator tofind, Ex::sibling_iterator st, Ex::iterator conditions) . . . 174

9.72.5.13 match_subsum(const Ex &, Ex::sibling_iterator lhs, Ex::sibling_iterator tofind, Ex::sibling_iterator st, Ex::iterator conditions) . . . 175

9.72.5.14 report(match_t r) const . . . 175

9.72.5.15 satisfies_conditions(Ex::iterator conditions, std::string &error) . . . 175

9.72.5.16 should_swap(Ex::iterator obj, match_t subtree_comparison) . . . 175

9.72.5.17 tab() const . . . 175

9.72.6 Member Data Documentation . . . 175

9.72.6.1 factor_locations . . . 175

9.72.6.2 factor_moving_signs . . . 175

9.72.6.3 index_value_map . . . 175

9.72.6.4 lhs_contains_dummies . . . 175

9.72.6.5 offset . . . 176

9.72.6.6 properties . . . 176

9.72.6.7 replacement_map . . . 176

9.72.6.8 subtree_replacement_map . . . 176

9.72.6.9 term_ratio . . . 176

9.73 cadabra::Ex_is_equivalent Class Reference . . . 176
9.73.1 Description . . . 176
9.73.2 Constructor & Destructor Documentation . . . 176
9.73.2.1 Ex_is_equivalent(const Properties &) . . . 176
9.73.3 Member Function Documentation . . . 176
9.73.3.1 operator()(const Ex &, const Ex &) . . . 176
9.73.4 Member Data Documentation . . . 176
9.73.4.1 properties . . . 176
9.74 cadabra::Ex_is_less Class Reference . . . 177
9.74.1 Constructor & Destructor Documentation . . . 177
9.74.1.1 Ex_is_less(const Properties &, int mod_prel) . . . 177
9.74.2 Member Function Documentation . . . 177
9.74.2.1 operator()(const Ex &, const Ex &) . . . 177
9.74.3 Member Data Documentation . . . 177
9.74.3.1 mod_prel . . . 177
9.74.3.2 properties . . . 177
9.75 cadabra::exchange Class Reference . . . 177
9.75.1 Member Function Documentation . . . 178
9.75.1.1 collect_identical_tensors(const Properties &pr, Ex &tr, Ex::iterator it, std::vector< identical_tensors_t > &idts) . . . 178
9.75.1.2 get_node_gs(const Properties &, Ex &, Ex::iterator, std::vector< std::vector< int > > &) . . . 178
9.75.1.3 possible_singlets(Ex &, Ex::iterator) . . . 178
9.76 exec_stream_t Class Reference . . . 178
9.76.1 Member Typedef Documentation . . . 179
9.76.1.1 error_code_t . . . 179
9.76.1.2 timeout_t . . . 179
9.76.2 Member Enumeration Documentation . . . 179
9.76.2.1 stream_kind_t . . . 179
9.76.3 Constructor & Destructor Documentation . . . 180
9.76.3.1 exec_stream_t() . . . 180

9.76.3.2 exec_stream_t(std::string const &program, std::string const &arguments) 180
9.76.3.3 exec_stream_t(std::string const &program, iterator args_begin, iterator args_end) 180
9.76.3.4 ∼exec_stream_t() 180
9.76.3.5 exec_stream_t(exec_stream_t const &) 180
9.76.4 Member Function Documentation 180
9.76.4.1 close() 180
9.76.4.2 close_in() 180
9.76.4.3 err() 180
9.76.4.4 exceptions(bool enable) 180
9.76.4.5 exit_code() 180
9.76.4.6 in() 180
9.76.4.7 kill() 180
9.76.4.8 new_impl() 180
9.76.4.9 operator=(exec_stream_t const &) 180
9.76.4.10 out() 180
9.76.4.11 set_binary_mode(int stream_kind) 180
9.76.4.12 set_buffer_limit(int stream_kind, std::size_t size) 180
9.76.4.13 set_text_mode(int stream_kind) 180
9.76.4.14 set_wait_timeout(int stream_kind, timeout_t milliseconds) 180
9.76.4.15 start(std::string const &program, std::string const &arguments) 180
9.76.4.16 start(std::string const &program, iterator args_begin, iterator args_end) 181
9.76.4.17 start(std::string const &program, char const ∗arg1, char const ∗arg2) 181
9.76.4.18 start(std::string const &program, char ∗arg1, char ∗arg2) 181
9.76.4.19 start(std::string const &program, next_arg_t &next_arg) 181
9.76.5 Friends And Related Function Documentation 181
9.76.5.1 impl_t 181
9.76.6 Member Data Documentation 181
9.76.6.1 m_impl 181
9.77 cadabra::CodeInput::exp_input_tv Class Reference 181
9.77.1 Description 181

9.77.2 Constructor & Destructor Documentation . . . 182
9.77.2.1 exp_input_tv(DTree::iterator, Glib::RefPtr< Gtk::TextBuffer >, double scale) . . . 182
9.77.3 Member Function Documentation . . . 182
9.77.3.1 on_draw(const Cairo::RefPtr< Cairo::Context > &) override . . . 182
9.77.3.2 on_focus_in_event(GdkEventFocus *) override . . . 182
9.77.3.3 on_key_press_event(GdkEventKey *) override . . . 182
9.77.3.4 on_show() override . . . 182
9.77.3.5 shift_enter_pressed() . . . 182
9.77.4 Member Data Documentation . . . 182
9.77.4.1 cell_got_focus . . . 182
9.77.4.2 CodeInput . . . 182
9.77.4.3 content_changed . . . 182
9.77.4.4 content_erase . . . 182
9.77.4.5 content_execute . . . 182
9.77.4.6 content_insert . . . 182
9.77.4.7 datacell . . . 182
9.77.4.8 scale_ . . . 182
9.78 cadabra::expand Class Reference . . . 183
9.78.1 Constructor & Destructor Documentation . . . 183
9.78.1.1 expand(const Kernel &, Ex &) . . . 183
9.78.2 Member Function Documentation . . . 183
9.78.2.1 apply(iterator &) override . . . 183
9.78.2.2 can_apply(iterator) override . . . 183
9.78.3 Member Data Documentation . . . 184
9.78.3.1 ii_first . . . 184
9.78.3.2 ii_last . . . 184
9.78.3.3 mx_first . . . 184
9.78.3.4 mx_last . . . 184
9.78.3.5 one_index . . . 184
9.79 cadabra::expand_delta Class Reference . . . 184

9.79.1 Constructor & Destructor Documentation . . . 184
9.79.1.1 expand_delta(const Kernel &, Ex &) . . . 184
9.79.2 Member Function Documentation . . . 184
9.79.2.1 apply(iterator &) override . . . 184
9.79.2.2 can_apply(iterator) override . . . 185
9.80 cadabra::expand_diracbar Class Reference . . . 185
9.80.1 Constructor & Destructor Documentation . . . 185
9.80.1.1 expand_diracbar(const Kernel &, Ex &) . . . 185
9.80.2 Member Function Documentation . . . 185
9.80.2.1 apply(iterator &) override . . . 185
9.80.2.2 can_apply(iterator) override . . . 185
9.81 cadabra::expand_power Class Reference . . . 186
9.81.1 Constructor & Destructor Documentation . . . 186
9.81.1.1 expand_power(const Kernel &, Ex &) . . . 186
9.81.2 Member Function Documentation . . . 186
9.81.2.1 apply(iterator &) override . . . 186
9.81.2.2 can_apply(iterator) override . . . 186
9.82 cadabra::ExteriorDerivative Class Reference . . . 187
9.82.1 Member Function Documentation . . . 187
9.82.1.1 degree(const Properties &, Ex::iterator) const override . . . 187
9.82.1.2 name() const override . . . 187
9.83 cadabra::factor_in Class Reference . . . 188
9.83.1 Member Typedef Documentation . . . 188
9.83.1.1 term_hash_iterator_t . . . 188
9.83.1.2 term_hash_t . . . 188
9.83.2 Constructor & Destructor Documentation . . . 188
9.83.2.1 factor_in(const Kernel &, Ex &, Ex &) . . . 188
9.83.3 Member Function Documentation . . . 188
9.83.3.1 apply(iterator &) override . . . 188
9.83.3.2 calc_restricted_hash(iterator it) const . . . 189

9.83.3.3 can_apply(iterator) override . . . 189
9.83.3.4 compare_prod_nonprod(iterator prod, iterator nonprod) const . . . 189
9.83.3.5 compare_restricted(iterator one, iterator two) const . . . 189
9.83.3.6 fill_hash_map(iterator) . . . 189
9.83.4 Member Data Documentation . . . 189
9.83.4.1 factnodes . . . 189
9.83.4.2 factors . . . 189
9.83.4.3 term_hash . . . 189
9.84 cadabra::factor_out Class Reference . . . 189
9.84.1 Member Typedef Documentation . . . 190
9.84.1.1 to_factor_out_t . . . 190
9.84.2 Constructor & Destructor Documentation . . . 190
9.84.2.1 factor_out(const Kernel &, Ex &, Ex &, bool right) . . . 190
9.84.3 Member Function Documentation . . . 190
9.84.3.1 apply(iterator &) override . . . 190
9.84.3.2 can_apply(iterator) override . . . 190
9.84.4 Member Data Documentation . . . 190
9.84.4.1 to_factor_out . . . 190
9.84.4.2 to_right . . . 190
9.85 cadabra::fierz Class Reference . . . 191
9.85.1 Constructor & Destructor Documentation . . . 191
9.85.1.1 fierz(const Kernel &, Ex &, Ex &) . . . 191
9.85.2 Member Function Documentation . . . 191
9.85.2.1 apply(iterator &) override . . . 191
9.85.2.2 can_apply(iterator) override . . . 192
9.85.3 Member Data Documentation . . . 192
9.85.3.1 dim . . . 192
9.85.3.2 gam1 . . . 192
9.85.3.3 gam2 . . . 192
9.85.3.4 indprop . . . 192

9.85.3.5 prop1 . . . 192
9.85.3.6 prop2 . . . 192
9.85.3.7 prop3 . . . 192
9.85.3.8 prop4 . . . 192
9.85.3.9 spin1 . . . 192
9.85.3.10 spin2 . . . 192
9.85.3.11 spin3 . . . 192
9.85.3.12 spin4 . . . 192
9.85.3.13 spinor_list . . . 192
9.85.3.14 spinordim . . . 192
9.86 yngtab::filled_tableau< T > Class Template Reference . . . 193
9.86.1 Member Typedef Documentation . . . 194
9.86.1.1 box_row . . . 194
9.86.1.2 row_stack . . . 194
9.86.1.3 value_type . . . 194
9.86.2 Constructor & Destructor Documentation . . . 194
9.86.2.1 ~filled_tableau() . . . 194
9.86.3 Member Function Documentation . . . 194
9.86.3.1 add_box(unsigned int row) . . . 194
9.86.3.2 add_box(unsigned int rownum, T val) . . . 195
9.86.3.3 begin() const . . . 195
9.86.3.4 begin_column(unsigned int column_number) . . . 195
9.86.3.5 canonicalise() . . . 195
9.86.3.6 canonicalise(StrictWeakOrdering comp, bool only_col_ex=false) . . . 195
9.86.3.7 clear() . . . 195
9.86.3.8 compare_without_multiplicity(const filled_tableau< T > &other) const . . . 195
9.86.3.9 copy_shape(const tableau &) . . . 195
9.86.3.10 end() const . . . 195
9.86.3.11 end_column(unsigned int column_number) . . . 195
9.86.3.12 find(const T &) const . . . 195

9.86.3.13 Garnir_set(OutputIterator, unsigned int, unsigned int) const . . . 195
9.86.3.14 has_nullifying_trace() const . . . 195
9.86.3.15 nonstandard_loc() const . . . 195
9.86.3.16 number_of_rows() const . . . 195
9.86.3.17 operator()(unsigned int row, unsigned int col) . . . 196
9.86.3.18 operator()(unsigned int row, unsigned int col) const . . . 196
9.86.3.19 operator=(const filled_tableau< T > &) . . . 196
9.86.3.20 operator[](unsigned int boxnum) const . . . 196
9.86.3.21 projector(combin::symmetriser< T > &, bool modulo_monoterm=false) const . . . 196
9.86.3.22 projector(combin::symmetriser< T > &, combin::range_vector_t &) const . . . 196
9.86.3.23 projector_normalisation() const . . . 196
9.86.3.24 remove_box(unsigned int row) . . . 196
9.86.3.25 row_size(unsigned int row) const . . . 196
9.86.3.26 sort_columns() . . . 196
9.86.3.27 sort_columns(StrictWeakOrdering comp) . . . 196
9.86.3.28 sort_within_columns() . . . 196
9.86.3.29 sort_within_columns(StrictWeakOrdering comp) . . . 196
9.86.3.30 swap_columns(unsigned int c1, unsigned int c2) . . . 196
9.86.4 Member Data Documentation . . . 196
9.86.4.1 rows . . . 196
9.87 cadabra::FilledTableau Class Reference . . . 197
9.87.1 Constructor & Destructor Documentation . . . 197
9.87.1.1 ~FilledTableau() . . . 197
9.87.2 Member Function Documentation . . . 197
9.87.2.1 name() const . . . 197
9.87.2.2 parse(Ex &, Ex::iterator, Ex::iterator, keyval_t &) . . . 197
9.87.3 Member Data Documentation . . . 197
9.87.3.1 dimension . . . 197
9.88 cadabra::str_node::flag_t Struct Reference . . . 198
9.88.1 Member Data Documentation . . . 198

9.88.1.1 bracket . . . 198
9.88.1.2 keep_after_eval . . . 198
9.88.1.3 line_per_node . . . 198
9.88.1.4 parent_rel . . . 198
9.89 cadabra::flatten_product Class Reference . . . 198
9.89.1 Constructor & Destructor Documentation . . . 199
9.89.1.1 flatten_product(const Kernel &, Ex &) . . . 199
9.89.2 Member Function Documentation . . . 199
9.89.2.1 apply(iterator &) . . . 199
9.89.2.2 can_apply(iterator) . . . 199
9.89.3 Member Data Documentation . . . 199
9.89.3.1 is_diff . . . 199
9.89.3.2 make_consistent_only . . . 199
9.90 cadabra::flatten_sum Class Reference . . . 199
9.90.1 Constructor & Destructor Documentation . . . 200
9.90.1.1 flatten_sum(const Kernel &, Ex &) . . . 200
9.90.2 Member Function Documentation . . . 200
9.90.2.1 apply(iterator &) . . . 200
9.90.2.2 can_apply(iterator) . . . 200
9.90.3 Member Data Documentation . . . 200
9.90.3.1 make_consistent_only . . . 200
9.91 snoop::Flush Class Reference . . . 200
9.92 cadabra::GammaMatrix Class Reference . . . 201
9.92.1 Member Function Documentation . . . 201
9.92.1.1 latex(std::ostream &) const override . . . 201
9.92.1.2 name() const override . . . 201
9.92.1.3 parse(const Kernel &, keyval_t &keyvals) override . . . 201
9.92.2 Member Data Documentation . . . 202
9.92.2.1 metric . . . 202
9.93 cadabra::GammaTraceless Class Reference . . . 202

9.93.1 Constructor & Destructor Documentation . . . 202
9.93.1.1 ~GammaTraceless() . . . 202
9.93.2 Member Function Documentation . . . 202
9.93.2.1 name() const . . . 202
9.94 cadabra::GUIBase Class Reference . . . 203
9.94.1 Description . . . 203
9.94.2 Member Function Documentation . . . 203
9.94.2.1 add_cell(const DTree &, DTree::iterator, bool visible)=0 . . . 203
9.94.2.2 get_cursor_position(const DTree &, DTree::iterator)=0 . . . 204
9.94.2.3 on_connect()=0 . . . 204
9.94.2.4 on_disconnect(const std::string &reason)=0 . . . 204
9.94.2.5 on_kernel_runstatus(bool)=0 . . . 204
9.94.2.6 on_network_error()=0 . . . 204
9.94.2.7 position_cursor(const DTree &, DTree::iterator, int)=0 . . . 204
9.94.2.8 process_data()=0 . . . 204
9.94.2.9 remove_all_cells()=0 . . . 204
9.94.2.10 remove_cell(const DTree &, DTree::iterator)=0 . . . 205
9.94.2.11 update_cell(const DTree &, DTree::iterator)=0 . . . 205
9.95 cadabra::DataCell::id_t Class Reference . . . 205
9.95.1 Description . . . 205
9.95.2 Constructor & Destructor Documentation . . . 206
9.95.2.1 id_t() . . . 206
9.95.3 Member Function Documentation . . . 206
9.95.3.1 operator<(const id_t &other) const . . . 206
9.95.4 Member Data Documentation . . . 206
9.95.4.1 created_by_client . . . 206
9.95.4.2 id . . . 206
9.96 cadabra::exchange::identical_tensors_t Struct Reference . . . 206
9.96.1 Member Data Documentation . . . 207
9.96.1.1 comm . . . 207

9.96.1.2 extra_sign . . . 207
9.96.1.3 gammatraceless . . . 207
9.96.1.4 number_of_indices . . . 207
9.96.1.5 seq_numbers_of_first_indices . . . 207
9.96.1.6 spino . . . 207
9.96.1.7 tab . . . 207
9.96.1.8 tensors . . . 207
9.96.1.9 traceless . . . 207
9.97 Images Class Reference . . . 207
9.97.1 Member Function Documentation . . . 207
9.97.1.1 find(iterator start1, iterator end1, iterator start2, iterator end2) . . . 207
9.97.2 Member Data Documentation . . . 208
9.97.2.1 images . . . 208
9.98 cadabra::ImageView Class Reference . . . 208
9.98.1 Description . . . 208
9.98.2 Constructor & Destructor Documentation . . . 208
9.98.2.1 ImageView() . . . 208
9.98.2.2 ∼ImageView() . . . 208
9.98.3 Member Function Documentation . . . 208
9.98.3.1 set_image_from_base64(const std::string &b64) . . . 208
9.98.4 Member Data Documentation . . . 208
9.98.4.1 image . . . 208
9.99 cadabra::ImaginaryI Class Reference . . . 209
9.99.1 Member Function Documentation . . . 209
9.99.1.1 name() const override . . . 209
9.100 cadabra::ImplicitIndex Class Reference . . . 209
9.100.1 Member Function Documentation . . . 210
9.100.1.1 latex(std::ostream &str) const override . . . 210
9.100.1.2 name() const override . . . 210
9.100.1.3 parse(const Kernel &, keyval_t &) override . . . 210

9.100.1.4 unnamed_argument() const override . . . 210
9.100.2 Member Data Documentation . . . 211
9.100.2.1 set_names . . . 211
9.101 yngtab::filled_tableau< T >::in_column_iterator Class Reference . . . 211
9.101.1 Description . . . 211
9.101.2 Constructor & Destructor Documentation . . . 212
9.101.2.1 in_column_iterator(unsigned int r, unsigned int c, filled_tableau< T > *) . . . 212
9.101.3 Member Function Documentation . . . 212
9.101.3.1 operator"!=(const in_column_iterator &) const . . . 212
9.101.3.2 operator*() const . . . 212
9.101.3.3 operator+(unsigned int) . . . 212
9.101.3.4 operator++() . . . 212
9.101.3.5 operator++(int) . . . 212
9.101.3.6 operator+=(unsigned int) . . . 212
9.101.3.7 operator-(unsigned int) . . . 212
9.101.3.8 operator-(const in_column_iterator &) const . . . 212
9.101.3.9 operator--() . . . 212
9.101.3.10 operator--(int) . . . 212
9.101.3.11 operator-=(unsigned int) . . . 213
9.101.3.12 operator->() const . . . 213
9.101.3.13 operator<(const in_column_iterator &other) const . . . 213
9.101.3.14 operator<=(const in_column_iterator &other) const . . . 213
9.101.3.15 operator==(const in_column_iterator &) const . . . 213
9.101.3.16 operator>(const in_column_iterator &other) const . . . 213
9.101.3.17 operator>=(const in_column_iterator &other) const . . . 213
9.101.4 Friends And Related Function Documentation . . . 213
9.101.4.1 filled_tableau< T > . . . 213
9.101.5 Member Data Documentation . . . 213
9.101.5.1 column_number . . . 213
9.101.5.2 row_number . . . 213

9.101.5.3 tab . . . 213
9.102 cadabra::index_iterator Class Reference . . . 213
9.102.1 Description . . . 213
9.102.2 Constructor & Destructor Documentation . . . 214
9.102.2.1 index_iterator(const Properties &) . . . 214
9.102.2.2 index_iterator(const index_iterator &) . . . 214
9.102.3 Member Function Documentation . . . 214
9.102.3.1 begin(const Properties &, const iterator_base &, int offset=0) . . . 214
9.102.3.2 create(const Properties &, const iterator_base &) . . . 214
9.102.3.3 end(const Properties &, const iterator_base &) . . . 214
9.102.3.4 is_index(Ex::iterator) const . . . 215
9.102.3.5 operator"!=(const index_iterator &) const . . . 215
9.102.3.6 operator++() . . . 215
9.102.3.7 operator++(int) . . . 215
9.102.3.8 operator+=(unsigned int) . . . 215
9.102.3.9 operator=(const index_iterator &) . . . 215
9.102.3.10 operator==(const index_iterator &) const . . . 215
9.102.4 Member Data Documentation . . . 215
9.102.4.1 halt . . . 215
9.102.4.2 properties . . . 215
9.102.4.3 roof . . . 215
9.102.4.4 walk . . . 215
9.103 cadabra::IndexInherit Class Reference . . . 215
9.103.1 Member Function Documentation . . . 216
9.103.1.1 name() const . . . 216
9.104 cadabra::indexsort Class Reference . . . 216
9.104.1 Constructor & Destructor Documentation . . . 217
9.104.1.1 indexsort(const Kernel &, Ex &) . . . 217
9.104.2 Member Function Documentation . . . 217
9.104.2.1 apply(iterator &) . . . 217

9.104.2.2 can_apply(iterator) . . . 217
9.104.3 Member Data Documentation . . . 217
9.104.3.1 tb . . . 217
9.105cadabra::Indices Class Reference . . . 217
9.105.1 Member Enumeration Documentation . . . 218
9.105.1.1 position_t . . . 218
9.105.2 Constructor & Destructor Documentation . . . 218
9.105.2.1 Indices() . . . 218
9.105.3 Member Function Documentation . . . 218
9.105.3.1 collect_index_values(Ex::iterator ind_values) . . . 218
9.105.3.2 equals(const property ∗) const override . . . 218
9.105.3.3 latex(std::ostream &) const override . . . 219
9.105.3.4 name() const override . . . 219
9.105.3.5 parse(const Kernel &, keyval_t &) override . . . 219
9.105.3.6 unnamed_argument() const override . . . 219
9.105.4 Member Data Documentation . . . 219
9.105.4.1 parent_name . . . 219
9.105.4.2 position_type . . . 219
9.105.4.3 set_name . . . 219
9.105.4.4 values . . . 219
9.106cadabra::Inherit< T > Class Template Reference . . . 219
9.106.1 Description . . . 219
9.106.2 Constructor & Destructor Documentation . . . 220
9.106.2.1 ∼Inherit() . . . 220
9.106.3 Member Function Documentation . . . 220
9.106.3.1 name() const . . . 220
9.107cadabra::Integer Class Reference . . . 220
9.107.1 Description . . . 220
9.107.2 Constructor & Destructor Documentation . . . 221
9.107.2.1 ∼Integer() . . . 221

9.107.3 Member Function Documentation . . . 221
9.107.3.1 display(std::ostream &) const . . . 221
9.107.3.2 name() const override . . . 221
9.107.3.3 parse(const Kernel &, keyval_t &keyvals) override . . . 221
9.107.3.4 unnamed_argument() const override . . . 221
9.107.4 Member Data Documentation . . . 221
9.107.4.1 difference . . . 221
9.107.4.2 from . . . 221
9.107.4.3 to . . . 221
9.108 Integral Class Reference . . . 222
9.108.1 Constructor & Destructor Documentation . . . 222
9.108.1.1 ∼Integral() . . . 222
9.108.2 Member Function Documentation . . . 222
9.108.2.1 display(std::ostream &) const . . . 222
9.108.2.2 name() const . . . 222
9.108.2.3 parse(const Properties &, keyval_t &keyvals) override . . . 222
9.109 cadabra::integrate_by_parts Class Reference . . . 222
9.109.1 Description . . . 222
9.109.2 Constructor & Destructor Documentation . . . 223
9.109.2.1 integrate_by_parts(const Kernel &, Ex &, Ex &) . . . 223
9.109.3 Member Function Documentation . . . 223
9.109.3.1 apply(iterator &) override . . . 223
9.109.3.2 can_apply(iterator) override . . . 223
9.109.3.3 derivative_acting_on_arg(iterator der_it) const . . . 223
9.109.3.4 handle_term(iterator, iterator &) . . . 223
9.109.3.5 int_and_derivative_related(iterator int_it, iterator der_it) const . . . 223
9.109.3.6 wrap(iterator, sibling_iterator, sibling_iterator) const . . . 223
9.109.4 Member Data Documentation . . . 223
9.109.4.1 away_from . . . 223
9.110 InternalError Class Reference . . . 224

9.110.1 Description . . . 224
9.110.2 Constructor & Destructor Documentation . . . 224
9.110.2.1 InternalError(std::string="""") . . . 224
9.110.3 Member Function Documentation . . . 224
9.110.3.1 py_what() const . . . 224
9.111 InterruptionException Class Reference . . . 224
9.111.1 Constructor & Destructor Documentation . . . 225
9.111.1.1 InterruptionException(std::string="""") . . . 225
9.112 cadabra::InverseMetric Class Reference . . . 225
9.112.1 Constructor & Destructor Documentation . . . 225
9.112.1.1 InverseMetric() . . . 225
9.112.2 Member Function Documentation . . . 225
9.112.2.1 name() const override . . . 225
9.112.2.2 parse(const Kernel &, keyval_t &) override . . . 226
9.112.2.3 validate(const Kernel &, const Ex &) const override . . . 226
9.112.3 Member Data Documentation . . . 226
9.112.3.1 signature . . . 226
9.113 iterable_converter Class Reference . . . 226
9.113.1 Member Function Documentation . . . 226
9.113.1.1 construct(PyObject *object, boost::python::converter::rvalue_from_python_↩ stage1_data *data) . . . 226
9.113.1.2 convertible(PyObject *object) . . . 226
9.113.1.3 from_python() . . . 226
9.114 yngtab::filled_tableau< T >::iterator Class Reference . . . 227
9.114.1 Description . . . 227
9.114.2 Constructor & Destructor Documentation . . . 228
9.114.2.1 iterator(unsigned int r, unsigned int c, filled_tableau< T > *) . . . 228
9.114.3 Member Function Documentation . . . 228
9.114.3.1 operator"!=(const iterator &) const . . . 228
9.114.3.2 operator*() const . . . 228
9.114.3.3 operator+(unsigned int) . . . 228

9.114.3.4 operator++() . . . 228
9.114.3.5 operator++(int) . . . 228
9.114.3.6 operator+=(unsigned int) . . . 228
9.114.3.7 operator-(unsigned int) . . . 228
9.114.3.8 operator-(const iterator &) const . . . 228
9.114.3.9 operator--() . . . 228
9.114.3.10 operator--(int) . . . 228
9.114.3.11 operator-=(unsigned int) . . . 228
9.114.3.12 operator->() const . . . 228
9.114.3.13 operator<(const iterator &other) const . . . 228
9.114.3.14 operator==(const iterator &) const . . . 228
9.114.3.15 operator>(const iterator &other) const . . . 228
9.114.4 Friends And Related Function Documentation . . . 228
9.114.4.1 filled_tableau< T > . . . 228
9.114.5 Member Data Documentation . . . 228
9.114.5.1 column_number . . . 228
9.114.5.2 row_number . . . 228
9.114.5.3 tab . . . 228
9.115 yngtab::filled_tableau< T >::iterator_base Class Reference . . . 229
9.115.1 Member Typedef Documentation . . . 229
9.115.1.1 difference_type . . . 229
9.115.1.2 iterator_category . . . 229
9.115.1.3 pointer . . . 229
9.115.1.4 reference . . . 229
9.115.1.5 size_type . . . 229
9.115.1.6 value_type . . . 229
9.116 cadabra::join_gamma Class Reference . . . 229
9.116.1 Constructor & Destructor Documentation . . . 230
9.116.1.1 join_gamma(const Kernel &, Ex &, bool expand, bool use_gendelta) . . . 230
9.116.2 Member Function Documentation . . . 230

9.116.2.1 append_prod_(const std::vector< Ex > &r1, const std::vector< Ex > &r2, unsigned int num1, unsigned int num2, unsigned int i, multiplier_t mult, Ex &rep, iterator loc) . . . 230
9.116.2.2 apply(iterator &) . . . 230
9.116.2.3 can_apply(iterator) . . . 230
9.116.2.4 regroup_indices_(sibling_iterator, sibling_iterator, unsigned int, std::vector< Ex > &, std::vector< Ex > &) . . . 231
9.116.3 Member Data Documentation . . . 231
9.116.3.1 expand . . . 231
9.116.3.2 gamma_bracket_ . . . 231
9.116.3.3 gamma_name_ . . . 231
9.116.3.4 gm1 . . . 231
9.116.3.5 gm2 . . . 231
9.116.3.6 only_expand . . . 231
9.116.3.7 use_generalised_delta_ . . . 231
9.117 cadabra::keep_terms Class Reference . . . 231
9.117.1 Constructor & Destructor Documentation . . . 232
9.117.1.1 keep_terms(const Kernel &, Ex &, std::vector< int > terms) . . . 232
9.117.2 Member Function Documentation . . . 232
9.117.2.1 apply(iterator &) . . . 232
9.117.2.2 can_apply(iterator) . . . 232
9.117.3 Member Data Documentation . . . 232
9.117.3.1 terms_ . . . 232
9.118 cadabra::keep_weight Class Reference . . . 232
9.118.1 Constructor & Destructor Documentation . . . 233
9.118.1.1 keep_weight(const Kernel &, Ex &, Ex &) . . . 233
9.118.2 Member Function Documentation . . . 233
9.118.2.1 apply(iterator &) override . . . 233
9.119 cadabra::Kernel Class Reference . . . 233
9.119.1 Constructor & Destructor Documentation . . . 233
9.119.1.1 Kernel() . . . 233
9.119.1.2 Kernel(const Kernel &other)=delete . . . 233

9.119.1.3 ~Kernel() . . . 233
9.119.2 Member Function Documentation . . . 233
9.119.2.1 ex_from_string(const std::string &) . . . 233
9.119.2.2 inject_property(property *prop, std::shared_ptr< Ex > pattern, std::shared_ptr< Ex > property_arguments) . . . 234
9.119.3 Member Data Documentation . . . 234
9.119.3.1 properties . . . 234
9.120 cadabra::keyval_t Class Reference . . . 234
9.120.1 Description . . . 234
9.120.2 Member Typedef Documentation . . . 235
9.120.2.1 const_iterator . . . 235
9.120.2.2 iterator . . . 235
9.120.2.3 kvlist_t . . . 235
9.120.2.4 kvpair_t . . . 235
9.120.2.5 value_type . . . 235
9.120.3 Member Function Documentation . . . 235
9.120.3.1 begin() const . . . 235
9.120.3.2 end() const . . . 235
9.120.3.3 erase(iterator) . . . 235
9.120.3.4 find(const std::string &) const . . . 235
9.120.3.5 find(const std::string &) . . . 235
9.120.3.6 push_back(const kvpair_t &) . . . 235
9.120.4 Member Data Documentation . . . 235
9.120.4.1 keyvals . . . 235
9.121 cadabra::KroneckerDelta Class Reference . . . 235
9.121.1 Constructor & Destructor Documentation . . . 236
9.121.1.1 ~KroneckerDelta() . . . 236
9.121.2 Member Function Documentation . . . 236
9.121.2.1 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const . . . 236
9.121.2.2 name() const . . . 236
9.121.2.3 size(const Properties &, Ex &, Ex::iterator) const . . . 236

9.122cadabra::labelled_property Class Reference . . . 236
9.122.1 Member Function Documentation . . . 237
9.122.1.1 parse(const Kernel &, keyval_t &) override . . . 237
9.122.2 Member Data Documentation . . . 237
9.122.2.1 label . . . 237
9.123cadabra::LaTeXForm Class Reference . . . 237
9.123.1 Member Function Documentation . . . 238
9.123.1.1 latex_form() const . . . 238
9.123.1.2 name() const override . . . 238
9.123.1.3 parse(const Kernel &, keyval_t &) override . . . 238
9.123.1.4 unnamed_argument() const override . . . 238
9.123.2 Member Data Documentation . . . 238
9.123.2.1 latex_ . . . 238
9.124cadabra::indexsort::less_indexed_treenode Class Reference . . . 238
9.124.1 Constructor & Destructor Documentation . . . 239
9.124.1.1 less_indexed_treenode(const Kernel &, Ex &, iterator it) . . . 239
9.124.2 Member Function Documentation . . . 239
9.124.2.1 operator()(unsigned int, unsigned int) const . . . 239
9.124.3 Member Data Documentation . . . 239
9.124.3.1 it . . . 239
9.124.3.2 kernel . . . 239
9.124.3.3 tr . . . 239
9.125cadabra::list_property Class Reference . . . 239
9.125.1 Description . . . 239
9.126snoop::Snoop::LogEntry Class Reference . . . 240
9.126.1 Description . . . 240
9.126.2 Constructor & Destructor Documentation . . . 241
9.126.2.1 LogEntry() . . . 241
9.126.2.2 LogEntry(int log_id_, int client_log_id_, int id_, const std::string &, uint64_t, uint64_t, const std::string &, int, const std::string &, const std::string &, const std::string &, int status) . . . 241

9.126.3 Member Function Documentation . . . 241
9.126.3.1 from_json(const Json::Value &) . . . 241
9.126.3.2 to_json(bool human_readable) const . . . 241
9.126.4 Member Data Documentation . . . 241
9.126.4.1 client_log_id . . . 241
9.126.4.2 create_millis . . . 241
9.126.4.3 id . . . 241
9.126.4.4 loc_file . . . 241
9.126.4.5 loc_line . . . 241
9.126.4.6 loc_method . . . 241
9.126.4.7 log_id . . . 241
9.126.4.8 message . . . 241
9.126.4.9 receive_millis . . . 241
9.126.4.10 server_status . . . 241
9.126.4.11 type . . . 241
9.126.4.12 uuid . . . 241
9.127 cadabra::lr_tensor Class Reference . . . 242
9.127.1 Constructor & Destructor Documentation . . . 242
9.127.1.1 lr_tensor(const Kernel &, Ex &) . . . 242
9.127.2 Member Function Documentation . . . 242
9.127.2.1 apply(iterator &) override . . . 242
9.127.2.2 can_apply(iterator) override . . . 242
9.127.2.3 do_filledtableau(iterator &) . . . 243
9.127.2.4 do_tableau(iterator &) . . . 243
9.127.3 Member Data Documentation . . . 243
9.127.3.1 tab1 . . . 243
9.127.3.2 tab2 . . . 243
9.128 cadabra::map_sympy Class Reference . . . 243
9.128.1 Description . . . 243
9.128.2 Constructor & Destructor Documentation . . . 244

9.128.2.1 map_sympy(const Kernel &, Ex &, const std::string &head) . . . 244
9.128.3 Member Function Documentation . . . 244
9.128.3.1 apply(iterator &) . . . 244
9.128.3.2 can_apply(iterator) . . . 244
9.128.4 Member Data Documentation . . . 244
9.128.4.1 head_ . . . 244
9.128.4.2 index_factors . . . 244
9.128.4.3 left . . . 244
9.129 cadabra::Matrix Class Reference . . . 244
9.129.1 Constructor & Destructor Documentation . . . 245
9.129.1.1 ~Matrix() . . . 245
9.129.2 Member Function Documentation . . . 245
9.129.2.1 name() const . . . 245
9.130 cadabra::Metric Class Reference . . . 245
9.130.1 Constructor & Destructor Documentation . . . 246
9.130.1.1 Metric() . . . 246
9.130.2 Member Function Documentation . . . 246
9.130.2.1 latex(std::ostream &) const override . . . 246
9.130.2.2 name() const override . . . 246
9.130.2.3 parse(const Kernel &, keyval_t &) override . . . 246
9.130.2.4 validate(const Kernel &, const Ex &) const override . . . 246
9.130.3 Member Data Documentation . . . 246
9.130.3.1 signature . . . 246
9.131 MyClient Class Reference . . . 247
9.131.1 Constructor & Destructor Documentation . . . 247
9.131.1.1 MyClient() . . . 247
9.131.2 Member Function Documentation . . . 247
9.131.2.1 after_tree_change(ActionBase &) . . . 247
9.131.2.2 before_tree_change(ActionBase &) . . . 247
9.131.2.3 on_connect() . . . 247

9.131.2.4 on_disconnect() . . . 247
9.131.2.5 on_network_error() . . . 247
9.131.2.6 on_progress() . . . 247
9.132 exec_stream_t::next_arg_impl_t< iterator > Class Template Reference . . . 248
9.132.1 Constructor & Destructor Documentation . . . 248
9.132.1.1 next_arg_impl_t(iterator args_begin, iterator args_end) . . . 248
9.132.2 Member Function Documentation . . . 248
9.132.2.1 next() . . . 248
9.132.3 Member Data Documentation . . . 248
9.132.3.1 m_arg . . . 248
9.132.3.2 m_args_end . . . 248
9.132.3.3 m_args_i . . . 248
9.133 exec_stream_t::next_arg_t Class Reference . . . 249
9.133.1 Constructor & Destructor Documentation . . . 249
9.133.1.1 ~next_arg_t() . . . 249
9.133.2 Member Function Documentation . . . 249
9.133.2.1 next()=0 . . . 249
9.134 cadabra::NonCommuting Class Reference . . . 249
9.134.1 Member Function Documentation . . . 250
9.134.1.1 name() const . . . 250
9.134.1.2 sign() const . . . 250
9.135 NonScalarException Class Reference . . . 250
9.135.1 Description . . . 250
9.135.2 Constructor & Destructor Documentation . . . 251
9.135.2.1 NonScalarException(std::string="") . . . 251
9.135.3 Member Function Documentation . . . 251
9.135.3.1 py_what() const . . . 251
9.136 cadabra::NotebookCanvas Class Reference . . . 251
9.136.1 Constructor & Destructor Documentation . . . 252
9.136.1.1 NotebookCanvas() . . . 252

9.136.1.2 ~NotebookCanvas() . . . 252
9.136.2 Member Function Documentation . . . 252
9.136.2.1 refresh_all() . . . 252
9.136.3 Member Data Documentation . . . 252
9.136.3.1 bottomline . . . 252
9.136.3.2 ebox . . . 252
9.136.3.3 scroll . . . 252
9.136.3.4 visualcells . . . 252
9.136.3.5 visualcells . . . 252
9.137 NotebookController Class Reference . . . 252
9.137.1 Property Documentation . . . 253
9.137.1.1 scrollbox . . . 253
9.137.1.2 status_label . . . 253
9.138 NotebookController() Category Reference . . . 253
9.139 cadabra_osx::NotebookWindow Class Reference . . . 253
9.139.1 Description . . . 253
9.139.2 Constructor & Destructor Documentation . . . 254
9.139.2.1 NotebookWindow(NotebookController *) . . . 254
9.139.3 Member Function Documentation . . . 254
9.139.3.1 add_cell(const DTree &, DTree::iterator, bool visible) override . . . 254
9.139.3.2 on_connect() override . . . 254
9.139.3.3 on_disconnect() override . . . 254
9.139.3.4 on_network_error() override . . . 254
9.139.3.5 position_cursor(const DTree &, DTree::iterator) override . . . 254
9.139.3.6 process_data() override . . . 254
9.139.3.7 remove_all_cells() override . . . 254
9.139.3.8 remove_cell(const DTree &, DTree::iterator) override . . . 254
9.139.3.9 update_cell(const DTree &, DTree::iterator) override . . . 254
9.139.4 Member Data Documentation . . . 254
9.139.4.1 canvasses . . . 254

9.139.4.2 controller . . . 254
9.139.4.3 current_canvas . . . 254
9.140 cadabra::NotebookWindow Class Reference . . . 255
9.140.1 Description . . . 255
9.140.2 Constructor & Destructor Documentation . . . 257
9.140.2.1 NotebookWindow(Cadabra *, bool read_only=false) . . . 257
9.140.2.2 ~NotebookWindow() . . . 257
9.140.3 Member Function Documentation . . . 258
9.140.3.1 add_cell(const DTree &, DTree::iterator, bool) override . . . 258
9.140.3.2 cell_content_changed(const std::string &content, DTree::iterator it, int canvas_number) . . . 258
9.140.3.3 cell_content_erase(int, int, DTree::iterator, int) . . . 258
9.140.3.4 cell_content_execute(DTree::iterator, int, bool shift_enter_pressed) . . . 258
9.140.3.5 cell_content_insert(const std::string &, int, DTree::iterator, int) . . . 258
9.140.3.6 cell_got_focus(DTree::iterator, int) . . . 258
9.140.3.7 cell_toggle_visibility(DTree::iterator it, int) . . . 258
9.140.3.8 dim_output_cells(DTree::iterator it) . . . 258
9.140.3.9 get_cursor_position(const DTree &, DTree::iterator) override . . . 258
9.140.3.10 idle_handler() . . . 258
9.140.3.11 load_file(const std::string &notebook_contents) . . . 258
9.140.3.12 on_configure_event(GdkEventConfigure *cfg) override . . . 258
9.140.3.13 on_connect() override . . . 258
9.140.3.14 on_crash_window_closed(int) . . . 259
9.140.3.15 on_delete_event(GdkEventAny *) override . . . 259
9.140.3.16 on_disconnect(const std::string &) override . . . 259
9.140.3.17 on_edit_cell_is_latex() . . . 259
9.140.3.18 on_edit_cell_is_python() . . . 259
9.140.3.19 on_edit_delete() . . . 259
9.140.3.20 on_edit_insert_above() . . . 259
9.140.3.21 on_edit_insert_below() . . . 259
9.140.3.22 on_edit_split() . . . 259

9.140.3.23on_edit_undo() . . . 259
9.140.3.24on_file_close() . . . 259
9.140.3.25on_file_export_html() . . . 259
9.140.3.26on_file_export_html_segment() . . . 259
9.140.3.27on_file_export_latex() . . . 259
9.140.3.28on_file_export_python() . . . 259
9.140.3.29on_file_new() . . . 259
9.140.3.30on_file_open() . . . 259
9.140.3.31on_file_quit() . . . 259
9.140.3.32on_file_save() . . . 259
9.140.3.33on_file_save_as() . . . 259
9.140.3.34on_help() const . . . 259
9.140.3.35on_help_about() . . . 259
9.140.3.36on_kernel_restart() . . . 259
9.140.3.37on_kernel_runstatus(bool) override . . . 259
9.140.3.38on_key_press_event(GdkEventKey ∗) override . . . 260
9.140.3.39on_network_error() override . . . 260
9.140.3.40on_prefs_font_size(int num) . . . 260
9.140.3.41on_run_cell() . . . 260
9.140.3.42on_run_runall() . . . 260
9.140.3.43on_run_runtocursor() . . . 260
9.140.3.44on_run_stop() . . . 260
9.140.3.45on_scroll(GdkEventScroll ∗) . . . 260
9.140.3.46on_scroll_size_allocate(Gtk::Allocation &) . . . 260
9.140.3.47on_tex_error(const std::string &, DTree::iterator) . . . 260
9.140.3.48on_text_scaling_factor_changed(const std::string &key) . . . 260
9.140.3.49on_view_close() . . . 260
9.140.3.50on_view_split() . . . 260
9.140.3.51on_vscroll_changed(Gtk::ScrollType, double) . . . 260
9.140.3.52position_cursor(const DTree &, DTree::iterator, int pos) override . . . 260

9.140.3.53 process_data() override . . . 260
9.140.3.54 process_todo_queue() . . . 261
9.140.3.55 quit_safeguard(bool quit) . . . 261
9.140.3.56 remove_all_cells() override . . . 261
9.140.3.57 remove_cell(const DTree &, DTree::iterator) override . . . 261
9.140.3.58 save(const std::string &fn) const . . . 261
9.140.3.59 scroll_current_cell_into_view() . . . 261
9.140.3.60 set_name(const std::string &) . . . 261
9.140.3.61 set_stop_sensitive(bool) . . . 261
9.140.3.62 set_title_prefix(const std::string &) . . . 261
9.140.3.63 setup_css_provider() . . . 261
9.140.3.64 update_cell(const DTree &, DTree::iterator) override . . . 261
9.140.3.65 update_title() . . . 262
9.140.4 Member Data Documentation . . . 262
9.140.4.1 actiongroup . . . 262
9.140.4.2 canvasses . . . 262
9.140.4.3 cdbapp . . . 262
9.140.4.4 crash_window_hidden . . . 262
9.140.4.5 css_provider . . . 262
9.140.4.6 current_canvas . . . 262
9.140.4.7 current_cell . . . 262
9.140.4.8 dispatcher . . . 262
9.140.4.9 engine . . . 262
9.140.4.10 follow_cell . . . 262
9.140.4.11 is_configured . . . 262
9.140.4.12 kernel_label . . . 262
9.140.4.13 kernel_spinner . . . 262
9.140.4.14 kernel_spinner_status . . . 262
9.140.4.15 kernel_string . . . 262
9.140.4.16 last_configure_width . . . 262

9.140.4.17 mainbox . . . 262
9.140.4.18 modified . . . 262
9.140.4.19 name . . . 262
9.140.4.20 prefs . . . 262
9.140.4.21 progressbar . . . 262
9.140.4.22 read_only . . . 263
9.140.4.23 scale . . . 263
9.140.4.24 settings . . . 263
9.140.4.25 status_label . . . 263
9.140.4.26 status_mutex . . . 263
9.140.4.27 status_string . . . 263
9.140.4.28 statusbarbox . . . 263
9.140.4.29 supermainbox . . . 263
9.140.4.30 title_prefix . . . 263
9.140.4.31 topbox . . . 263
9.140.4.32 uimanager . . . 263
9.141 NotYetImplemented Class Reference . . . 263
9.141.1 Description . . . 263
9.141.2 Constructor & Destructor Documentation . . . 264
9.141.2.1 NotYetImplemented(std::string="") . . . 264
9.141.3 Member Function Documentation . . . 264
9.141.3.1 py_what() const . . . 264
9.142 cadabra::nset_it_less Class Reference . . . 264
9.142.1 Description . . . 264
9.142.2 Member Function Documentation . . . 264
9.142.2.1 operator()(nset_t::iterator first, nset_t::iterator second) const . . . 264
9.143 cadabra::NumericalFlat Class Reference . . . 265
9.143.1 Description . . . 265
9.143.2 Member Function Documentation . . . 265
9.143.2.1 name() const . . . 265

9.144 cadabra::order Class Reference . . . 265
9.144.1 Constructor & Destructor Documentation . . . 266
9.144.1.1 order(const Kernel &, Ex &, Ex &objs, bool ac) . . . 266
9.144.2 Member Function Documentation . . . 266
9.144.2.1 apply(iterator &) override . . . 266
9.144.2.2 can_apply(iterator) override . . . 266
9.144.3 Member Data Documentation . . . 266
9.144.3.1 anticomm . . . 266
9.144.3.2 objects . . . 266
9.145 ParseException Class Reference . . . 267
9.145.1 Constructor & Destructor Documentation . . . 267
9.145.1.1 ParseException(std::string) . . . 267
9.146 cadabra::Parser Class Reference . . . 267
9.146.1 Member Enumeration Documentation . . . 268
9.146.1.1 mode_t . . . 268
9.146.2 Constructor & Destructor Documentation . . . 269
9.146.2.1 Parser() . . . 269
9.146.2.2 Parser(std::shared_ptr< Ex >) . . . 269
9.146.2.3 Parser(std::shared_ptr< Ex >, const std::string &) . . . 269
9.146.3 Member Function Documentation . . . 269
9.146.3.1 advance(unsigned int &i) . . . 269
9.146.3.2 erase() . . . 269
9.146.3.3 finalise() . . . 269
9.146.3.4 get_token(unsigned int i) . . . 269
9.146.3.5 is_closing_bracket(const unsigned char &br) const . . . 269
9.146.3.6 is_link(const unsigned char &ln) const . . . 269
9.146.3.7 is_number(const std::string &str) const . . . 269
9.146.3.8 is_opening_bracket(const unsigned char &br) const . . . 269
9.146.3.9 remove_empty_nodes() . . . 269
9.146.3.10 string2tree(const std::string &inp) . . . 269

9.146.4 Member Data Documentation . . . 269
9.146.4.1 current_bracket . . . 269
9.146.4.2 current_mode . . . 269
9.146.4.3 current_parent_rel . . . 269
9.146.4.4 parts . . . 269
9.146.4.5 str . . . 269
9.146.4.6 tree . . . 269
9.147 cadabra::PartialDerivative Class Reference . . . 270
9.147.1 Constructor & Destructor Documentation . . . 270
9.147.1.1 ∼PartialDerivative() . . . 270
9.147.2 Member Function Documentation . . . 270
9.147.2.1 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const . . . 270
9.147.2.2 name() const . . . 271
9.147.2.3 size(const Properties &, Ex &, Ex::iterator) const . . . 271
9.148 cadabra::pattern Class Reference . . . 271
9.148.1 Constructor & Destructor Documentation . . . 271
9.148.1.1 pattern() . . . 271
9.148.1.2 pattern(const Ex &) . . . 271
9.148.2 Member Function Documentation . . . 271
9.148.2.1 children_wildcard() const . . . 271
9.148.2.2 match(const Properties &, const Ex::iterator &, bool ignore_parent_rel=false) const . . . 271
9.148.3 Member Data Documentation . . . 271
9.148.3.1 obj . . . 271
9.149 Perm Class Reference . . . 272
9.149.1 Member Function Documentation . . . 272
9.149.1.1 apply(iterator start1, iterator end1) . . . 272
9.149.1.2 find(iterator start1, iterator end1, iterator start2, iterator end2) . . . 272
9.149.2 Member Data Documentation . . . 272
9.149.2.1 perm . . . 272
9.150 PermutationException Class Reference . . . 273

9.150.1 Description . . . 273
9.150.2 Constructor & Destructor Documentation . . . 273
9.150.2.1 PermutationException(const std::string &ex) . . . 273
9.151 PreferencesWindow Class Reference . . . 273
9.151.1 Member Data Documentation . . . 274
9.151.1.1 mainbox . . . 274
9.152 cadabra::NotebookWindow::Prefs Class Reference . . . 274
9.152.1 Constructor & Destructor Documentation . . . 274
9.152.1.1 Prefs() . . . 274
9.152.2 Member Data Documentation . . . 274
9.152.2.1 font_step . . . 274
9.153 preprocessor Class Reference . . . 274
9.153.1 Description . . . 274
9.153.2 Member Enumeration Documentation . . . 276
9.153.2.1 order_labels . . . 276
9.153.3 Constructor & Destructor Documentation . . . 276
9.153.3.1 preprocessor() . . . 276
9.153.4 Member Function Documentation . . . 276
9.153.4.1 bracket_strings_(unsigned int cb, std::string &obrack, std::string &cbrack) const 276
9.153.4.2 current_bracket_(bool deep=false) const . . . 276
9.153.4.3 default_is_product_() const . . . 277
9.153.4.4 erase() . . . 277
9.153.4.5 get_token_(unsigned char prev_token) . . . 277
9.153.4.6 is_already_bracketed_(const std::string &str) const . . . 277
9.153.4.7 is_bracket_(unsigned char c) const . . . 277
9.153.4.8 is_closing_bracket_(unsigned char c) const . . . 277
9.153.4.9 is_digits_(const std::string &str) const . . . 277
9.153.4.10 is_infix_operator_(unsigned char c) const . . . 277
9.153.4.11 is_link_(unsigned char c) const . . . 277
9.153.4.12 is_opening_bracket_(unsigned char c) const . . . 277

9.153.4.13 parse_(const std::string &) . . . 277
9.153.4.14 parse_internal_() . . . 277
9.153.4.15 print_stack() const . . . 277
9.153.4.16 show_and_throw_(const std::string &str) const . . . 277
9.153.4.17 strip_outer_brackets() const . . . 277
9.153.4.18 unwind_(unsigned int tolevel, unsigned int bracketgoal=0, bool usebracket=true) const . . . 277
9.153.5 Friends And Related Function Documentation . . . 277
9.153.5.1 operator<< . . . 277
9.153.5.2 operator>> . . . 277
9.153.6 Member Data Documentation . . . 277
9.153.6.1 accus . . . 277
9.153.6.2 close_brackets . . . 277
9.153.6.3 cur . . . 278
9.153.6.4 cur_pos . . . 278
9.153.6.5 cur_str . . . 278
9.153.6.6 eat_initial_whitespace_ . . . 278
9.153.6.7 next_is_product_ . . . 278
9.153.6.8 open_brackets . . . 278
9.153.6.9 order_names . . . 278
9.153.6.10 orders . . . 278
9.153.6.11 verbatim_ . . . 278
9.154 cadabra::product_rule Class Reference . . . 278
9.154.1 Constructor & Destructor Documentation . . . 279
9.154.1.1 product_rule(const Kernel &, Ex &) . . . 279
9.154.2 Member Function Documentation . . . 279
9.154.2.1 apply(iterator &) override . . . 279
9.154.2.2 can_apply(iterator) override . . . 279
9.154.3 Member Data Documentation . . . 279
9.154.3.1 number_of_indices . . . 279
9.154.3.2 prodnode . . . 279

9.155 ProgressMonitor Class Reference . . . 279
9.155.1 Constructor & Destructor Documentation . . . 280
9.155.1.1 ProgressMonitor() . . . 280
9.155.1.2 ~ProgressMonitor() . . . 280
9.155.2 Member Function Documentation . . . 280
9.155.2.1 group(std::string name="") . . . 280
9.155.2.2 print() const . . . 280
9.155.2.3 progress(int n, int total) . . . 280
9.155.2.4 totals() const . . . 280
9.155.3 Member Data Documentation . . . 280
9.155.3.1 call_stack . . . 280
9.155.3.2 call_totals . . . 280
9.156 cadabra::Properties Class Reference . . . 281
9.156.1 Description . . . 281
9.156.2 Member Typedef Documentation . . . 282
9.156.2.1 pat_prop_pair_t . . . 282
9.156.2.2 pattern_map_t . . . 282
9.156.2.3 property_map_t . . . 282
9.156.3 Member Function Documentation . . . 282
9.156.3.1 clear() . . . 282
9.156.3.2 get(Ex::iterator, bool ignore_parent_rel=false) const . . . 282
9.156.3.3 get() const . . . 282
9.156.3.4 get_composite(Ex::iterator, bool ignore_parent_rel=false) const . . . 282
9.156.3.5 get_composite(Ex::iterator, int &serialnum, bool doserial=true, bool ignore_↩ parent_rel=false) const . . . 282
9.156.3.6 get_composite(Ex::iterator, const std::string &label) const . . . 282
9.156.3.7 get_composite(Ex::iterator, int &serialnum, const std::string &label, bool doserial=true) const . . . 282
9.156.3.8 get_composite(Ex::iterator, Ex::iterator, bool ignore_parent_rel=false) const . . . 283
9.156.3.9 get_composite(Ex::iterator, Ex::iterator, int &, int &, bool ignore_parent_rel=false) const . . . 283
9.156.3.10 has(const property *, Ex::iterator) . . . 283

9.156.3.11head(Ex::iterator, bool ignore_parent_rel=false) const . . . 283
9.156.3.12insert_list_prop(const std::vector< Ex > &, const list_property *) . . . 283
9.156.3.13insert_prop(const Ex &, const property *) . . . 283
9.156.3.14master_insert(Ex proptree, property *thepropbase) . . . 283
9.156.3.15register_property(property *(*)(), const std::string &name) . . . 283
9.156.3.16serial_number(const property *, const pattern *) const . . . 283
9.156.4 Member Data Documentation . . . 283
9.156.4.1 pats . . . 283
9.156.4.2 props . . . 283
9.156.4.3 registered_properties . . . 283
9.157Property< T > Class Template Reference . . . 284
9.157.1 Description . . . 284
9.157.2 Constructor & Destructor Documentation . . . 285
9.157.2.1 Property(std::shared_ptr< cadabra::Ex > obj, std::shared_ptr< cadabra::Ex > params=0) . . . 285
9.157.3 Member Function Documentation . . . 285
9.157.3.1 latex_() const . . . 285
9.157.3.2 latex_() const . . . 285
9.157.3.3 repr_() const . . . 285
9.157.3.4 str_() const . . . 285
9.157.4 Member Data Documentation . . . 285
9.157.4.1 for_obj . . . 285
9.157.4.2 prop . . . 285
9.158cadabra::property Class Reference . . . 286
9.158.1 Description . . . 286
9.158.2 Member Enumeration Documentation . . . 287
9.158.2.1 match_t . . . 287
9.158.3 Constructor & Destructor Documentation . . . 287
9.158.3.1 property(bool hidden=false) . . . 287
9.158.3.2 ~property() . . . 287
9.158.4 Member Function Documentation . . . 287

9.158.4.1 equals(const property ∗) const . . . 287
9.158.4.2 hidden(bool h) . . . 288
9.158.4.3 hidden(void) const . . . 288
9.158.4.4 latex(std::ostream &) const . . . 288
9.158.4.5 name() const =0 . . . 288
9.158.4.6 parse(const Kernel &, keyval_t &keyvals) . . . 288
9.158.4.7 parse_one_argument(Ex::iterator arg, keyval_t &keyvals) . . . 288
9.158.4.8 parse_to_keyvals(const Ex &, keyval_t &) . . . 288
9.158.4.9 unnamed_argument() const . . . 288
9.158.4.10 validate(const Kernel &, const Ex &) const . . . 289
9.158.5 Member Data Documentation . . . 289
9.158.5.1 hidden_ . . . 289
9.159 cadabra::PropertyInherit Class Reference . . . 289
9.159.1 Description . . . 289
9.159.2 Member Function Documentation . . . 289
9.159.2.1 name() const . . . 289
9.160 cadabra::reduce_delta Class Reference . . . 290
9.160.1 Constructor & Destructor Documentation . . . 290
9.160.1.1 reduce_delta(const Kernel &, Ex &) . . . 290
9.160.2 Member Function Documentation . . . 290
9.160.2.1 apply(iterator &) override . . . 290
9.160.2.2 can_apply(iterator) override . . . 290
9.160.2.3 one_step_(sibling_iterator) . . . 290
9.161 cadabra::Properties::registered_property_map_t Class Reference . . . 291
9.161.1 Member Typedef Documentation . . . 291
9.161.1.1 internal_property_map_t . . . 291
9.161.1.2 iterator . . . 291
9.161.2 Constructor & Destructor Documentation . . . 291
9.161.2.1 ∼registered_property_map_t() . . . 291
9.161.3 Member Data Documentation . . . 291

9.161.3.1 store . . . 291
9.162rename_dummies Class Reference . . . 291
9.162.1 Constructor & Destructor Documentation . . . 292
9.162.1.1 rename_dummies(Ex &, iterator) . . . 292
9.162.2 Member Function Documentation . . . 292
9.162.2.1 apply(iterator &) . . . 292
9.162.2.2 can_apply(iterator) . . . 292
9.163cadabra::rename_dummies Class Reference . . . 292
9.163.1 Constructor & Destructor Documentation . . . 293
9.163.1.1 rename_dummies(const Kernel &, Ex &, std::string, std::string) . . . 293
9.163.2 Member Function Documentation . . . 293
9.163.2.1 apply(iterator &) . . . 293
9.163.2.2 can_apply(iterator) . . . 293
9.163.3 Member Data Documentation . . . 293
9.163.3.1 dset1 . . . 293
9.163.3.2 dset2 . . . 293
9.164cadabra::replace_match Class Reference . . . 293
9.164.1 Constructor & Destructor Documentation . . . 294
9.164.1.1 replace_match(const Kernel &, Ex &) . . . 294
9.164.2 Member Function Documentation . . . 294
9.164.2.1 apply(iterator &) override . . . 294
9.164.2.2 can_apply(iterator) override . . . 294
9.165cadabra::rewrite_indices Class Reference . . . 294
9.165.1 Constructor & Destructor Documentation . . . 295
9.165.1.1 rewrite_indices(const Kernel &, Ex &, Ex &, Ex &) . . . 295
9.165.2 Member Function Documentation . . . 295
9.165.2.1 apply(iterator &) override . . . 295
9.165.2.2 can_apply(iterator) override . . . 295
9.165.3 Member Data Documentation . . . 295
9.165.3.1 converters . . . 295

9.165.3.2 is_derivative_argument . . . 295
9.165.3.3 preferred . . . 295
9.166 cadabra::RiemannTensor Class Reference . . . 295
9.166.1 Constructor & Destructor Documentation . . . 296
9.166.1.1 RiemannTensor() . . . 296
9.166.2 Member Function Documentation . . . 296
9.166.2.1 name() const override . . . 296
9.166.2.2 validate(const Kernel &, const Ex &) const override . . . 296
9.167 RuntimeException Class Reference . . . 296
9.167.1 Description . . . 296
9.167.2 Constructor & Destructor Documentation . . . 297
9.167.2.1 RuntimeException(std::string="") . . . 297
9.168 cadabra::SatisfiesBianchi Class Reference . . . 297
9.168.1 Member Function Documentation . . . 297
9.168.1.1 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const override . . . 297
9.168.1.2 name() const override . . . 297
9.168.1.3 size(const Properties &, Ex &, Ex::iterator) const override . . . 298
9.169 cadabra::SelfAntiCommuting Class Reference . . . 298
9.169.1 Member Function Documentation . . . 298
9.169.1.1 name() const override . . . 298
9.169.1.2 sign() const override . . . 298
9.170 cadabra::SelfCommuting Class Reference . . . 299
9.170.1 Member Function Documentation . . . 299
9.170.1.1 name() const override . . . 299
9.170.1.2 sign() const override . . . 299
9.171 cadabra::SelfCommutingBehaviour Class Reference . . . 299
9.171.1 Member Function Documentation . . . 300
9.171.1.1 sign() const =0 . . . 300
9.172 cadabra::SelfNonCommuting Class Reference . . . 300
9.172.1 Member Function Documentation . . . 300

9.172.1.1 name() const override . . . 300
9.172.1.2 sign() const override . . . 301
9.173 cadabra2_defaults.Server Class Reference . . . 301
9.173.1 Member Function Documentation . . . 301
9.173.1.1 architecture(self) . . . 301
9.173.1.2 handles(self, otype) . . . 301
9.173.1.3 send(self, data, typestr) . . . 301
9.173.1.4 test(self) . . . 301
9.174 Server Class Reference . . . 302
9.174.1 Description . . . 302
9.174.2 Member Typedef Documentation . . . 304
9.174.2.1 ConnectionMap . . . 304
9.174.2.2 WebsocketServer . . . 304
9.174.3 Constructor & Destructor Documentation . . . 304
9.174.3.1 Server() . . . 304
9.174.3.2 Server(const Server &)=delete . . . 304
9.174.3.3 Server(const std::string &socket) . . . 304
9.174.3.4 ∼Server() . . . 304
9.174.4 Member Function Documentation . . . 304
9.174.4.1 architecture() const . . . 304
9.174.4.2 dispatch_message(websocketpp::connection_hdl, const std::string &json_string) 304
9.174.4.3 handles(const std::string &otype) const . . . 304
9.174.4.4 init() . . . 304
9.174.4.5 on_block_error(Block) . . . 304
9.174.4.6 on_block_finished(Block) . . . 304
9.174.4.7 on_close(websocketpp::connection_hdl hdl) . . . 305
9.174.4.8 on_kernel_fault(Block) . . . 305
9.174.4.9 on_message(websocketpp::connection_hdl hdl, WebsocketServer::message_ptr msg) . . . 305
9.174.4.10 on_open(websocketpp::connection_hdl hdl) . . . 305
9.174.4.11 on_socket_init(websocketpp::connection_hdl hdl, boost::asio::ip::tcp::socket &s) . 305

9.174.4.12 run() . . . 305
9.174.4.13 run_string(const std::string &, bool handle_output=true) . . . 305
9.174.4.14 send(const std::string &output, const std::string &msg_type) . . . 305
9.174.4.15 send_json(const std::string &) . . . 305
9.174.4.16 stop_block() . . . 305
9.174.4.17 wait_for_job() . . . 306
9.174.5 Member Data Documentation . . . 306
9.174.5.1 block_available . . . 306
9.174.5.2 block_available_mutex . . . 306
9.174.5.3 block_queue . . . 306
9.174.5.4 catchErr . . . 306
9.174.5.5 catchOut . . . 306
9.174.5.6 cells_ran . . . 306
9.174.5.7 connections . . . 306
9.174.5.8 current_hdl . . . 306
9.174.5.9 current_id . . . 306
9.174.5.10 job . . . 306
9.174.5.11 main_module . . . 306
9.174.5.12 main_namespace . . . 306
9.174.5.13 runner . . . 306
9.174.5.14 server_stopwatch . . . 306
9.174.5.15 socket_name . . . 306
9.174.5.16 started . . . 306
9.174.5.17 sympy_stopwatch . . . 306
9.174.5.18 ws_mutex . . . 306
9.174.5.19 wserver . . . 306
9.175 ServerWrapper Class Reference . . . 307
9.175.1 Description . . . 307
9.175.2 Constructor & Destructor Documentation . . . 307
9.175.2.1 ServerWrapper() . . . 307

9.175.3 Member Function Documentation . . . 307
9.175.3.1 group(std::string name="""") override . . . 307
9.175.3.2 progress(int n, int total) override . . . 307
9.175.4 Member Data Documentation . . . 308
9.175.4.1 python_group . . . 308
9.175.4.2 python_progress . . . 308
9.175.4.3 server . . . 308
9.176 snoop::Snoop Class Reference . . . 308
9.176.1 Description . . . 308
9.176.2 Constructor & Destructor Documentation . . . 309
9.176.2.1 Snoop() . . . 309
9.176.2.2 ∼Snoop() . . . 309
9.176.3 Member Function Documentation . . . 309
9.176.3.1 get_user_uuid(const std::string &app_name) . . . 309
9.176.3.2 init(const std::string &app_name, const std::string &app_version, std::string server="""", std::string local_log_file="""") . . . 309
9.176.3.3 operator()(const std::string &type, std::string fl="""", int loc=-1, std::string method="""") . . . 309
9.176.3.4 operator<<(const T &obj) . . . 309
9.176.3.5 operator<<(const Flush &) . . . 309
9.176.3.6 set_sync_immediately(bool) . . . 309
9.176.3.7 sync_logs_with_server(bool from_wsthread=false) . . . 310
9.176.3.8 sync_runs_with_server(bool from_wsthread=false) . . . 310
9.176.3.9 sync_with_server(bool from_wsthread=false) . . . 310
9.176.4 Member Data Documentation . . . 310
9.176.4.1 impl . . . 310
9.176.4.2 out_ . . . 310
9.176.4.3 SnoopImpl . . . 310
9.176.4.4 sync_immediately_ . . . 310
9.177 snoop::SnoopImpl Class Reference . . . 310
9.177.1 Constructor & Destructor Documentation . . . 312

9.177.1.1 SnoopImpl(Snoop *) . . . 312
9.177.1.2 ~SnoopImpl() . . . 312
9.177.2 Member Function Documentation . . . 312
9.177.2.1 create_tables() . . . 312
9.177.2.2 get_app_registrations(std::string uuid_filter="") . . . 312
9.177.2.3 get_user_uuid(const std::string &app_name) . . . 312
9.177.2.4 init(const std::string &app_name, const std::string &app_version, std::string server="", std::string local_log_file="") . . . 312
9.177.2.5 obtain_uuid() . . . 313
9.177.2.6 on_client_close(websocketpp::connection_hdl hdl) . . . 313
9.177.2.7 on_client_fail(websocketpp::connection_hdl hdl) . . . 313
9.177.2.8 on_client_message(websocketpp::connection_hdl hdl, message_ptr msg) . . . 313
9.177.2.9 on_client_open(websocketpp::connection_hdl hdl) . . . 313
9.177.2.10 operator()(const std::string &type, std::string fl="", int loc=-1, std::string method="") . . . 313
9.177.2.11 operator<<(const Flush &) . . . 313
9.177.2.12 start_websocket_client() . . . 313
9.177.2.13 store_app_entry(Snoop::AppEntry &) . . . 313
9.177.2.14 store_app_entry_without_lock(Snoop::AppEntry &) . . . 313
9.177.2.15 store_log_entry(Snoop::LogEntry &, bool avoid_server_duplicates) . . . 313
9.177.2.16 sync_logs_with_server(bool from_wsthread=false) . . . 314
9.177.2.17 sync_runs_with_server(bool from_wsthread=false) . . . 314
9.177.2.18 sync_with_server(bool from_wsthread=false) . . . 314
9.177.3 Member Data Documentation . . . 314
9.177.3.1 call_mutex . . . 314
9.177.3.2 connection . . . 314
9.177.3.3 connection_attempt_failed . . . 314
9.177.3.4 connection_cv . . . 314
9.177.3.5 connection_is_open . . . 314
9.177.3.6 connection_mutex . . . 314
9.177.3.7 db . . . 314

9.177.3.8 id_for_uuid_statement . . . 314
9.177.3.9 insert_statement . . . 314
9.177.3.10 our_connection_hdl . . . 314
9.177.3.11 server_ . . . 314
9.177.3.12 snoop_ . . . 314
9.177.3.13 sqlite_mutex . . . 314
9.177.3.14 this_app_ . . . 314
9.177.3.15 this_log_ . . . 314
9.177.3.16 wsclient . . . 314
9.177.3.17 wsclient_thread . . . 315
9.178 cadabra::sort_product Class Reference . . . 315
9.178.1 Constructor & Destructor Documentation . . . 315
9.178.1.1 sort_product(const Kernel &, Ex &) . . . 315
9.178.2 Member Function Documentation . . . 315
9.178.2.1 apply(iterator &) . . . 315
9.178.2.2 can_apply(iterator) . . . 315
9.178.3 Member Data Documentation . . . 316
9.178.3.1 ignore_numbers_ . . . 316
9.179 cadabra::sort_spinors Class Reference . . . 316
9.179.1 Constructor & Destructor Documentation . . . 316
9.179.1.1 sort_spinors(const Kernel &, Ex &) . . . 316
9.179.2 Member Function Documentation . . . 316
9.179.2.1 apply(iterator &) override . . . 316
9.179.2.2 can_apply(iterator) override . . . 317
9.179.3 Member Data Documentation . . . 317
9.179.3.1 gammamat . . . 317
9.179.3.2 one . . . 317
9.179.3.3 two . . . 317
9.180 cadabra::sort_sum Class Reference . . . 317
9.180.1 Constructor & Destructor Documentation . . . 317

9.180.1.1 sort_sum(const Kernel &, Ex &) . . . 317
9.180.2 Member Function Documentation . . . 317
9.180.2.1 apply(iterator &) override . . . 317
9.180.2.2 can_apply(iterator) override . . . 318
9.180.2.3 should_swap(iterator obj, int subtree_comparison) const . . . 318
9.181 cadabra::SortOrder Class Reference . . . 318
9.181.1 Member Function Documentation . . . 318
9.181.1.1 equals(const property ∗) const . . . 318
9.181.1.2 name() const . . . 318
9.182 cadabra::Spinor Class Reference . . . 319
9.182.1 Member Enumeration Documentation . . . 319
9.182.1.1 anonymous enum . . . 319
9.182.2 Constructor & Destructor Documentation . . . 319
9.182.2.1 Spinor() . . . 319
9.182.2.2 ∼Spinor() . . . 319
9.182.3 Member Function Documentation . . . 319
9.182.3.1 name() const override . . . 319
9.182.3.2 parse(const Kernel &, keyval_t &keyvals) override . . . 320
9.182.4 Member Data Documentation . . . 320
9.182.4.1 chirality . . . 320
9.182.4.2 dimension . . . 320
9.182.4.3 majorana . . . 320
9.182.4.4 weyl . . . 320
9.183 split Class Reference . . . 320
9.183.1 Description . . . 320
9.183.2 Constructor & Destructor Documentation . . . 321
9.183.2.1 split(const Kernel &, Ex &) . . . 321
9.183.3 Member Function Documentation . . . 321
9.183.3.1 apply(iterator &) . . . 321
9.183.3.2 can_apply(iterator) . . . 321

9.184 cadabra::split_gamma Class Reference . . . 321
9.184.1 Constructor & Destructor Documentation . . . 321
9.184.1.1 split_gamma(const Kernel &, Ex &, bool on_back) . . . 321
9.184.2 Member Function Documentation . . . 321
9.184.2.1 apply(iterator &) override . . . 321
9.184.2.2 can_apply(iterator) override . . . 322
9.184.3 Member Data Documentation . . . 322
9.184.3.1 on_back . . . 322
9.185 cadabra::split_index Class Reference . . . 322
9.185.1 Constructor & Destructor Documentation . . . 323
9.185.1.1 split_index(const Kernel &, Ex &, Ex &) . . . 323
9.185.2 Member Function Documentation . . . 323
9.185.2.1 apply(iterator &) . . . 323
9.185.2.2 can_apply(iterator) . . . 323
9.185.3 Member Data Documentation . . . 323
9.185.3.1 full_class . . . 323
9.185.3.2 num1 . . . 323
9.185.3.3 num2 . . . 323
9.185.3.4 part1_class . . . 323
9.185.3.5 part1_coord . . . 323
9.185.3.6 part1_coord_node . . . 323
9.185.3.7 part1_is_number . . . 323
9.185.3.8 part2_class . . . 323
9.185.3.9 part2_coord . . . 323
9.185.3.10 part2_coord_node . . . 323
9.185.3.11 part2_is_number . . . 323
9.186 Stopwatch Class Reference . . . 324
9.186.1 Constructor & Destructor Documentation . . . 324
9.186.1.1 Stopwatch() . . . 324
9.186.2 Member Function Documentation . . . 324

9.186.2.1 checkpoint_() const . . . 324
9.186.2.2 reset() . . . 324
9.186.2.3 seconds() const . . . 324
9.186.2.4 start() . . . 324
9.186.2.5 stop() . . . 324
9.186.2.6 stopped() const . . . 325
9.186.2.7 useconds() const . . . 325
9.186.3 Friends And Related Function Documentation . . . 325
9.186.3.1 operator<< . . . 325
9.186.4 Member Data Documentation . . . 325
9.186.4.1 diffsec . . . 325
9.186.4.2 diffusec . . . 325
9.186.4.3 stopped_ . . . 325
9.186.4.4 tv2 . . . 325
9.186.4.5 tz . . . 325
9.187 cadabra::str_node Class Reference . . . 325
9.187.1 Description . . . 325
9.187.2 Member Enumeration Documentation . . . 326
9.187.2.1 bracket_t . . . 326
9.187.2.2 parent_rel_t . . . 327
9.187.3 Constructor & Destructor Documentation . . . 327
9.187.3.1 str_node(void) . . . 327
9.187.3.2 str_node(nset_t::iterator name, bracket_t btype=b_none, parent_rel_t ptype=p_none) . . . 327
9.187.3.3 str_node(const std::string &name, bracket_t btype=b_none, parent_rel_t ptype=p_none) . . . 327
9.187.4 Member Function Documentation . . . 327
9.187.4.1 compare_name_brack_par(const str_node &, const str_node &) . . . 327
9.187.4.2 compare_name_inverse_par(const str_node &, const str_node &) . . . 327
9.187.4.3 compare_names_only(const str_node &, const str_node &) . . . 327
9.187.4.4 flip_parent_rel() . . . 327

9.187.4.5 is_autodeclare_wildcard() const . . . 328
9.187.4.6 is_command() const . . . 328
9.187.4.7 is_identity() const . . . 328
9.187.4.8 is_index() const . . . 328
9.187.4.9 is_indexplus_wildcard() const . . . 328
9.187.4.10 is_indexstar_wildcard() const . . . 328
9.187.4.11 is_inert_command() const . . . 328
9.187.4.12 is_integer() const . . . 328
9.187.4.13 is_name_wildcard() const . . . 328
9.187.4.14 is_numbered_symbol() const . . . 328
9.187.4.15 is_object_wildcard() const . . . 328
9.187.4.16 is_quoted_string() const . . . 328
9.187.4.17 is_range_wildcard() const . . . 328
9.187.4.18 is_rational() const . . . 328
9.187.4.19 is_siblings_wildcard() const . . . 328
9.187.4.20 is_unsimplified_integer() const . . . 328
9.187.4.21 is_unsimplified_rational() const . . . 328
9.187.4.22 is_zero() const . . . 328
9.187.4.23 name_only() . . . 328
9.187.4.24 operator<(const str_node &) const . . . 328
9.187.4.25 operator==(const str_node &) const . . . 328
9.187.5 Member Data Documentation . . . 328
9.187.5.1 fl . . . 328
9.187.5.2 multiplier . . . 328
9.187.5.3 name . . . 328
9.188 cadabra::substitute Class Reference . . . 329
9.188.1 Description . . . 329
9.188.2 Constructor & Destructor Documentation . . . 329
9.188.2.1 substitute(const Kernel &, Ex &tr, Ex &args) . . . 329
9.188.3 Member Function Documentation . . . 329

9.188.3.1 apply(iterator &) . . . 329
9.188.3.2 can_apply(iterator st) . . . 330
9.188.4 Member Data Documentation . . . 330
9.188.4.1 args . . . 330
9.188.4.2 comparator . . . 330
9.188.4.3 conditions . . . 330
9.188.4.4 lhs_contains_dummies . . . 330
9.188.4.5 rhs_contains_dummies . . . 330
9.188.4.6 sort_product_ . . . 330
9.188.4.7 use_rule . . . 330
9.189 cadabra::sym Class Reference . . . 330
9.189.1 Constructor & Destructor Documentation . . . 331
9.189.1.1 sym(const Kernel &, Ex &, Ex &, bool sign) . . . 331
9.189.2 Member Function Documentation . . . 331
9.189.2.1 apply(iterator &it) override . . . 331
9.189.2.2 can_apply(iterator) override . . . 331
9.189.2.3 doit(iterator &, bool) . . . 331
9.189.3 Member Data Documentation . . . 331
9.189.3.1 argloc_2_treeloc . . . 331
9.189.3.2 objects . . . 331
9.189.3.3 raw_ints . . . 331
9.189.3.4 sign . . . 331
9.190 cadabra::Symbol Class Reference . . . 332
9.190.1 Member Function Documentation . . . 332
9.190.1.1 get(const Properties &, Ex::iterator, bool ignore_parent_rel=false) . . . 332
9.190.1.2 name() const . . . 332
9.191 combin::symm_helper< T > Class Template Reference . . . 332
9.191.1 Constructor & Destructor Documentation . . . 333
9.191.1.1 symm_helper(symmetriser< T > &) . . . 333
9.191.2 Member Function Documentation . . . 333

9.191.2.1 clear() . . . 333
9.191.2.2 vector_generated(const std::vector< unsigned int > &) . . . 333
9.191.3 Member Data Documentation . . . 333
9.191.3.1 current_multiplicity . . . 333
9.191.3.2 first_one . . . 333
9.191.3.3 owner_ . . . 333
9.192combin::symm_val_helper< T > Class Template Reference . . . 334
9.192.1 Constructor & Destructor Documentation . . . 334
9.192.1.1 symm_val_helper(symmetriser< T > &) . . . 334
9.192.2 Member Function Documentation . . . 334
9.192.2.1 clear() . . . 334
9.192.2.2 vector_generated(const std::vector< unsigned int > &) . . . 334
9.192.3 Member Data Documentation . . . 335
9.192.3.1 current_multiplicity . . . 335
9.192.3.2 first_one . . . 335
9.192.3.3 owner_ . . . 335
9.193cadabra::Symmetric Class Reference . . . 335
9.193.1 Member Function Documentation . . . 335
9.193.1.1 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const override . . . 335
9.193.1.2 name() const override . . . 335
9.193.1.3 size(const Properties &, Ex &, Ex::iterator) const override . . . 336
9.194combin::symmetriser< T > Class Template Reference . . . 336
9.194.1 Constructor & Destructor Documentation . . . 337
9.194.1.1 symmetriser() . . . 337
9.194.2 Member Function Documentation . . . 337
9.194.2.1 apply_symmetry(long start=-1, long end=-1) . . . 337
9.194.2.2 clear() . . . 337
9.194.2.3 collect() . . . 337
9.194.2.4 operator[](unsigned int) const . . . 337
9.194.2.5 remove_multiplicity_zero() . . . 337

9.194.2.6 set_multiplicity(unsigned int pos, int val) . . . 337
9.194.2.7 signature(unsigned int) const . . . 337
9.194.2.8 size() const . . . 337
9.194.2.9 values_to_locations(const std::vector< T > &values) const . . . 337
9.194.3 Friends And Related Function Documentation . . . 338
9.194.3.1 symm_helper< T > . . . 338
9.194.3.2 symm_val_helper< T > . . . 338
9.194.4 Member Data Documentation . . . 338
9.194.4.1 block_length . . . 338
9.194.4.2 current_ . . . 338
9.194.4.3 input_asym . . . 338
9.194.4.4 multiplicity . . . 338
9.194.4.5 original . . . 338
9.194.4.6 originals . . . 338
9.194.4.7 permutation_sign . . . 338
9.194.4.8 permute_blocks . . . 338
9.194.4.9 sh_ . . . 338
9.194.4.10 sublengths . . . 338
9.194.4.11 sublengths_scattered . . . 338
9.194.4.12 svh_ . . . 338
9.194.4.13 value_permute . . . 338
9.195 cadabra2_defaults.Sympy Class Reference . . . 338
9.195.1 Member Data Documentation . . . 339
9.195.1.1 __version__ . . . 339
9.196 cadabra::tab_basics Class Reference . . . 339
9.196.1 Member Typedef Documentation . . . 340
9.196.1.1 uinttab_t . . . 340
9.196.1.2 uinttabs_t . . . 340
9.196.2 Constructor & Destructor Documentation . . . 340
9.196.2.1 tab_basics(const Kernel &, Ex &) . . . 340

9.196.3 Member Function Documentation . . . 340
9.196.3.1 find_obj(const Ex &other) . . . 340
9.196.3.2 tabs_to_singlet_rules(uinttabs_t &, iterator) . . . 340
9.196.3.3 tabs_to_tree(uinttabs_t &, iterator, iterator, bool even_only) . . . 340
9.196.3.4 tree_to_numerical_tab(iterator, uinttab_t &) . . . 340
9.196.4 Member Data Documentation . . . 340
9.196.4.1 num_to_it . . . 340
9.197 cadabra::Tableau Class Reference . . . 340
9.197.1 Constructor & Destructor Documentation . . . 341
9.197.1.1 ~Tableau() . . . 341
9.197.2 Member Function Documentation . . . 341
9.197.2.1 name() const . . . 341
9.197.2.2 parse(Ex &, Ex::iterator, Ex::iterator, keyval_t &) . . . 341
9.197.3 Member Data Documentation . . . 341
9.197.3.1 dimension . . . 341
9.198 yngtab::tableau Class Reference . . . 341
9.198.1 Constructor & Destructor Documentation . . . 342
9.198.1.1 ~tableau() . . . 342
9.198.2 Member Function Documentation . . . 342
9.198.2.1 add_box(unsigned int row) . . . 342
9.198.2.2 clear() . . . 342
9.198.2.3 number_of_rows() const . . . 342
9.198.2.4 operator=(const tableau &) . . . 343
9.198.2.5 remove_box(unsigned int row) . . . 343
9.198.2.6 row_size(unsigned int row) const . . . 343
9.198.3 Member Data Documentation . . . 343
9.198.3.1 rows . . . 343
9.199 yngtab::tableau_base Class Reference . . . 343
9.199.1 Constructor & Destructor Documentation . . . 344
9.199.1.1 tableau_base() . . . 344

9.199.1.2 ~tableau_base() . . . 344
9.199.2 Member Function Documentation . . . 344
9.199.2.1 add_box(unsigned int row)=0 . . . 344
9.199.2.2 add_row(unsigned int row_size) . . . 344
9.199.2.3 clear()=0 . . . 344
9.199.2.4 column_size(unsigned int col) const . . . 344
9.199.2.5 dimension(unsigned int) const . . . 344
9.199.2.6 hook_length(unsigned int row, unsigned int col) const . . . 344
9.199.2.7 hook_length_prod() const . . . 344
9.199.2.8 number_of_rows() const =0 . . . 344
9.199.2.9 remove_box(unsigned int row)=0 . . . 344
9.199.2.10 row_size(unsigned int row) const =0 . . . 345
9.199.3 Member Data Documentation . . . 345
9.199.3.1 multiplicity . . . 345
9.199.3.2 selfdual_column . . . 345
9.200 cadabra::TableauBase Class Reference . . . 345
9.200.1 Member Typedef Documentation . . . 346
9.200.1.1 tab_t . . . 346
9.200.2 Constructor & Destructor Documentation . . . 346
9.200.2.1 ~TableauBase() . . . 346
9.200.3 Member Function Documentation . . . 346
9.200.3.1 get_indexgroup(const Properties &, Ex &, Ex::iterator, int) const . . . 346
9.200.3.2 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const =0 . . . 346
9.200.3.3 is_simple_symmetry(const Properties &, Ex &, Ex::iterator) const . . . 346
9.200.3.4 only_column_exchange() const . . . 346
9.200.3.5 size(const Properties &, Ex &, Ex::iterator) const =0 . . . 346
9.201 cadabra::TableauSymmetry Class Reference . . . 346
9.201.1 Constructor & Destructor Documentation . . . 347
9.201.1.1 ~TableauSymmetry() . . . 347
9.201.2 Member Function Documentation . . . 347

9.201.2.1 get_tab(const Properties &, Ex &, Ex::iterator, unsigned int) const override . . . 347
9.201.2.2 latex(std::ostream &) const override . . . 347
9.201.2.3 name() const override . . . 347
9.201.2.4 only_column_exchange() const override . . . 348
9.201.2.5 parse(const Kernel &, keyval_t &) override . . . 348
9.201.2.6 size(const Properties &, Ex &, Ex::iterator) const override . . . 348
9.201.3 Member Data Documentation . . . 348
9.201.3.1 only_col_ . . . 348
9.201.3.2 tabs . . . 348
9.202yngtab::tableaux< T > Class Template Reference . . . 348
9.202.1 Member Typedef Documentation . . . 349
9.202.1.1 back_insert_iterator . . . 349
9.202.1.2 tableau_container_t . . . 349
9.202.2 Member Function Documentation . . . 349
9.202.2.1 add_tableau(const T &) . . . 349
9.202.2.2 get_back_insert_iterator() . . . 349
9.202.2.3 remove_nullifying_traces() . . . 349
9.202.2.4 standard_form() . . . 349
9.202.2.5 symmetrise(const T &tabsym) . . . 349
9.202.2.6 total_dimension(unsigned int dim) . . . 349
9.202.3 Member Data Documentation . . . 349
9.202.3.1 storage . . . 349
9.203cadabra::take_match Class Reference . . . 349
9.203.1 Constructor & Destructor Documentation . . . 350
9.203.1.1 take_match(const Kernel &k, Ex &e, Ex &r) . . . 350
9.203.2 Member Function Documentation . . . 350
9.203.2.1 apply(iterator &) override . . . 350
9.203.2.2 can_apply(iterator) override . . . 350
9.203.3 Member Data Documentation . . . 350
9.203.3.1 rules . . . 350

9.204 cadabra::exchange::tensor_type_t Struct Reference . . . 350
9.204.1 Member Data Documentation . . . 351
9.204.1.1 name . . . 351
9.204.1.2 number_of_indices . . . 351
9.205 cadabra::TerminalStream Class Reference . . . 351
9.205.1 Constructor & Destructor Documentation . . . 351
9.205.1.1 TerminalStream(const Kernel &, std::ostream &) . . . 351
9.205.2 Member Function Documentation . . . 351
9.205.2.1 operator<<(const Ex &) . . . 351
9.205.2.2 operator<<(std::shared_ptr< Ex >) . . . 351
9.205.2.3 operator<<(const T &obj) . . . 351
9.205.2.4 operator<<(std::ostream &(*os)(std::ostream &)) . . . 351
9.205.3 Member Data Documentation . . . 351
9.205.3.1 kernel . . . 351
9.205.3.2 out_ . . . 351
9.206 cadabra::TeXEngine Class Reference . . . 352
9.206.1 Description . . . 352
9.206.2 Constructor & Destructor Documentation . . . 353
9.206.2.1 TeXEngine() . . . 353
9.206.2.2 ~TeXEngine() . . . 353
9.206.3 Member Function Documentation . . . 353
9.206.3.1 checkin(const std::string &, const std::string &startwrap, const std::string &endwrap) 353
9.206.3.2 checkout(std::shared_ptr< TeXRequest >) . . . 353
9.206.3.3 checkout_all() . . . 353
9.206.3.4 convert_all() . . . 353
9.206.3.5 convert_one(std::shared_ptr< TeXRequest >) . . . 353
9.206.3.6 convert_set(std::set< std::shared_ptr< TeXRequest > > &) . . . 353
9.206.3.7 erase_file(const std::string &) const . . . 353
9.206.3.8 handle_latex_errors(const std::string &, int exit_code) const . . . 353
9.206.3.9 invalidate_all() . . . 353

9.206.3.10 modify(std::shared_ptr< TeXRequest >, const std::string &) . . . 354
9.206.3.11 set_font_size(int font_size) . . . 354
9.206.3.12 set_geometry(int horizontal_pixels) . . . 354
9.206.3.13 set_scale(double) . . . 354
9.206.4 Member Data Documentation . . . 354
9.206.4.1 font_size_ . . . 354
9.206.4.2 horizontal_pixels_ . . . 354
9.206.4.3 latex_packages . . . 354
9.206.4.4 millimeter_per_inch . . . 354
9.206.4.5 preamble_string . . . 354
9.206.4.6 requests . . . 354
9.206.4.7 scale_ . . . 354
9.207 cadabra::TeXEngine::TeXException Class Reference . . . 354
9.207.1 Constructor & Destructor Documentation . . . 355
9.207.1.1 TeXException(const std::string &) . . . 355
9.208 cadabra::TeXEngine::TeXRequest Class Reference . . . 355
9.208.1 Constructor & Destructor Documentation . . . 356
9.208.1.1 TeXRequest() . . . 356
9.208.2 Member Function Documentation . . . 356
9.208.2.1 height() const . . . 356
9.208.2.2 image() const . . . 356
9.208.2.3 width() const . . . 356
9.208.3 Friends And Related Function Documentation . . . 356
9.208.3.1 TeXEngine . . . 356
9.208.4 Member Data Documentation . . . 356
9.208.4.1 end_wrap . . . 356
9.208.4.2 height_ . . . 356
9.208.4.3 image_ . . . 356
9.208.4.4 latex_string . . . 356
9.208.4.5 needs_generating . . . 356

9.208.4.6 start_wrap . . . 356
9.208.4.7 width_ . . . 356
9.209 cadabra::TeXView Class Reference . . . 356
9.209.1 Description . . . 356
9.209.2 Constructor & Destructor Documentation . . . 357
9.209.2.1 TeXView(TeXEngine &, DTree::iterator, int hmargin=25) . . . 357
9.209.2.2 ∼TeXView() . . . 357
9.209.3 Member Function Documentation . . . 357
9.209.3.1 convert() . . . 357
9.209.3.2 dim(bool) . . . 357
9.209.3.3 on_button_release_event(GdkEventButton ∗) override . . . 358
9.209.3.4 on_show() override . . . 358
9.209.3.5 update_image() . . . 358
9.209.4 Member Data Documentation . . . 358
9.209.4.1 content . . . 358
9.209.4.2 datacell . . . 358
9.209.4.3 engine . . . 358
9.209.4.4 hbox . . . 358
9.209.4.5 image . . . 358
9.209.4.6 pixbuf . . . 358
9.209.4.7 show_hide_requested . . . 358
9.209.4.8 tex_error . . . 358
9.209.4.9 vbox . . . 358
9.210 ProgressMonitor::Total Class Reference . . . 358
9.210.1 Constructor & Destructor Documentation . . . 359
9.210.1.1 Total() . . . 359
9.210.2 Member Function Documentation . . . 359
9.210.2.1 operator==(const Total &other) const . . . 359
9.210.2.2 str() const . . . 359
9.210.2.3 time_spent_as_long() const . . . 359

9.210.3 Member Data Documentation . . . 359
9.210.3.1 call_count . . . 359
9.210.3.2 name . . . 359
9.210.3.3 time_spent . . . 359
9.210.3.4 total_steps . . . 359
9.211 cadabra::Traceless Class Reference . . . 360
9.211.1 Constructor & Destructor Documentation . . . 360
9.211.1.1 ∼Traceless() . . . 360
9.211.2 Member Function Documentation . . . 360
9.211.2.1 name() const . . . 360
9.212 cadabra::tree_equal_obj Class Reference . . . 360
9.212.1 Constructor & Destructor Documentation . . . 361
9.212.1.1 tree_equal_obj(const Properties ∗) . . . 361
9.212.2 Member Function Documentation . . . 361
9.212.2.1 operator()(const Ex &first, const Ex &second) const . . . 361
9.212.3 Member Data Documentation . . . 361
9.212.3.1 properties . . . 361
9.213 cadabra::tree_exact_equal_mod_prel_obj Class Reference . . . 361
9.213.1 Constructor & Destructor Documentation . . . 362
9.213.1.1 tree_exact_equal_mod_prel_obj(const Properties ∗) . . . 362
9.213.2 Member Function Documentation . . . 362
9.213.2.1 operator()(const Ex &first, const Ex &second) const . . . 362
9.213.3 Member Data Documentation . . . 362
9.213.3.1 properties . . . 362
9.214 cadabra::tree_exact_equal_obj Class Reference . . . 362
9.214.1 Constructor & Destructor Documentation . . . 362
9.214.1.1 tree_exact_equal_obj(const Properties ∗) . . . 362
9.214.2 Member Function Documentation . . . 362
9.214.2.1 operator()(const Ex &first, const Ex &second) const . . . 362
9.214.3 Member Data Documentation . . . 362

9.214.3.1 properties . . . 362
9.215 cadabra::tree_exact_less_for_indexmap_obj Class Reference . . . 363
9.215.1 Description . . . 363
9.215.2 Member Function Documentation . . . 363
9.215.2.1 operator()(const Ex &first, const Ex &second) const . . . 363
9.216 cadabra::tree_exact_less_mod_prel_obj Class Reference . . . 363
9.216.1 Constructor & Destructor Documentation . . . 364
9.216.1.1 tree_exact_less_mod_prel_obj(const Properties *) . . . 364
9.216.2 Member Function Documentation . . . 364
9.216.2.1 operator()(const Ex &first, const Ex &second) const . . . 364
9.216.3 Member Data Documentation . . . 364
9.216.3.1 properties . . . 364
9.217 cadabra::tree_exact_less_no_wildcards_mod_prel_obj Class Reference . . . 364
9.217.1 Constructor & Destructor Documentation . . . 364
9.217.1.1 tree_exact_less_no_wildcards_mod_prel_obj(const Properties *) . . . 364
9.217.2 Member Function Documentation . . . 364
9.217.2.1 operator()(const Ex &first, const Ex &second) const . . . 364
9.217.3 Member Data Documentation . . . 364
9.217.3.1 properties . . . 364
9.218 cadabra::tree_exact_less_no_wildcards_obj Class Reference . . . 365
9.218.1 Description . . . 365
9.218.2 Constructor & Destructor Documentation . . . 365
9.218.2.1 tree_exact_less_no_wildcards_obj() . . . 365
9.218.2.2 tree_exact_less_no_wildcards_obj(const Properties *) . . . 365
9.218.3 Member Function Documentation . . . 365
9.218.3.1 operator()(const Ex &first, const Ex &second) const . . . 365
9.218.4 Member Data Documentation . . . 365
9.218.4.1 properties . . . 365
9.219 cadabra::tree_exact_less_obj Class Reference . . . 365
9.219.1 Description . . . 365

9.219.2 Constructor & Destructor Documentation . . . 366
9.219.2.1 tree_exact_less_obj(const Properties *) . . . 366
9.219.3 Member Function Documentation . . . 366
9.219.3.1 operator()(const Ex &first, const Ex &second) const . . . 366
9.219.4 Member Data Documentation . . . 366
9.219.4.1 properties . . . 366
9.220 cadabra::tree_less_modprel_obj Class Reference . . . 366
9.220.1 Constructor & Destructor Documentation . . . 367
9.220.1.1 tree_less_modprel_obj(const Properties *) . . . 367
9.220.2 Member Function Documentation . . . 367
9.220.2.1 operator()(const Ex &first, const Ex &second) const . . . 367
9.220.3 Member Data Documentation . . . 367
9.220.3.1 properties . . . 367
9.221 cadabra::tree_less_obj Class Reference . . . 367
9.221.1 Description . . . 367
9.221.2 Constructor & Destructor Documentation . . . 367
9.221.2.1 tree_less_obj(const Properties *) . . . 367
9.221.3 Member Function Documentation . . . 367
9.221.3.1 operator()(const Ex &first, const Ex &second) const . . . 367
9.221.4 Member Data Documentation . . . 367
9.221.4.1 properties . . . 367
9.222 UI Class Reference . . . 368
9.222.1 Member Function Documentation . . . 368
9.222.1.1 run() . . . 368
9.223 cadabra::unwrap Class Reference . . . 368
9.223.1 Constructor & Destructor Documentation . . . 368
9.223.1.1 unwrap(const Kernel &, Ex &, Ex &) . . . 368
9.223.2 Member Function Documentation . . . 368
9.223.2.1 apply(iterator &) override . . . 368
9.223.2.2 can_apply(iterator) override . . . 369

9.223.3 Member Data Documentation . . . 369
9.223.3.1 wrappers . . . 369
9.224 cadabra::vary Class Reference . . . 369
9.224.1 Description . . . 369
9.224.2 Constructor & Destructor Documentation . . . 369
9.224.2.1 vary(const Kernel &, Ex &tr, Ex &args) . . . 369
9.224.3 Member Function Documentation . . . 369
9.224.3.1 apply(iterator &) override . . . 369
9.224.3.2 can_apply(iterator st) override . . . 370
9.224.4 Member Data Documentation . . . 370
9.224.4.1 args . . . 370
9.225 cadabra::VisualCell Class Reference . . . 370
9.225.1 Description . . . 370
9.225.2 Member Data Documentation . . . 370
9.225.2.1 "@4 . . . 370
9.225.2.2 document . . . 371
9.225.2.3 imagebox . . . 371
9.225.2.4 inbox . . . 371
9.225.2.5 outbox . . . 371
9.226 cadabra::Weight Class Reference . . . 371
9.226.1 Member Function Documentation . . . 371
9.226.1.1 name() const override . . . 371
9.226.1.2 parse(const Kernel &, keyval_t &) override . . . 372
9.226.1.3 unnamed_argument() const override . . . 372
9.226.1.4 value(const Kernel &, Ex::iterator, const std::string &forcedlabel) const override . . . 372
9.226.2 Member Data Documentation . . . 372
9.226.2.1 value_ . . . 372
9.227 cadabra::WeightBase Class Reference . . . 372
9.227.1 Member Function Documentation . . . 373
9.227.1.1 value(const Kernel &, Ex::iterator, const std::string &forcedlabel) const =0 . . . 373

9.228cadabra::WeightInherit::WeightException Class Reference . . . 373
9.228.1 Constructor & Destructor Documentation . . . 373
9.228.1.1 WeightException(const std::string &) . . . 373
9.229cadabra::WeightInherit Class Reference . . . 374
9.229.1 Member Enumeration Documentation . . . 374
9.229.1.1 anonymous enum . . . 374
9.229.2 Member Function Documentation . . . 375
9.229.2.1 name() const override . . . 375
9.229.2.2 parse(const Kernel &, keyval_t &) override . . . 375
9.229.2.3 unnamed_argument() const override . . . 375
9.229.2.4 value(const Kernel &, Ex::iterator, const std::string &forcedlabel) const override . . . 375
9.229.3 Member Data Documentation . . . 375
9.229.3.1 combination_type . . . 375
9.229.3.2 value_self . . . 375
9.230cadabra::WeylTensor Class Reference . . . 375
9.230.1 Constructor & Destructor Documentation . . . 376
9.230.1.1 WeylTensor() . . . 376
9.230.2 Member Function Documentation . . . 376
9.230.2.1 latex(std::ostream &) const override . . . 376
9.230.2.2 name() const override . . . 376
9.230.2.3 validate(const Kernel &, const Ex &) const override . . . 376
9.231X Struct Reference . . . 376
9.231.1 Description . . . 376
9.232Y Struct Reference . . . 377
9.233cadabra::young_project Class Reference . . . 377
9.233.1 Member Typedef Documentation . . . 378
9.233.1.1 name_tab_t . . . 378
9.233.1.2 pos_tab_t . . . 378
9.233.2 Constructor & Destructor Documentation . . . 378
9.233.2.1 young_project(const Kernel &, Ex &) . . . 378

9.233.2.2 young_project(const Kernel &, Ex &, const std::vector< int > &shape, const std::vector< int > &indices) . . . 378
9.233.3 Member Function Documentation . . . 378
9.233.3.1 apply(iterator &) . . . 378
9.233.3.2 can_apply(iterator) . . . 378
9.233.3.3 nth_index_node(iterator, unsigned int) . . . 378
9.233.4 Member Data Documentation . . . 378
9.233.4.1 asym_ranges . . . 378
9.233.4.2 nametab . . . 378
9.233.4.3 remove_traces . . . 378
9.233.4.4 sym . . . 378
9.233.4.5 tab . . . 378
9.234 cadabra::young_project_product Class Reference . . . 379
9.234.1 Constructor & Destructor Documentation . . . 379
9.234.1.1 young_project_product(const Kernel &, Ex &) . . . 379
9.234.2 Member Function Documentation . . . 379
9.234.2.1 apply(iterator &) override . . . 379
9.234.2.2 can_apply(iterator) override . . . 379
9.235 cadabra::young_project_tensor Class Reference . . . 379
9.235.1 Constructor & Destructor Documentation . . . 380
9.235.1.1 young_project_tensor(const Kernel &, Ex &, bool) . . . 380
9.235.2 Member Function Documentation . . . 380
9.235.2.1 apply(iterator &) . . . 380
9.235.2.2 can_apply(iterator) . . . 380
9.235.3 Member Data Documentation . . . 380
9.235.3.1 modulo_monoterm . . . 380
9.235.3.2 sym . . . 380
9.235.3.3 tb . . . 380
9.236 Z Struct Reference . . . 381
9.236.1 Constructor & Destructor Documentation . . . 381
9.236.1.1 Z() . . . 381

10 File Documentation **383**

10.1 client_server/Actions.cc File Reference . . . 383

10.2 client_server/Actions.hh File Reference . . . 383

10.3 client_server/cadabra-server.cc File Reference . . . 384

10.3.1 Function Documentation . . . 384

10.3.1.1 main() . . . 384

10.4 client_server/cadabra2cadabra.cc File Reference . . . 384

10.4.1 Function Documentation . . . 384

10.4.1.1 main(int argc, char **argv) . . . 384

10.5 client_server/cadabra2html.cc File Reference . . . 384

10.5.1 Function Documentation . . . 385

10.5.1.1 main(int argc, char **argv) . . . 385

10.6 client_server/ComputeThread.cc File Reference . . . 385

10.6.1 Typedef Documentation . . . 385

10.6.1.1 client . . . 385

10.6.1.2 SlotSpawnChildSetup . . . 385

10.7 client_server/ComputeThread.hh File Reference . . . 385

10.7.1 Typedef Documentation . . . 386

10.7.1.1 message_ptr . . . 386

10.7.1.2 scoped_lock . . . 386

10.7.1.3 WSClient . . . 386

10.8 client_server/DataCell.cc File Reference . . . 386

10.9 client_server/DataCell.hh File Reference . . . 386

10.10 client_server/DocumentThread.cc File Reference . . . 387

10.11 client_server/DocumentThread.hh File Reference . . . 388

10.12 client_server/GUIBase.hh File Reference . . . 388

10.13 client_server/popen2.cc File Reference . . . 388

10.13.1 Macro Definition Documentation . . . 389

10.13.1.1 READ . . . 389

10.13.1.2 WRITE . . . 389

10.13.2 Function Documentation . . . 389
10.13.2.1 pclose2(FILE *fp, pid_t pid) . . . 389
10.13.2.2 popen2(std::string command, std::string type, int &pid) . . . 389
10.14client_server/popen2.hh File Reference . . . 389
10.14.1 Function Documentation . . . 389
10.14.1.1 pclose2(FILE *fp, pid_t pid) . . . 389
10.14.1.2 popen2(std::string command, std::string type, int &pid) . . . 389
10.15client_server/regexp_tester.cc File Reference . . . 389
10.15.1 Function Documentation . . . 390
10.15.1.1 main(int argc, char **argv) . . . 390
10.16client_server/Server.cc File Reference . . . 390
10.16.1 Function Documentation . . . 390
10.16.1.1 ProgressMonitor_totals_helper(ProgressMonitor &self) . . . 390
10.16.1.2 quit(void *) . . . 390
10.17client_server/Server.hh File Reference . . . 390
10.18client_server/Snoop.cc File Reference . . . 391
10.18.1 Macro Definition Documentation . . . 392
10.18.1.1 BOOST_SPIRIT_THREADSAFE . . . 392
10.18.2 Typedef Documentation . . . 392
10.18.2.1 u64_millis . . . 392
10.18.3 Function Documentation . . . 392
10.18.3.1 safestring(const unsigned char *c) . . . 392
10.19client_server/Snoop.hh File Reference . . . 392
10.19.1 Macro Definition Documentation . . . 393
10.19.1.1 LOC . . . 393
10.20client_server/SnoopPrivate.hh File Reference . . . 393
10.20.1 Typedef Documentation . . . 393
10.20.1.1 message_ptr . . . 393
10.20.1.2 WebsocketClient . . . 393
10.21client_server/test_client.cc File Reference . . . 393

10.21.1 Function Documentation . . . 394
10.21.1.1 main(int, char **) . . . 394
10.21.2 Variable Documentation . . . 394
10.21.2.1 client . . . 394
10.21.2.2 ui . . . 394
10.22client_server/test_talk_to_server.cc File Reference . . . 394
10.22.1 Typedef Documentation . . . 395
10.22.1.1 client . . . 395
10.22.1.2 message_ptr . . . 395
10.22.2 Function Documentation . . . 395
10.22.2.1 main(int argc, char **argv) . . . 395
10.22.2.2 on_message(client *c, websocketpp::connection_hdl hdl, message_ptr msg) . . . 395
10.22.2.3 on_open(client *c, websocketpp::connection_hdl hdl) . . . 395
10.22.3 Variable Documentation . . . 395
10.22.3.1 stopit . . . 395
10.23core/Algorithm.cc File Reference . . . 395
10.24core/Algorithm.hh File Reference . . . 395
10.25core/algorithms/canonicalise.cc File Reference . . . 396
10.26core/algorithms/canonicalise.hh File Reference . . . 396
10.27core/algorithms/collect_components.cc File Reference . . . 396
10.28core/algorithms/collect_components.hh File Reference . . . 397
10.29core/algorithms/collect_factors.cc File Reference . . . 397
10.30core/algorithms/collect_factors.hh File Reference . . . 397
10.31core/algorithms/collect_terms.cc File Reference . . . 397
10.32core/algorithms/collect_terms.hh File Reference . . . 398
10.33core/algorithms/combine.cc File Reference . . . 398
10.34core/algorithms/combine.hh File Reference . . . 398
10.35core/algorithms/complete.cc File Reference . . . 398
10.36core/algorithms/complete.hh File Reference . . . 399
10.37core/algorithms/component.hh File Reference . . . 399

10.38core/algorithms/decompose_product.cc File Reference . . . 399
10.39core/algorithms/decompose_product.hh File Reference . . . 399
10.40core/algorithms/distribute.cc File Reference . . . 400
10.41core/algorithms/distribute.hh File Reference . . . 400
10.42core/algorithms/drop_weight.cc File Reference . . . 400
10.43core/algorithms/drop_weight.hh File Reference . . . 400
10.44core/algorithms/einsteinify.cc File Reference . . . 401
10.45core/algorithms/einsteinify.hh File Reference . . . 401
10.46core/algorithms/eliminate_kronecker.cc File Reference . . . 401
10.47core/algorithms/eliminate_kronecker.hh File Reference . . . 401
10.48core/algorithms/eliminate_metric.cc File Reference . . . 402
10.49core/algorithms/eliminate_metric.hh File Reference . . . 402
10.50core/algorithms/epsilon_to_delta.cc File Reference . . . 402
10.51core/algorithms/epsilon_to_delta.hh File Reference . . . 402
10.52core/algorithms/evaluate.cc File Reference . . . 403
10.53core/algorithms/evaluate.hh File Reference . . . 403
10.54core/algorithms/expand.cc File Reference . . . 403
10.55core/algorithms/expand.hh File Reference . . . 404
10.56core/algorithms/expand_delta.cc File Reference . . . 404
10.57core/algorithms/expand_delta.hh File Reference . . . 404
10.58core/algorithms/expand_diracbar.cc File Reference . . . 404
10.59core/algorithms/expand_diracbar.hh File Reference . . . 405
10.60core/algorithms/expand_power.cc File Reference . . . 405
10.61core/algorithms/expand_power.hh File Reference . . . 405
10.62core/algorithms/factor_in.cc File Reference . . . 405
10.63core/algorithms/factor_in.hh File Reference . . . 406
10.64core/algorithms/factor_out.cc File Reference . . . 406
10.65core/algorithms/factor_out.hh File Reference . . . 406
10.66core/algorithms/fierz.cc File Reference . . . 406
10.67core/algorithms/fierz.hh File Reference . . . 407

10.68core/algorithms/flatten_product.cc File Reference . . . 407
10.69core/algorithms/flatten_product.hh File Reference . . . 407
10.70core/algorithms/flatten_sum.cc File Reference . . . 407
10.71core/algorithms/flatten_sum.hh File Reference . . . 408
10.72core/algorithms/indexsort.cc File Reference . . . 408
10.73core/algorithms/indexsort.hh File Reference . . . 408
10.74core/algorithms/integrate_by_parts.cc File Reference . . . 408
10.75core/algorithms/integrate_by_parts.hh File Reference . . . 409
10.76core/algorithms/join_gamma.cc File Reference . . . 409
10.77core/algorithms/join_gamma.hh File Reference . . . 409
10.78core/algorithms/keep_terms.cc File Reference . . . 409
10.79core/algorithms/keep_terms.hh File Reference . . . 410
10.80core/algorithms/lr_tensor.cc File Reference . . . 410
10.81core/algorithms/lr_tensor.hh File Reference . . . 410
10.82core/algorithms/map_sympy.cc File Reference . . . 410
10.82.1 Macro Definition Documentation . . . 411
10.82.1.1 DEBUG . . . 411
10.83core/algorithms/map_sympy.hh File Reference . . . 411
10.84core/algorithms/order.cc File Reference . . . 411
10.85core/algorithms/order.hh File Reference . . . 411
10.86core/algorithms/product_rule.cc File Reference . . . 412
10.87core/algorithms/product_rule.hh File Reference . . . 412
10.88core/algorithms/reduce_delta.cc File Reference . . . 412
10.89core/algorithms/reduce_delta.hh File Reference . . . 412
10.90core/algorithms/rename_dummies.cc File Reference . . . 413
10.91core/algorithms/rename_dummies.hh File Reference . . . 413
10.92core/algorithms/replace_match.cc File Reference . . . 413
10.93core/algorithms/replace_match.hh File Reference . . . 413
10.94core/algorithms/rewrite_indices.cc File Reference . . . 414
10.95core/algorithms/rewrite_indices.hh File Reference . . . 414

10.96 core/algorithms/sort_product.cc File Reference 414
10.97 core/algorithms/sort_product.hh File Reference 414
10.98 core/algorithms/sort_spinors.cc File Reference 415
10.99 core/algorithms/sort_spinors.hh File Reference 415
10.100 core/algorithms/sort_sum.cc File Reference 415
10.101 core/algorithms/sort_sum.hh File Reference 415
10.102 core/algorithms/split.cc File Reference 416
10.103 core/algorithms/split.hh File Reference 416
10.104 core/algorithms/split_gamma.cc File Reference 416
10.105 core/algorithms/split_gamma.hh File Reference 416
10.106 core/algorithms/split_index.cc File Reference 416
10.107 core/algorithms/split_index.hh File Reference 417
10.108 core/algorithms/substitute.cc File Reference 417
10.109 core/algorithms/substitute.hh File Reference 417
10.110 core/algorithms/sym.cc File Reference 417
10.111 core/algorithms/sym.hh File Reference 418
10.112 core/algorithms/tab_basics.cc File Reference 418
10.113 core/algorithms/tab_basics.hh File Reference 418
10.114 core/algorithms/take_match.cc File Reference 418
10.115 core/algorithms/take_match.hh File Reference 419
10.116 core/algorithms/unwrap.cc File Reference 419
10.117 core/algorithms/unwrap.hh File Reference 419
10.118 core/algorithms/vary.cc File Reference 420
10.119 core/algorithms/vary.hh File Reference 420
10.120 core/algorithms/young_project.cc File Reference 420
10.121 core/algorithms/young_project.hh File Reference 420
10.122 core/algorithms/young_project_product.cc File Reference 421
10.123 core/algorithms/young_project_product.hh File Reference 421
10.124 core/algorithms/young_project_tensor.cc File Reference 421
10.125 core/algorithms/young_project_tensor.hh File Reference 421

10.126 core/Bridge.cc File Reference . . . 422
10.127 core/Bridge.hh File Reference . . . 422
10.128 core/cadabra2 File Reference . . . 422
10.129 core/cadabra2_defaults.py File Reference . . . 422
10.130 core/cadabra2python.cc File Reference . . . 423
10.130.1 Function Documentation . . . 423
10.130.1.1 main(int argc, char ∗∗argv) . . . 423
10.131 core/CdbPython.cc File Reference . . . 423
10.132 core/CdbPython.hh File Reference . . . 423
10.133 core/Cleanup.cc File Reference . . . 424
10.134 core/Cleanup.hh File Reference . . . 425
10.135 core/CMakeLists.txt File Reference . . . 425
10.135.1 Function Documentation . . . 426
10.135.1.1 cmake_minimum_required(VERSION 2.8) project(Cadabra) set(CMAKE_MOD↩ ULE_PATH $ . . . 426
10.135.1.2 endif() message(""-- Python site path at $ . . . 426
10.135.1.3 install(FILES""${PROJECT_BINARY_DIR}/cadabra2.so"""${PROJECT_SOU↩ RCE_DIR}/cadabra2_defaults.py""DESTINATION""${PYTHON_SITE_PATH}"") configure_file(""$ . . . 426
10.135.1.4 print(site.getsitepackages()[0]) . . . 426
10.136 client_server/CMakeLists.txt File Reference . . . 426
10.136.1 Function Documentation . . . 426
10.136.1.1 cmake_minimum_required(VERSION 2.8) project(CadabraServer) if(POLIC↩ Y CMP0042) cmake_policy(SET CMP0042 NEW) endif(POLICY CMP0042)() set(CMAKE_MODULE_PATH $ . . . 426
10.137 frontend/common/CMakeLists.txt File Reference . . . 426
10.137.1 Function Documentation . . . 426
10.137.1.1 cmake_minimum_required(VERSION 2.8) project(TeXEngine) if(POLICY CM↩ P0042) cmake_policy(SET CMP0042 NEW) endif(POLICY CMP0042) if(CMA↩ KE_COMPILER_IS_GNUCC OR CMAKE_COMPILER_IS_GNUCXX) set(CM↩ AKE_CXX_FLAGS""$ . . . 426
10.138 frontend/gtkmm/CMakeLists.txt File Reference . . . 426
10.139 core/Combinatorics.cc File Reference . . . 426
10.140 core/Combinatorics.hh File Reference . . . 427

10.141 core/Compare.cc File Reference . . . 428
10.141.1 Macro Definition Documentation . . . 429
10.141.1.1 DEBUG . . . 429
10.141.2 Function Documentation . . . 429
10.141.2.1 operator<(const cadabra::Ex::iterator &i1, const cadabra::Ex::iterator &i2) . . . 429
10.141.2.2 operator<(const cadabra::Ex &e1, const cadabra::Ex &e2) . . . 429
10.141.2.3 operator<<(std::ostream &s, cadabra::Ex_comparator::useprops_t up) . . . 429
10.142 core/Compare.hh File Reference . . . 429
10.142.1 Function Documentation . . . 430
10.142.1.1 operator<(const cadabra::Ex::iterator &, const cadabra::Ex::iterator &) . . . 430
10.142.1.2 operator<(const cadabra::Ex &, const cadabra::Ex &) . . . 430
10.142.1.3 operator<<(std::ostream &, cadabra::Ex_comparator::useprops_t up) . . . 430
10.143 core/DisplayBase.cc File Reference . . . 430
10.144 core/DisplayBase.hh File Reference . . . 430
10.145 core/DisplaySympy.cc File Reference . . . 431
10.146 core/DisplaySympy.hh File Reference . . . 431
10.146.1 Typedef Documentation . . . 431
10.146.1.1 kunichar . . . 431
10.147 core/DisplayTerminal.cc File Reference . . . 431
10.148 core/DisplayTerminal.hh File Reference . . . 432
10.148.1 Typedef Documentation . . . 432
10.148.1.1 kunichar . . . 432
10.149 core/DisplayTeX.cc File Reference . . . 432
10.149.1 Macro Definition Documentation . . . 433
10.149.1.1 nbsp . . . 433
10.149.1.2 zwnbsp . . . 433
10.149.2 Function Documentation . . . 433
10.149.2.1 k_unichar_to_utf8(kunichar c, char *buf) . . . 433
10.149.2.2 unichar(kunichar c) . . . 433
10.150 core/DisplayTeX.hh File Reference . . . 433

10.150.1 Typedef Documentation . . . 434
10.150.1.1 kunichar . . . 434
10.151 core/Dummies.hh File Reference . . . 434
10.152 core/echokernel.py File Reference . . . 434
10.153 core/Exceptions.cc File Reference . . . 434
10.154 core/Exceptions.hh File Reference . . . 434
10.155 core/Exchange.cc File Reference . . . 435
10.155.1 Function Documentation . . . 435
10.155.1.1 operator<(const exchange::tensor_type_t &one, const exchange::tensor_type_t &two) . . . 435
10.156 core/Exchange.hh File Reference . . . 435
10.157 core/Functional.cc File Reference . . . 436
10.158 core/Functional.hh File Reference . . . 436
10.159 core/IndexIterator.cc File Reference . . . 437
10.160 core/IndexIterator.hh File Reference . . . 437
10.161 core/Kernel.cc File Reference . . . 438
10.162 core/Kernel.hh File Reference . . . 438
10.163 core/Parser.cc File Reference . . . 438
10.163.1 Function Documentation . . . 438
10.163.1.1 operator>>(std::istream &str, cadabra::Parser &pa) . . . 438
10.164 core/Parser.hh File Reference . . . 438
10.164.1 Function Documentation . . . 439
10.164.1.1 operator>>(std::istream &, cadabra::Parser &) . . . 439
10.165 core/passing.cc File Reference . . . 439
10.165.1 Function Documentation . . . 439
10.165.1.1 fun(int *&p) . . . 439
10.165.1.2 main() . . . 439
10.165.2 Variable Documentation . . . 439
10.165.2.1 k . . . 439
10.166 core/Permutations.hh File Reference . . . 439
10.167 core/PreClean.cc File Reference . . . 440

10.168 core/PreClean.hh File Reference . . . 440
10.169 core/PreProcessor.cc File Reference . . . 441
10.169.1 Function Documentation . . . 441
10.169.1.1 operator<<(std::ostream &s, const preprocessor &p) . . . 441
10.169.1.2 operator>>(std::istream &s, preprocessor &p) . . . 441
10.170 core/PreProcessor.hh File Reference . . . 441
10.170.1 Enumeration Type Documentation . . . 442
10.170.1.1 anonymous enum . . . 442
10.170.2 Function Documentation . . . 442
10.170.2.1 operator<<(std::ostream &, const preprocessor &) . . . 442
10.170.2.2 operator>>(std::istream &, preprocessor &) . . . 442
10.171 core/ProgressMonitor.cc File Reference . . . 442
10.172 core/ProgressMonitor.hh File Reference . . . 442
10.173 core/properties/Accent.cc File Reference . . . 443
10.174 core/properties/Accent.hh File Reference . . . 443
10.175 core/properties/AntiCommuting.cc File Reference . . . 443
10.176 core/properties/AntiCommuting.hh File Reference . . . 443
10.177 core/properties/AntiSymmetric.cc File Reference . . . 444
10.178 core/properties/AntiSymmetric.hh File Reference . . . 444
10.179 core/properties/Commuting.cc File Reference . . . 444
10.180 core/properties/Commuting.hh File Reference . . . 444
10.181 core/properties/CommutingAsProduct.cc File Reference . . . 445
10.182 core/properties/CommutingAsProduct.hh File Reference . . . 445
10.183 core/properties/CommutingAsSum.cc File Reference . . . 445
10.184 core/properties/CommutingAsSum.hh File Reference . . . 445
10.185 core/properties/CommutingBehaviour.cc File Reference . . . 446
10.186 core/properties/CommutingBehaviour.hh File Reference . . . 446
10.187 core/properties/Coordinate.cc File Reference . . . 446
10.188 core/properties/Coordinate.hh File Reference . . . 446
10.189 core/properties/DAntiSymmetric.cc File Reference . . . 446

10.190core/properties/DAntiSymmetric.hh File Reference 447
10.191core/properties/Depends.cc File Reference 447
10.192core/properties/Depends.hh File Reference 447
10.193core/properties/DependsBase.hh File Reference 447
10.194core/properties/DependsInherit.cc File Reference 448
10.195core/properties/DependsInherit.hh File Reference 448
10.196core/properties/Derivative.cc File Reference 448
10.197core/properties/Derivative.hh File Reference 448
10.198core/properties/Diagonal.cc File Reference 449
10.199core/properties/Diagonal.hh File Reference 449
10.200core/properties/DifferentialForm.cc File Reference 449
10.201core/properties/DifferentialForm.hh File Reference 449
10.202core/properties/DifferentialFormBase.hh File Reference 450
10.203core/properties/DiracBar.cc File Reference 450
10.204core/properties/DiracBar.hh File Reference 450
10.205core/properties/Distributable.cc File Reference 451
10.206core/properties/Distributable.hh File Reference 451
10.207core/properties/EpsilonTensor.cc File Reference 451
10.208core/properties/EpsilonTensor.hh File Reference 451
10.209core/properties/ExteriorDerivative.cc File Reference 451
10.210core/properties/ExteriorDerivative.hh File Reference 452
10.211core/properties/FilledTableau.cc File Reference 452
10.212core/properties/FilledTableau.hh File Reference 452
10.213core/properties/GammaMatrix.cc File Reference 452
10.214core/properties/GammaMatrix.hh File Reference 453
10.215core/properties/GammaTraceless.cc File Reference 453
10.216core/properties/GammaTraceless.hh File Reference 453
10.217core/properties/ImaginaryI.cc File Reference 453
10.218core/properties/ImaginaryI.hh File Reference 454
10.219core/properties/ImplicitIndex.cc File Reference 454

10.220core/properties/ImplicitIndex.hh File Reference 454
10.221core/properties/IndexInherit.hh File Reference 454
10.222core/properties/Indices.cc File Reference 455
10.223core/properties/Indices.hh File Reference 455
10.224core/properties/Integer.cc File Reference 455
10.225core/properties/Integer.hh File Reference 455
10.226core/properties/Integral.hh File Reference 456
10.227core/properties/InverseMetric.cc File Reference 456
10.228core/properties/InverseMetric.hh File Reference 456
10.229core/properties/KroneckerDelta.cc File Reference 456
10.230core/properties/KroneckerDelta.hh File Reference 457
10.231core/properties/LaTeXForm.cc File Reference 457
10.232core/properties/LaTeXForm.hh File Reference 457
10.233core/properties/Matrix.cc File Reference 457
10.234core/properties/Matrix.hh File Reference 457
10.235core/properties/Metric.cc File Reference 458
10.236core/properties/Metric.hh File Reference 458
10.237core/properties/NonCommuting.cc File Reference 458
10.238core/properties/NonCommuting.hh File Reference 458
10.239core/properties/NumericalFlat.cc File Reference 459
10.240core/properties/NumericalFlat.hh File Reference 459
10.241core/properties/PartialDerivative.cc File Reference 459
10.242core/properties/PartialDerivative.hh File Reference 459
10.243core/properties/RiemannTensor.cc File Reference 460
10.244core/properties/RiemannTensor.hh File Reference 460
10.245core/properties/SatisfiesBianchi.cc File Reference 460
10.246core/properties/SatisfiesBianchi.hh File Reference 460
10.247core/properties/SelfAntiCommuting.cc File Reference 461
10.248core/properties/SelfAntiCommuting.hh File Reference 461
10.249core/properties/SelfCommuting.cc File Reference 461

10.250core/properties/SelfCommuting.hh File Reference 461
10.251core/properties/SelfCommutingBehaviour.hh File Reference 462
10.252core/properties/SelfNonCommuting.cc File Reference 462
10.253core/properties/SelfNonCommuting.hh File Reference 462
10.254core/properties/SortOrder.cc File Reference 462
10.255core/properties/SortOrder.hh File Reference 462
10.256core/properties/Spinor.cc File Reference 463
10.257core/properties/Spinor.hh File Reference 463
10.258core/properties/Symbol.cc File Reference 463
10.259core/properties/Symbol.hh File Reference 463
10.260core/properties/Symmetric.cc File Reference 464
10.261core/properties/Symmetric.hh File Reference 464
10.262core/properties/Tableau.cc File Reference 464
10.263core/properties/Tableau.hh File Reference 464
10.264core/properties/TableauBase.cc File Reference 465
10.265core/properties/TableauBase.hh File Reference 465
10.266core/properties/TableauSymmetry.cc File Reference 465
10.267core/properties/TableauSymmetry.hh File Reference 465
10.268core/properties/Traceless.cc File Reference 466
10.269core/properties/Traceless.hh File Reference 466
10.270core/properties/Weight.cc File Reference 466
10.271core/properties/Weight.hh File Reference 466
10.272core/properties/WeightBase.hh File Reference 466
10.273core/properties/WeightInherit.cc File Reference 467
10.274core/properties/WeightInherit.hh File Reference 467
10.275core/properties/WeylTensor.cc File Reference 467
10.276core/properties/WeylTensor.hh File Reference 467
10.277core/Props.cc File Reference 468
10.278core/Props.hh File Reference 468
10.279core/ptest.cc File Reference 469

10.279.1 Function Documentation . . . 469
10.279.1.1 BOOST_PYTHON_MODULE(example) . . . 469
10.279.1.2 fun() . . . 469
10.279.1.3 make_x_and_inject_y() . . . 469
10.280 core/PythonCdb.cc File Reference . . . 471
10.280.1 Function Documentation . . . 473
10.280.1.1 BOOST_PYTHON_MODULE(cadabra2) . . . 473
10.280.1.2 construct_Ex_from_int(int num) . . . 473
10.280.1.3 construct_Ex_from_int_2(int num, bool add_ref) . . . 473
10.280.1.4 construct_Ex_from_string(const std::string &ex_) . . . 474
10.280.1.5 construct_Ex_from_string_2(const std::string &ex_, bool add_ref) . . . 474
10.280.1.6 create_empty_scope() . . . 474
10.280.1.7 create_scope_from_global() . . . 474
10.280.1.8 createExceptionClass(const char *name, PyObject *baseTypeObj=PyExc_↩ Exception) . . . 474
10.280.1.9 def_algo_1(const std::string &name) . . . 474
10.280.1.10 def_prop() . . . 474
10.280.1.11 dispatch_base(Ex &ex, F &algo, bool deep, bool repeat, unsigned int depth, bool pre_order) . . . 474
10.280.1.12 dispatch_ex(Ex &ex, bool deep, bool repeat, unsigned int depth) . . . 474
10.280.1.13 dispatch_ex(Ex &ex, Arg1 arg, bool deep, bool repeat, unsigned int depth) . . . 474
10.280.1.14 dispatch_ex(Ex &ex, Arg1 arg1, Arg2 arg2, bool deep, bool repeat, unsigned int depth) . . . 474
10.280.1.15 dispatch_string(const std::string &ex, Args...args, bool deep, bool repeat, unsigned int depth) . . . 474
10.280.1.16 Ex_getitem(Ex &ex, int index) . . . 474
10.280.1.17 Ex_getslice(Ex &ex, boost::python::slice slice) . . . 474
10.280.1.18 Ex_len(Ex &ex) . . . 474
10.280.1.19 Ex_repr_(const Ex &ex) . . . 474
10.280.1.20 Ex_setitem(Ex &ex, int index, Ex val) . . . 474
10.280.1.21 fetch_from_python(const std::string &nm) . . . 474
10.280.1.22 init_ipython() . . . 475

10.280.1.23 free(const Ex &ex) . . . 475
10.280.1.24 list_properties() . . . 475
10.280.1.25 make_Ex_from_int(int num, bool make_ref=true) . . . 475
10.280.1.26 make_Ex_from_string(const std::string &ex_, bool make_ref=true) . . . 475
10.280.1.27 map_sympy_wrapper(Ex &ex, std::string head) . . . 475
10.280.1.28 print_tree(Ex *ex) . . . 475
10.280.1.29 ProgressMonitor_totals_helper(ProgressMonitor &self) . . . 475
10.280.1.30 terms(const Ex &ex) . . . 475
10.280.1.31 terms(const Ex &ex) . . . 475
10.280.1.32 to_std_vector(const boost::python::list &iterable) . . . 475
10.280.1.33 translate_ArgumentException(const ArgumentException &x) . . . 475
10.280.1.34 translate_ConsistencyException(const ConsistencyException &x) . . . 475
10.280.1.35 translate_InternalError(const InternalError &e) . . . 475
10.280.1.36 translate_NonScalarException(const NonScalarException &e) . . . 475
10.280.1.37 translate_NotYetImplemented(const NotYetImplemented &e) . . . 475
10.280.1.38 translate_ParseException(const ParseException &e) . . . 475
10.280.1.39 translate_RuntimeException(const RuntimeException &e) . . . 475
10.280.2 Variable Documentation . . . 475
10.280.2.1 ArgumentExceptionType . . . 475
10.280.2.2 ConsistencyExceptionType . . . 475
10.280.2.3 InternalErrorType . . . 475
10.280.2.4 NonScalarExceptionType . . . 475
10.280.2.5 NotYetImplementedType . . . 476
10.280.2.6 output_ipython . . . 476
10.280.2.7 ParseExceptionType . . . 476
10.280.2.8 pm . . . 476
10.280.2.9 post_process_enabled . . . 476
10.280.2.10 RuntimeExceptionType . . . 476
10.281 core/PythonCdb.hh File Reference . . . 476
10.281.1 Function Documentation . . . 477

10.281.1.1 create_empty_scope() . . . 477
10.281.1.2 create_scope_from_global() . . . 477
10.281.1.3 Ex_repr_(const cadabra::Ex &) . . . 477
10.281.1.4 fetch_from_python(const std::string &nm) . . . 477
10.282 core/PythonException.cc File Reference . . . 477
10.282.1 Function Documentation . . . 477
10.282.1.1 parse_python_exception() . . . 477
10.283 core/PythonException.hh File Reference . . . 477
10.283.1 Function Documentation . . . 478
10.283.1.1 parse_python_exception() . . . 478
10.284 core/ServerWrapper.cc File Reference . . . 478
10.285 core/ServerWrapper.hh File Reference . . . 478
10.286 core/Stopwatch.cc File Reference . . . 478
10.286.1 Function Documentation . . . 478
10.286.1.1 operator<<(std::ostream &os, const Stopwatch &mt) . . . 478
10.287 core/Stopwatch.hh File Reference . . . 478
10.287.1 Function Documentation . . . 479
10.287.1.1 operator<<(std::ostream &, const Stopwatch &) . . . 479
10.288 core/Storage.cc File Reference . . . 479
10.288.1 Function Documentation . . . 480
10.288.1.1 operator<<(std::ostream &str, const cadabra::Ex &ex) . . . 480
10.289 core/Storage.hh File Reference . . . 480
10.289.1 Function Documentation . . . 481
10.289.1.1 operator<<(std::ostream &, const cadabra::Ex &) . . . 481
10.290 core/Symbols.cc File Reference . . . 481
10.291 core/Symbols.hh File Reference . . . 481
10.292 core/SympyCdb.cc File Reference . . . 481
10.293 core/SympyCdb.hh File Reference . . . 481
10.294 core/TerminalStream.cc File Reference . . . 482
10.295 core/TerminalStream.hh File Reference . . . 482

10.296 core/test_permutations.cc File Reference . . . 482
10.296.1 Function Documentation . . . 483
10.296.1.1 main() . . . 483
10.297 core/test_preprocessor.cc File Reference . . . 483
10.297.1 Function Documentation . . . 483
10.297.1.1 main(int argc, char **argv) . . . 483
10.297.1.2 testit(const std::string &input, const std::string &output) . . . 483
10.298 core/YoungTab.cc File Reference . . . 483
10.299 core/YoungTab.hh File Reference . . . 484
10.299.1 Typedef Documentation . . . 485
10.299.1.1 yngint_t . . . 485
10.299.1.2 yngrat_t . . . 485
10.300 doc/main.md File Reference . . . 485
10.301 doc/modules.dox File Reference . . . 485
10.302 frontend/common/exec-stream.cc File Reference . . . 485
10.302.1 Macro Definition Documentation . . . 486
10.302.1.1 HELPERS_CPP . . . 486
10.302.1.2 HELPERS_H . . . 486
10.302.1.3 IMPL_CPP . . . 486
10.303 frontend/common/exec-stream.h File Reference . . . 486
10.304 frontend/common/test_tex.cc File Reference . . . 486
10.304.1 Function Documentation . . . 486
10.304.1.1 main() . . . 486
10.305 frontend/common/TeXEngine.cc File Reference . . . 486
10.306 frontend/common/TeXEngine.hh File Reference . . . 487
10.307 frontend/gtkmm/Cadabra.cc File Reference . . . 487
10.307.1 Function Documentation . . . 488
10.307.1.1 signal_handler(int signal) . . . 488
10.307.2 Variable Documentation . . . 488
10.307.2.1 signal_window . . . 488

10.308 frontend/gtkmm/Cadabra.hh File Reference . . . 488
10.309 frontend/gtkmm/CodeInput.cc File Reference . . . 488
10.309.1 Function Documentation . . . 488
10.309.1.1 trim(const std::string &s) . . . 488
10.310 frontend/gtkmm/CodeInput.hh File Reference . . . 488
10.311 frontend/gtkmm/ImageView.cc File Reference . . . 489
10.312 frontend/gtkmm/ImageView.hh File Reference . . . 489
10.313 frontend/gtkmm/main.cc File Reference . . . 489
10.313.1 Function Documentation . . . 490
10.313.1.1 main(int argc, char **argv) . . . 490
10.314 frontend/gtkmm/NotebookCanvas.cc File Reference . . . 490
10.315 frontend/gtkmm/NotebookCanvas.hh File Reference . . . 490
10.316 frontend/osx/Cadabra/Cadabra/NotebookCanvas.hh File Reference . . . 490
10.317 frontend/gtkmm/NotebookWindow.cc File Reference . . . 491
10.318 frontend/gtkmm/NotebookWindow.hh File Reference . . . 491
10.319 frontend/osx/Cadabra/Cadabra/NotebookWindow.hh File Reference . . . 492
10.320 frontend/gtkmm/Preferences.hh File Reference . . . 492
10.321 frontend/gtkmm/TeXView.cc File Reference . . . 492
10.322 frontend/gtkmm/TeXView.hh File Reference . . . 492
10.323 frontend/gtkmm/VisualCell.hh File Reference . . . 493
10.324 frontend/osx/Cadabra/Cadabra/AppDelegate.h File Reference . . . 493
10.325 frontend/osx/Cadabra/Cadabra/AppDelegate.mm File Reference . . . 493
10.326 frontend/osx/Cadabra/Cadabra/main.m File Reference . . . 494
10.326.1 Function Documentation . . . 494
10.326.1.1 main(int argc, const char *argv[]) . . . 494
10.327 frontend/osx/Cadabra/Cadabra/NotebookCanvas.mm File Reference . . . 494
10.328 frontend/osx/Cadabra/Cadabra/NotebookWindow.mm File Reference . . . 494
10.329 frontend/osx/Cadabra/NotebookController.h File Reference . . . 494
10.330 frontend/osx/Cadabra/NotebookController.mm File Reference . . . 494
10.330.1 Variable Documentation . . . 495
10.330.1.1 compute . . . 495
10.330.1.2 compute_thread . . . 495

Index **497**

Chapter 1

Cadabra

Author

Kasper Peeters

Version

2.x series

See also

http://cadabra.science

This repository holds the 2.x series of the Cadabra computer algebra system.

Cadabra was designed specifically for the solution of problems encountered in field theory. It has extensive functionality for tensor computer algebra, tensor polynomial simplification including multi-term symmetries, fermions and anti-commuting variables, Clifford algebras and Fierz transformations, implicit coordinate dependence, multiple index types and many more. The input format is a subset of TeX. Both a command-line and a graphical interface are available.

To get started with the source code documentation, navigate to the modules overview.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Core 25
Cleanup 28
Expression comparison 29
Python Core 31
Scalar algebra 34
Algorithms 35
Properties 36
Client-Server 37
Front-end 38
Gtk+ 39
OS-X 40

Chapter 3

Namespace Index

3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

cadabra
    Class to display expressions in a format that Sympy can parse . . . 41
cadabra2_defaults . . . 53
cadabra::symbols . . . 54
cadabra_osx . . . 55
combin . . . 55
echokernel . . . 57
snoop . . . 57
sympy . . . 58
yngtab
    Generic Young tableaux routines . . . 58

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

preprocessor::accu_t . . . 62
cadabra::ActionBase . . . 65
  cadabra::ActionAddCell . . . 63
  cadabra::ActionEraseText . . . 66
  cadabra::ActionInsertText . . . 67
  cadabra::ActionPositionCursor . . . 69
  cadabra::ActionRemoveCell . . . 70
  cadabra::ActionSetRunStatus . . . 72
  cadabra::ActionSplitCell . . . 73
cadabra::Algorithm . . . 75
  cadabra::canonicalise . . . 94
  cadabra::collect_components . . . 99
  cadabra::collect_factors . . . 100
  cadabra::collect_terms . . . 101
  cadabra::combine . . . 108
  cadabra::complete . . . 112
  cadabra::decompose_product . . . 123
  cadabra::distribute . . . 146
  cadabra::drop_keep_weight . . . 151
    cadabra::drop_weight . . . 153
    cadabra::keep_weight . . . 232
  cadabra::einsteinify . . . 154
  cadabra::eliminate_converter . . . 155
    cadabra::eliminate_metric . . . 158
  cadabra::eliminate_kronecker . . . 157
  cadabra::epsilon_to_delta . . . 159
  cadabra::evaluate . . . 162
  cadabra::expand . . . 183
  cadabra::expand_delta . . . 184
  cadabra::expand_diracbar . . . 185
  cadabra::expand_power . . . 186
  cadabra::factor_in . . . 188
  cadabra::factor_out . . . 189
  cadabra::fierz . . . 191
  cadabra::flatten_product . . . 198

cadabra::flatten_sum . . . 199
cadabra::indexsort . . . 216
cadabra::integrate_by_parts . . . 222
cadabra::join_gamma . . . 229
cadabra::keep_terms . . . 231
cadabra::map_sympy . . . 243
cadabra::order . . . 265
cadabra::product_rule . . . 278
cadabra::reduce_delta . . . 290
cadabra::rename_dummies . . . 292
cadabra::replace_match . . . 293
cadabra::rewrite_indices . . . 294
cadabra::sort_product . . . 315
cadabra::sort_spinors . . . 316
cadabra::sort_sum . . . 317
cadabra::split_gamma . . . 321
cadabra::split_index . . . 322
cadabra::substitute . . . 329
cadabra::sym . . . 330
cadabra::tab_basics . . . 339
cadabra::lr_tensor . . . 242
cadabra::take_match . . . 349
cadabra::unwrap . . . 368
cadabra::vary . . . 369
cadabra::young_project . . . 377
cadabra::young_project_product . . . 379
cadabra::young_project_tensor . . . 379

Algorithm
component . . . 113
split . . . 320

algorithm
rename_dummies . . . 291

AppDelegate() . . . 87
snoop::Snoop::AppEntry . . . 87

Application
Cadabra . . . 91

BaseProperty . . . 89
Property< T > . . . 284

ProgressMonitor::Block . . . 90
Server::Block . . . 90
Server::CatchOutput . . . 95

Client
MyClient . . . 247

combin::combinations_base< T > . . . 105
combin::combinations< T > . . . 103
combin::symm_helper< T > . . . 332
combin::symm_val_helper< T > . . . 334

combin::combinations_base< unsigned int > . . . 105
combin::combinations< unsigned int > . . . 103
combin::symm_helper< unsigned int > . . . 332
combin::symm_val_helper< unsigned int > . . . 334

cadabra::ComputeThread . . . 114
Server::Connection . . . 117
cadabra::DataCell . . . 120
cadabra::DisplayBase . . . 133
cadabra::DisplaySympy . . . 135
cadabra::DisplayTerminal . . . 138

cadabra::DisplayTeX . . . 141
cadabra::DocumentThread . . . 147
cadabra::NotebookWindow . . . 255
DocumentThread
cadabra_osx::NotebookWindow . . . 253
EventBox
cadabra::TeXView . . . 356
cadabra::Ex_comparator . . . 171
cadabra::Ex_is_equivalent . . . 176
cadabra::Ex_is_less . . . 177
exception
exec_stream_t::error_t . . . 161
cadabra::exchange . . . 177
exec_stream_t . . . 178
cadabra::str_node::flag_t . . . 198
snoop::Flush . . . 200
cadabra::GUIBase . . . 203
cadabra::NotebookWindow . . . 255
GUIBase
cadabra_osx::NotebookWindow . . . 253
cadabra::DataCell::id_t . . . 205
cadabra::exchange::identical_tensors_t . . . 206
Images . . . 207
cadabra::Inherit< T > . . . 219
cadabra::Inherit< SortOrder > . . . 219
cadabra::Derivative . . . 129
cadabra::ExteriorDerivative . . . 187
cadabra::PartialDerivative . . . 270
iterable_converter . . . 226
iterator_base
cadabra::index_iterator . . . 213
yngtab::filled_tableau< T >::iterator_base . . . 229
yngtab::filled_tableau< T >::in_column_iterator . . . 211
yngtab::filled_tableau< T >::iterator . . . 227
cadabra::Kernel . . . 233
cadabra::keyval_t . . . 234
cadabra::indexsort::less_indexed_treenode . . . 238
snoop::Snoop::LogEntry . . . 240
logic_error
cadabra::TeXEngine::TeXException . . . 354
CadabraException . . . 93
ArgumentException . . . 89
ConsistencyException . . . 118
cadabra::WeightInherit::WeightException . . . 373
InternalError . . . 224
InterruptionException . . . 224
NonScalarException . . . 250
NotYetImplemented . . . 263
ParseException . . . 267
RuntimeException . . . 296
PermutationException . . . 273
exec_stream_t::next_arg_t . . . 249
exec_stream_t::next_arg_impl_t< iterator > . . . 248
NotebookController() . . . 253
<NSApplicationDelegate>
AppDelegate . . . 86

cadabra::nset_it_less . . . 264
NSObject
  AppDelegate . . . 86
NSViewController
  NotebookController . . . 252
cadabra::Parser . . . 267
cadabra::pattern . . . 271
Perm . . . 272
cadabra::NotebookWindow::Prefs . . . 274
preprocessor . . . 274
ProgressMonitor . . . 279
  cadabra2_defaults.Server . . . 301
  Server . . . 302
  ServerWrapper . . . 307
cadabra::Properties . . . 281
property
  Integral . . . 222
cadabra::property . . . 286
  cadabra::Accent . . . 61
    cadabra::DiracBar . . . 133
  cadabra::AntiSymmetric . . . 85
    cadabra::EpsilonTensor . . . 160
    cadabra::GammaMatrix . . . 201
  cadabra::CommutingAsProduct . . . 110
    cadabra::Derivative . . . 129
  cadabra::CommutingAsSum . . . 111
  cadabra::Coordinate . . . 119
  cadabra::DAntiSymmetric . . . 119
  cadabra::Depends . . . 126
  cadabra::DependsBase . . . 127
    cadabra::Depends . . . 126
    cadabra::DependsInherit . . . 128
      cadabra::Derivative . . . 129
  cadabra::DependsInherit . . . 128
  cadabra::Derivative . . . 129
  cadabra::DifferentialFormBase . . . 132
    cadabra::DifferentialForm . . . 131
    cadabra::ExteriorDerivative . . . 187
  cadabra::DiracBar . . . 133
  cadabra::Distributable . . . 145
    cadabra::Derivative . . . 129
    cadabra::DiracBar . . . 133
  cadabra::EpsilonTensor . . . 160
  cadabra::FilledTableau . . . 197
  cadabra::GammaMatrix . . . 201
  cadabra::GammaTraceless . . . 202
  cadabra::ImaginaryI . . . 209
  cadabra::ImplicitIndex . . . 209
    cadabra::DifferentialForm . . . 131
    cadabra::Matrix . . . 244
      cadabra::GammaMatrix . . . 201
    cadabra::Spinor . . . 319
  cadabra::IndexInherit . . . 215
    cadabra::Accent . . . 61
    cadabra::Derivative . . . 129
    cadabra::DifferentialForm . . . 131
  cadabra::Integer . . . 220

cadabra::InverseMetric . . . 225
cadabra::KroneckerDelta . . . 235
cadabra::labelled_property . . . 236
cadabra::WeightBase . . . 372
cadabra::Derivative . . . 129
cadabra::Weight . . . 371
cadabra::WeightInherit . . . 374
cadabra::LaTeXForm . . . 237
cadabra::list_property . . . 239
cadabra::CommutingBehaviour . . . 112
cadabra::AntiCommuting . . . 84
cadabra::Commuting . . . 109
cadabra::NonCommuting . . . 249
cadabra::Indices . . . 217
cadabra::SortOrder . . . 318
cadabra::Matrix . . . 244
cadabra::Metric . . . 245
cadabra::NumericalFlat . . . 265
cadabra::Accent . . . 61
cadabra::Derivative . . . 129
cadabra::PartialDerivative . . . 270
cadabra::PropertyInherit . . . 289
cadabra::Accent . . . 61
cadabra::RiemannTensor . . . 295
cadabra::SatisfiesBianchi . . . 297
cadabra::SelfCommutingBehaviour . . . 299
cadabra::SelfAntiCommuting . . . 298
cadabra::SelfCommuting . . . 299
cadabra::SelfNonCommuting . . . 300
cadabra::Spinor . . . 319
cadabra::Symbol . . . 332
cadabra::Symmetric . . . 335
cadabra::Diagonal . . . 130
cadabra::Metric . . . 245
cadabra::Tableau . . . 340
cadabra::TableauSymmetry . . . 346
cadabra::InverseMetric . . . 225
cadabra::RiemannTensor . . . 295
cadabra::WeylTensor . . . 375
cadabra::Traceless . . . 360
cadabra::AntiSymmetric . . . 85
cadabra::WeylTensor . . . 375
cadabra::WeylTensor . . . 375
cadabra::Properties::registered_property_map_t . . . 291
snoop::Snoop . . . 308
snoop::SnoopImpl . . . 310
Stopwatch . . . 324
cadabra::str_node . . . 325
combin::symmetriser< T > . . . 336
combin::symmetriser< unsigned int > . . . 336
cadabra2_defaults.Sympy . . . 338
yngtab::tableau_base . . . 343
yngtab::tableau . . . 341
yngtab::filled_tableau< iterator > . . . 193
yngtab::filled_tableau< unsigned int > . . . 193
yngtab::filled_tableau< T > . . . 193
cadabra::TableauBase . . . 345

cadabra::AntiSymmetric . . . 85
cadabra::DAntiSymmetric . . . 119
cadabra::Derivative . . . 129
cadabra::KroneckerDelta . . . 235
cadabra::SatisfiesBianchi . . . 297
cadabra::Symmetric . . . 335
cadabra::TableauSymmetry . . . 346
yngtab::tableaux< T > . . . 348
cadabra::exchange::tensor_type_t . . . 350
cadabra::TerminalStream . . . 351
cadabra::TeXEngine . . . 352
cadabra::TeXEngine::TeXRequest . . . 355
TextView
cadabra::CodeInput::exp_input_tv . . . 181
ProgressMonitor::Total . . . 358
tree
cadabra::Ex . . . 165
cadabra::tree_equal_obj . . . 360
cadabra::tree_exact_equal_mod_prel_obj . . . 361
cadabra::tree_exact_equal_obj . . . 362
cadabra::tree_exact_less_for_indexmap_obj . . . 363
cadabra::tree_exact_less_mod_prel_obj . . . 363
cadabra::tree_exact_less_no_wildcards_mod_prel_obj . . . 364
cadabra::tree_exact_less_no_wildcards_obj . . . 365
cadabra::tree_exact_less_obj . . . 365
cadabra::tree_less_modprel_obj . . . 366
cadabra::tree_less_obj . . . 367
UI . . . 368
VBox
cadabra::CodeInput . . . 96
cadabra::ImageView . . . 208
cadabra::VisualCell . . . 370
VPaned
cadabra::NotebookCanvas . . . 251
Window
cadabra::NotebookWindow . . . 255
Window
PreferencesWindow . . . 273
X . . . 376
Y . . . 377
Z . . . 381
Kernel
echokernel.EchoKernel . . . 153

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

cadabra::Accent
: Turns a symbol into an accent . . . 61

preprocessor::accu_t . . . 62

cadabra::ActionAddCell
: Add a cell to the notebook . . . 63

cadabra::ActionBase
: All actions derive from the ActionBase object, which defines the interface they need to implement 65

cadabra::ActionEraseText
: Remove a text string starting at the indicated position, and with the indicated length, from the indicated cell . . . 66

cadabra::ActionInsertText
: Add a text string (can be just a single character) at the point of the cursor . . . 67

cadabra::ActionPositionCursor
: Position the cursor relative to the indicated cell . . . 69

cadabra::ActionRemoveCell
: Remove a cell and all its child cells from the document . . . 70

cadabra::ActionSetRunStatus
: Update the running status of the indicated cell . . . 72

cadabra::ActionSplitCell
: Split a cell into two separate cells, at the point of the cursor . . . 73

cadabra::Algorithm
: Base class for all algorithms, containing generic routines and in particular the logic for index classification . . . 75

cadabra::AntiCommuting . . . 84

cadabra::AntiSymmetric . . . 85

AppDelegate . . . 86

AppDelegate() . . . 87

snoop::Snoop::AppEntry
: C++ representation of an run entry . . . 87

ArgumentException
: Exception thrown when arguments to an algorithm or property are not correct . . . 89

BaseProperty
: Helper class to ensure that all Python property objects derive from the same base class . . . 89

ProgressMonitor::Block . . . 90

Server::Block . . . 90

Cadabra
The Cadabra notebook application . . . 91
CadabraException . . . 93
cadabra::canonicalise
Canonicalise the index structure of a tensorial expression . . . 94
Server::CatchOutput
Python output catching . . . 95
cadabra::CodeInput
A text cell editor widget with support for editing Python and LaTeX . . . 96
cadabra::collect_components
Collect terms inside a sum, merging their substitution rules . . . 99
cadabra::collect_factors
Collect symbolically equal factors in a product . . . 100
cadabra::collect_terms
Collect symbolically equal terms in a sum . . . 101
combin::combinations< T > . . . 103
combin::combinations_base< T > . . . 105
cadabra::combine . . . 108
cadabra::Commuting . . . 109
cadabra::CommutingAsProduct . . . 110
cadabra::CommutingAsSum . . . 111
cadabra::CommutingBehaviour . . . 112
cadabra::complete
Complete a set of coordinate rules so that they also cover related tensors . . . 112
component . . . 113
cadabra::ComputeThread
Base class which talks to the server and sends Action objects back to the DocumentThread . . . 114
Server::Connection . . . 117
ConsistencyException . . . 118
cadabra::Coordinate . . . 119
cadabra::DAntiSymmetric . . . 119
cadabra::DataCell
DataCells are the basic building blocks for a document . . . 120
cadabra::decompose_product
Decompose a product of tensors by applying Young projectors . . . 123
cadabra::Depends . . . 126
cadabra::DependsBase . . . 127
cadabra::DependsInherit . . . 128
cadabra::Derivative . . . 129
cadabra::Diagonal . . . 130
cadabra::DifferentialForm . . . 131
cadabra::DifferentialFormBase . . . 132
cadabra::DiracBar . . . 133
cadabra::DisplayBase
Base class for all display classes . . . 133
cadabra::DisplaySympy . . . 135
cadabra::DisplayTerminal . . . 138
cadabra::DisplayTeX . . . 141
cadabra::Distributable . . . 145
cadabra::distribute
Distribute factors over a sum, that is, $(A+B)C \rightarrow AC+BC$ . . . 146
cadabra::DocumentThread
A base class with all the logic to manipulate a Cadabra notebook document . . . 147
cadabra::drop_keep_weight . . . 151
cadabra::drop_weight . . . 153
echokernel.EchoKernel . . . 153
cadabra::einsteinify . . . 154
cadabra::eliminate_converter . . . 155

cadabra::eliminate_kronecker . . . 157
cadabra::eliminate_metric . . . 158
cadabra::epsilon_to_delta . . . 159
cadabra::EpsilonTensor . . . 160
exec_stream_t::error_t . . . 161
cadabra::evaluate . . . 162
cadabra::Ex
Basic storage class for symbolic mathemematical expressions . . . 165
cadabra::Ex_comparator
A generic tree comparison class which will take into account index contractions and will also keep track of a replacement list for all types of cadabra wildcards . . . 171
cadabra::Ex_is_equivalent
Basic comparison operator for tree iterators, so we can use them as keys in maps . . . 176
cadabra::Ex_is_less . . . 177
cadabra::exchange . . . 177
exec_stream_t . . . 178
cadabra::CodeInput::exp_input_tv
The actual text widget used by CodeInput . . . 181
cadabra::expand . . . 183
cadabra::expand_delta . . . 184
cadabra::expand_diracbar . . . 185
cadabra::expand_power . . . 186
cadabra::ExteriorDerivative . . . 187
cadabra::factor_in . . . 188
cadabra::factor_out . . . 189
cadabra::fierz . . . 191
yngtab::filled_tableau< T > . . . 193
cadabra::FilledTableau . . . 197
cadabra::str_node::flag_t . . . 198
cadabra::flatten_product . . . 198
cadabra::flatten_sum . . . 199
snoop::Flush . . . 200
cadabra::GammaMatrix . . . 201
cadabra::GammaTraceless . . . 202
cadabra::GUIBase
Abstract base class with methods that need to be implemented by any GUI . . . 203
cadabra::DataCell::id_t
Each cell is identified by a serial number 'id' which is used to keep track of it across network calls, and a bool indicating whether the client or the server has created this cell . . . 205
cadabra::exchange::identical_tensors_t . . . 206
Images . . . 207
cadabra::ImageView
An image viewing widget . . . 208
cadabra::ImaginaryI . . . 209
cadabra::ImplicitIndex . . . 209
yngtab::filled_tableau< T >::in_column_iterator
An iterator which stays inside a given column of a tableau . . . 211
cadabra::index_iterator
An iterator which iterates over indices even if they are at lower levels, i.e . . . 213
cadabra::IndexInherit . . . 215
cadabra::indexsort . . . 216
cadabra::Indices . . . 217
cadabra::Inherit< T >
If a property X derives from Inherit<Y>, and get<Y> is called on an object which has an X property (but no Y property), then the get<Y> will look at the non-index child of the object to see if that has a Y property . . . 219
cadabra::Integer
Property indicating that a symbolic object always takes integer values . . . 220

Integral . . . 222
cadabra::integrate_by_parts
Integrate by parts away from the indicated derivative object . . . 222
InternalError
Exception thrown when there the Cadabra code detects an internal inconsistency; this should never happen, but is better than bailing out with an assert . . . 224
InterruptionException . . . 224
cadabra::InverseMetric . . . 225
iterable_converter . . . 226
yngtab::filled_tableau< T >::iterator
An iterator over all boxes of a tableau, left to right, top to bottom . . . 227
yngtab::filled_tableau< T >::iterator_base . . . 229
cadabra::join_gamma . . . 229
cadabra::keep_terms . . . 231
cadabra::keep_weight . . . 232
cadabra::Kernel . . . 233
cadabra::keyval_t
Arguments to properties get parsed into a keyval_t structure . . . 234
cadabra::KroneckerDelta . . . 235
cadabra::labelled_property . . . 236
cadabra::LaTeXForm . . . 237
cadabra::indexsort::less_indexed_treenode . . . 238
cadabra::list_property
Something cannot be both a list property and a normal property at the same time, so we can safely inherit without virtual . . . 239
snoop::Snoop::LogEntry
C++ representation of a log entry . . . 240
cadabra::lr_tensor . . . 242
cadabra::map_sympy
Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place 243
cadabra::Matrix . . . 244
cadabra::Metric . . . 245
MyClient . . . 247
exec_stream_t::next_arg_impl_t< iterator > . . . 248
exec_stream_t::next_arg_t . . . 249
cadabra::NonCommuting . . . 249
NonScalarException
Exception thrown when something requires that an expression is a pure scalar (i.e . . . 250
cadabra::NotebookCanvas . . . 251
NotebookController . . . 252
NotebookController() . . . 253
cadabra_osx::NotebookWindow
Objective-C++ class implementing DocumentThread and providing an OS-X notebook interface 253
cadabra::NotebookWindow
Each notebook has one main window which controls it . . . 255
NotYetImplemented
Exception thrown when there the Cadabra code detects an internal inconsistency; this should never happen, but is better than bailing out with an assert . . . 263
cadabra::nset_it_less
Compare two nset iterators by comparing the strings to which they point . . . 264
cadabra::NumericalFlat
Property indicating that an operator is numerically flat, so that numerical factors in the argument can be taken outside . . . 265
cadabra::order . . . 265
ParseException . . . 267
cadabra::Parser . . . 267
cadabra::PartialDerivative . . . 270
cadabra::pattern . . . 271

Perm . . . 272
PermutationException
Generic permutation group material . . . 273
PreferencesWindow . . . 273
cadabra::NotebookWindow::Prefs . . . 274
preprocessor
Preprocessing class which takes infix mathematical notation with all sorts of maths shortcuts and transforms it into a string which is properly formatted in prefix notation . . . 274
cadabra::product_rule . . . 278
ProgressMonitor . . . 279
cadabra::Properties
Class holding a collection of properties attached to expressions . . . 281
Property< T >
Property is a templated wrapper around a C++ property object . . . 284
cadabra::property
Base class for all properties, handling argument parsing and defining the interface . . . 286
cadabra::PropertyInherit
PropertyInherit is like Inherit<T> for all properties . . . 289
cadabra::reduce_delta . . . 290
cadabra::Properties::registered_property_map_t . . . 291
rename_dummies . . . 291
cadabra::rename_dummies . . . 292
cadabra::replace_match . . . 293
cadabra::rewrite_indices . . . 294
cadabra::RiemannTensor . . . 295
RuntimeException
Exception thrown when an algorithm cannot complete, e.g . . . 296
cadabra::SatisfiesBianchi . . . 297
cadabra::SelfAntiCommuting . . . 298
cadabra::SelfCommuting . . . 299
cadabra::SelfCommutingBehaviour . . . 299
cadabra::SelfNonCommuting . . . 300
cadabra2_defaults.Server . . . 301
Server
Object representing a Cadabra server, capable of receiving messages on a websocket, running Python code, and sending output back to the client . . . 302
ServerWrapper
A class which enables access to a Python-side defined Server object with ProgressMonitor interface . . . 307
snoop::Snoop
Logging class with functionality to send log information to a remote server using a websocket connection . . . 308
snoop::SnoopImpl . . . 310
cadabra::sort_product . . . 315
cadabra::sort_spinors . . . 316
cadabra::sort_sum . . . 317
cadabra::SortOrder . . . 318
cadabra::Spinor . . . 319
split
Split an expression into terms or factors . . . 320
cadabra::split_gamma . . . 321
cadabra::split_index . . . 322
Stopwatch . . . 324
cadabra::str_node
Elementary building block for a mathematical expression . . . 325
cadabra::substitute
Generic substitution algorithm . . . 329
cadabra::sym . . . 330

cadabra::Symbol . . . 332
combin::symm_helper< T > . . . 332
combin::symm_val_helper< T > . . . 334
cadabra::Symmetric . . . 335
combin::symmetriser< T > . . . 336
cadabra2_defaults.Sympy . . . 338
cadabra::tab_basics . . . 339
cadabra::Tableau . . . 340
yngtab::tableau . . . 341
yngtab::tableau_base . . . 343
cadabra::TableauBase . . . 345
cadabra::TableauSymmetry . . . 346
yngtab::tableaux< T > . . . 348
cadabra::take_match . . . 349
cadabra::exchange::tensor_type_t . . . 350
cadabra::TerminalStream . . . 351
cadabra::TeXEngine
TeXEngine is used to convert LaTeX strings into PNG images . . . 352
cadabra::TeXEngine::TeXException . . . 354
cadabra::TeXEngine::TeXRequest . . . 355
cadabra::TeXView
TeXView is a widget which knows how to turn a string into a LaTeX-rendered image and display that . . . 356
ProgressMonitor::Total . . . 358
cadabra::Traceless . . . 360
cadabra::tree_equal_obj . . . 360
cadabra::tree_exact_equal_mod_prel_obj . . . 361
cadabra::tree_exact_equal_obj . . . 362
cadabra::tree_exact_less_for_indexmap_obj
Compare for indexmap_t . . . 363
cadabra::tree_exact_less_mod_prel_obj . . . 363
cadabra::tree_exact_less_no_wildcards_mod_prel_obj . . . 364
cadabra::tree_exact_less_no_wildcards_obj
Compare two trees exactly, treat wildcard names as ordinary names . . . 365
cadabra::tree_exact_less_obj
Compare two trees exactly, i.e. including exact index names . . . 365
cadabra::tree_less_modprel_obj . . . 366
cadabra::tree_less_obj
Compare two trees by pattern logic, i.e. modulo index names . . . 367
UI . . . 368
cadabra::unwrap . . . 368
cadabra::vary
Generic variational derivative algorithm . . . 369
cadabra::VisualCell
Structure holding a pointer to one of the possible GUI widgets that can appear in a document . . . 370
cadabra::Weight . . . 371
cadabra::WeightBase . . . 372
cadabra::WeightInherit::WeightException . . . 373
cadabra::WeightInherit . . . 374
cadabra::WeylTensor . . . 375
X
Mockup types . . . 376
Y . . . 377
cadabra::young_project . . . 377
cadabra::young_project_product . . . 379
cadabra::young_project_tensor . . . 379
Z . . . 381

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

client_server/Actions.cc . . . 383
client_server/Actions.hh . . . 383
client_server/cadabra-server.cc . . . 384
client_server/cadabra2cadabra.cc . . . 384
client_server/cadabra2html.cc . . . 384
client_server/ComputeThread.cc . . . 385
client_server/ComputeThread.hh . . . 385
client_server/DataCell.cc . . . 386
client_server/DataCell.hh . . . 386
client_server/DocumentThread.cc . . . 387
client_server/DocumentThread.hh . . . 388
client_server/GUIBase.hh . . . 388
client_server/popen2.cc . . . 388
client_server/popen2.hh . . . 389
client_server/regexp_tester.cc . . . 389
client_server/Server.cc . . . 390
client_server/Server.hh . . . 390
client_server/Snoop.cc . . . 391
client_server/Snoop.hh . . . 392
client_server/SnoopPrivate.hh . . . 393
client_server/test_client.cc . . . 393
client_server/test_talk_to_server.cc . . . 394
core/Algorithm.cc . . . 395
core/Algorithm.hh . . . 395
core/Bridge.cc . . . 422
core/Bridge.hh . . . 422
core/cadabra2 . . . 422
core/cadabra2_defaults.py . . . 422
core/cadabra2python.cc . . . 423
core/CdbPython.cc . . . 423
core/CdbPython.hh . . . 423
core/Cleanup.cc . . . 424
core/Cleanup.hh . . . 425
core/Combinatorics.cc . . . 426
core/Combinatorics.hh . . . 427

core/Compare.cc . . . 428
core/Compare.hh . . . 429
core/DisplayBase.cc . . . 430
core/DisplayBase.hh . . . 430
core/DisplaySympy.cc . . . 431
core/DisplaySympy.hh . . . 431
core/DisplayTerminal.cc . . . 431
core/DisplayTerminal.hh . . . 432
core/DisplayTeX.cc . . . 432
core/DisplayTeX.hh . . . 433
core/Dummies.hh . . . 434
core/echokernel.py . . . 434
core/Exceptions.cc . . . 434
core/Exceptions.hh . . . 434
core/Exchange.cc . . . 435
core/Exchange.hh . . . 435
core/Functional.cc . . . 436
core/Functional.hh . . . 436
core/IndexIterator.cc . . . 437
core/IndexIterator.hh . . . 437
core/Kernel.cc . . . 438
core/Kernel.hh . . . 438
core/Parser.cc . . . 438
core/Parser.hh . . . 438
core/passing.cc . . . 439
core/Permutations.hh . . . 439
core/PreClean.cc . . . 440
core/PreClean.hh . . . 440
core/PreProcessor.cc . . . 441
core/PreProcessor.hh . . . 441
core/ProgressMonitor.cc . . . 442
core/ProgressMonitor.hh . . . 442
core/Props.cc . . . 468
core/Props.hh . . . 468
core/ptest.cc . . . 469
core/PythonCdb.cc . . . 471
core/PythonCdb.hh . . . 476
core/PythonException.cc . . . 477
core/PythonException.hh . . . 477
core/ServerWrapper.cc . . . 478
core/ServerWrapper.hh . . . 478
core/Stopwatch.cc . . . 478
core/Stopwatch.hh . . . 478
core/Storage.cc . . . 479
core/Storage.hh . . . 480
core/Symbols.cc . . . 481
core/Symbols.hh . . . 481
core/SympyCdb.cc . . . 481
core/SympyCdb.hh . . . 481
core/TerminalStream.cc . . . 482
core/TerminalStream.hh . . . 482
core/test_permutations.cc . . . 482
core/test_preprocessor.cc . . . 483
core/YoungTab.cc . . . 483
core/YoungTab.hh . . . 484
core/algorithms/canonicalise.cc . . . 396
core/algorithms/canonicalise.hh . . . 396
core/algorithms/collect_components.cc . . . 396

core/algorithms/collect_components.hh . . . 397
core/algorithms/collect_factors.cc . . . 397
core/algorithms/collect_factors.hh . . . 397
core/algorithms/collect_terms.cc . . . 397
core/algorithms/collect_terms.hh . . . 398
core/algorithms/combine.cc . . . 398
core/algorithms/combine.hh . . . 398
core/algorithms/complete.cc . . . 398
core/algorithms/complete.hh . . . 399
core/algorithms/component.hh . . . 399
core/algorithms/decompose_product.cc . . . 399
core/algorithms/decompose_product.hh . . . 399
core/algorithms/distribute.cc . . . 400
core/algorithms/distribute.hh . . . 400
core/algorithms/drop_weight.cc . . . 400
core/algorithms/drop_weight.hh . . . 400
core/algorithms/einsteinify.cc . . . 401
core/algorithms/einsteinify.hh . . . 401
core/algorithms/eliminate_kronecker.cc . . . 401
core/algorithms/eliminate_kronecker.hh . . . 401
core/algorithms/eliminate_metric.cc . . . 402
core/algorithms/eliminate_metric.hh . . . 402
core/algorithms/epsilon_to_delta.cc . . . 402
core/algorithms/epsilon_to_delta.hh . . . 402
core/algorithms/evaluate.cc . . . 403
core/algorithms/evaluate.hh . . . 403
core/algorithms/expand.cc . . . 403
core/algorithms/expand.hh . . . 404
core/algorithms/expand_delta.cc . . . 404
core/algorithms/expand_delta.hh . . . 404
core/algorithms/expand_diracbar.cc . . . 404
core/algorithms/expand_diracbar.hh . . . 405
core/algorithms/expand_power.cc . . . 405
core/algorithms/expand_power.hh . . . 405
core/algorithms/factor_in.cc . . . 405
core/algorithms/factor_in.hh . . . 406
core/algorithms/factor_out.cc . . . 406
core/algorithms/factor_out.hh . . . 406
core/algorithms/fierz.cc . . . 406
core/algorithms/fierz.hh . . . 407
core/algorithms/flatten_product.cc . . . 407
core/algorithms/flatten_product.hh . . . 407
core/algorithms/flatten_sum.cc . . . 407
core/algorithms/flatten_sum.hh . . . 408
core/algorithms/indexsort.cc . . . 408
core/algorithms/indexsort.hh . . . 408
core/algorithms/integrate_by_parts.cc . . . 408
core/algorithms/integrate_by_parts.hh . . . 409
core/algorithms/join_gamma.cc . . . 409
core/algorithms/join_gamma.hh . . . 409
core/algorithms/keep_terms.cc . . . 409
core/algorithms/keep_terms.hh . . . 410
core/algorithms/lr_tensor.cc . . . 410
core/algorithms/lr_tensor.hh . . . 410
core/algorithms/map_sympy.cc . . . 410
core/algorithms/map_sympy.hh . . . 411
core/algorithms/order.cc . . . 411
core/algorithms/order.hh . . . 411

core/algorithms/product_rule.cc . . . 412
core/algorithms/product_rule.hh . . . 412
core/algorithms/reduce_delta.cc . . . 412
core/algorithms/reduce_delta.hh . . . 412
core/algorithms/rename_dummies.cc . . . 413
core/algorithms/rename_dummies.hh . . . 413
core/algorithms/replace_match.cc . . . 413
core/algorithms/replace_match.hh . . . 413
core/algorithms/rewrite_indices.cc . . . 414
core/algorithms/rewrite_indices.hh . . . 414
core/algorithms/sort_product.cc . . . 414
core/algorithms/sort_product.hh . . . 414
core/algorithms/sort_spinors.cc . . . 415
core/algorithms/sort_spinors.hh . . . 415
core/algorithms/sort_sum.cc . . . 415
core/algorithms/sort_sum.hh . . . 415
core/algorithms/split.cc . . . 416
core/algorithms/split.hh . . . 416
core/algorithms/split_gamma.cc . . . 416
core/algorithms/split_gamma.hh . . . 416
core/algorithms/split_index.cc . . . 416
core/algorithms/split_index.hh . . . 417
core/algorithms/substitute.cc . . . 417
core/algorithms/substitute.hh . . . 417
core/algorithms/sym.cc . . . 417
core/algorithms/sym.hh . . . 418
core/algorithms/tab_basics.cc . . . 418
core/algorithms/tab_basics.hh . . . 418
core/algorithms/take_match.cc . . . 418
core/algorithms/take_match.hh . . . 419
core/algorithms/unwrap.cc . . . 419
core/algorithms/unwrap.hh . . . 419
core/algorithms/vary.cc . . . 420
core/algorithms/vary.hh . . . 420
core/algorithms/young_project.cc . . . 420
core/algorithms/young_project.hh . . . 420
core/algorithms/young_project_product.cc . . . 421
core/algorithms/young_project_product.hh . . . 421
core/algorithms/young_project_tensor.cc . . . 421
core/algorithms/young_project_tensor.hh . . . 421
core/properties/Accent.cc . . . 443
core/properties/Accent.hh . . . 443
core/properties/AntiCommuting.cc . . . 443
core/properties/AntiCommuting.hh . . . 443
core/properties/AntiSymmetric.cc . . . 444
core/properties/AntiSymmetric.hh . . . 444
core/properties/Commuting.cc . . . 444
core/properties/Commuting.hh . . . 444
core/properties/CommutingAsProduct.cc . . . 445
core/properties/CommutingAsProduct.hh . . . 445
core/properties/CommutingAsSum.cc . . . 445
core/properties/CommutingAsSum.hh . . . 445
core/properties/CommutingBehaviour.cc . . . 446
core/properties/CommutingBehaviour.hh . . . 446
core/properties/Coordinate.cc . . . 446
core/properties/Coordinate.hh . . . 446
core/properties/DAntiSymmetric.cc . . . 446
core/properties/DAntiSymmetric.hh . . . 447

core/properties/Depends.cc . . . 447
core/properties/Depends.hh . . . 447
core/properties/DependsBase.hh . . . 447
core/properties/DependsInherit.cc . . . 448
core/properties/DependsInherit.hh . . . 448
core/properties/Derivative.cc . . . 448
core/properties/Derivative.hh . . . 448
core/properties/Diagonal.cc . . . 449
core/properties/Diagonal.hh . . . 449
core/properties/DifferentialForm.cc . . . 449
core/properties/DifferentialForm.hh . . . 449
core/properties/DifferentialFormBase.hh . . . 450
core/properties/DiracBar.cc . . . 450
core/properties/DiracBar.hh . . . 450
core/properties/Distributable.cc . . . 451
core/properties/Distributable.hh . . . 451
core/properties/EpsilonTensor.cc . . . 451
core/properties/EpsilonTensor.hh . . . 451
core/properties/ExteriorDerivative.cc . . . 451
core/properties/ExteriorDerivative.hh . . . 452
core/properties/FilledTableau.cc . . . 452
core/properties/FilledTableau.hh . . . 452
core/properties/GammaMatrix.cc . . . 452
core/properties/GammaMatrix.hh . . . 453
core/properties/GammaTraceless.cc . . . 453
core/properties/GammaTraceless.hh . . . 453
core/properties/ImaginaryI.cc . . . 453
core/properties/ImaginaryI.hh . . . 454
core/properties/ImplicitIndex.cc . . . 454
core/properties/ImplicitIndex.hh . . . 454
core/properties/IndexInherit.hh . . . 454
core/properties/Indices.cc . . . 455
core/properties/Indices.hh . . . 455
core/properties/Integer.cc . . . 455
core/properties/Integer.hh . . . 455
core/properties/Integral.hh . . . 456
core/properties/InverseMetric.cc . . . 456
core/properties/InverseMetric.hh . . . 456
core/properties/KroneckerDelta.cc . . . 456
core/properties/KroneckerDelta.hh . . . 457
core/properties/LaTeXForm.cc . . . 457
core/properties/LaTeXForm.hh . . . 457
core/properties/Matrix.cc . . . 457
core/properties/Matrix.hh . . . 457
core/properties/Metric.cc . . . 458
core/properties/Metric.hh . . . 458
core/properties/NonCommuting.cc . . . 458
core/properties/NonCommuting.hh . . . 458
core/properties/NumericalFlat.cc . . . 459
core/properties/NumericalFlat.hh . . . 459
core/properties/PartialDerivative.cc . . . 459
core/properties/PartialDerivative.hh . . . 459
core/properties/RiemannTensor.cc . . . 460
core/properties/RiemannTensor.hh . . . 460
core/properties/SatisfiesBianchi.cc . . . 460
core/properties/SatisfiesBianchi.hh . . . 460
core/properties/SelfAntiCommuting.cc . . . 461
core/properties/SelfAntiCommuting.hh . . . 461

core/properties/SelfCommuting.cc . . . 461
core/properties/SelfCommuting.hh . . . 461
core/properties/SelfCommutingBehaviour.hh . . . 462
core/properties/SelfNonCommuting.cc . . . 462
core/properties/SelfNonCommuting.hh . . . 462
core/properties/SortOrder.cc . . . 462
core/properties/SortOrder.hh . . . 462
core/properties/Spinor.cc . . . 463
core/properties/Spinor.hh . . . 463
core/properties/Symbol.cc . . . 463
core/properties/Symbol.hh . . . 463
core/properties/Symmetric.cc . . . 464
core/properties/Symmetric.hh . . . 464
core/properties/Tableau.cc . . . 464
core/properties/Tableau.hh . . . 464
core/properties/TableauBase.cc . . . 465
core/properties/TableauBase.hh . . . 465
core/properties/TableauSymmetry.cc . . . 465
core/properties/TableauSymmetry.hh . . . 465
core/properties/Traceless.cc . . . 466
core/properties/Traceless.hh . . . 466
core/properties/Weight.cc . . . 466
core/properties/Weight.hh . . . 466
core/properties/WeightBase.hh . . . 466
core/properties/WeightInherit.cc . . . 467
core/properties/WeightInherit.hh . . . 467
core/properties/WeylTensor.cc . . . 467
core/properties/WeylTensor.hh . . . 467
frontend/common/exec-stream.cc . . . 485
frontend/common/exec-stream.h . . . 486
frontend/common/test_tex.cc . . . 486
frontend/common/TeXEngine.cc . . . 486
frontend/common/TeXEngine.hh . . . 487
frontend/gtkmm/Cadabra.cc . . . 487
frontend/gtkmm/Cadabra.hh . . . 488
frontend/gtkmm/CodeInput.cc . . . 488
frontend/gtkmm/CodeInput.hh . . . 488
frontend/gtkmm/ImageView.cc . . . 489
frontend/gtkmm/ImageView.hh . . . 489
frontend/gtkmm/main.cc . . . 489
frontend/gtkmm/NotebookCanvas.cc . . . 490
frontend/gtkmm/NotebookCanvas.hh . . . 490
frontend/gtkmm/NotebookWindow.cc . . . 491
frontend/gtkmm/NotebookWindow.hh . . . 491
frontend/gtkmm/Preferences.hh . . . 492
frontend/gtkmm/TeXView.cc . . . 492
frontend/gtkmm/TeXView.hh . . . 492
frontend/gtkmm/VisualCell.hh . . . 493
frontend/osx/Cadabra/NotebookController.h . . . 494
frontend/osx/Cadabra/NotebookController.mm . . . 494
frontend/osx/Cadabra/Cadabra/AppDelegate.h . . . 493
frontend/osx/Cadabra/Cadabra/AppDelegate.mm . . . 493
frontend/osx/Cadabra/Cadabra/main.m . . . 494
frontend/osx/Cadabra/Cadabra/NotebookCanvas.hh . . . 490
frontend/osx/Cadabra/Cadabra/NotebookCanvas.mm . . . 494
frontend/osx/Cadabra/Cadabra/NotebookWindow.hh . . . 492
frontend/osx/Cadabra/Cadabra/NotebookWindow.mm . . . 494

Chapter 7

Module Documentation

7.1 Core

7.1.1 Module Description

All computer algebra functionality in the form of C++ classes.

Modules

- Cleanup

  Functionality to clean up expressions so that they satisfy the requirements of a consistent expression tree.
- Expression comparison

  Functions to compare and match expressions stored in exptree objects.
- Python Core

  Python wrapper around the C++ core module, making the entire functionality of Cadabra accessible from Python.
- Scalar algebra

  Functionality to make Cadabra interface with the SymPy computer algebra system.
- Algorithms

  Containing all algorithms acting on symbolic expression trees.
- Properties

  Containing all properties that can be attached to symbolic expressions.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Classes

- class cadabra::Algorithm

  Base class for all algorithms, containing generic routines and in particular the logic for index classification.
- class cadabra::Ex_is_equivalent

  Basic comparison operator for tree iterators, so we can use them as keys in maps.
- class cadabra::index_iterator

  An iterator which iterates over indices even if they are at lower levels, i.e.
- class PermutationException

  Generic permutation group material.
- class preprocessor

  Preprocessing class which takes infix mathematical notation with all sorts of maths shortcuts and transforms it into a string which is properly formatted in prefix notation.
- class cadabra::property

  Base class for all properties, handling argument parsing and defining the interface.
- class cadabra::Properties

  Class holding a collection of properties attached to expressions.
- class cadabra::str_node

  Elementary building block for a mathematical expression.
- class cadabra::Ex

  Basic storage class for symbolic mathememetical expressions.
- class cadabra::nset_it_less

  Compare two nset iterators by comparing the strings to which they point.
- class cadabra::DependsBase

Functions

- void cadabra::do_list (const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f)

  Apply a function on every element of a list, or if the iterator 'it' does not point to a list, only on that single element.
- Ex::iterator cadabra::do_subtree (const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f)

  Apply a function on every node in the tree at and below the given node, depth-first.
- Ex::iterator cadabra::find_in_list (const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f)

  Returns an iterator to the first element for which 'f' does not return tr.end().
- Ex::iterator cadabra::find_in_subtree (const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f, bool including_head=true)

  Returns an iterator to the first element for which 'f' returns 'true', or 'tr.end()'.
- Ex cadabra::make_list (Ex el)

  Ensure that the tree is a list, even if it contains only a single element.
- void cadabra::multiply (rset_t::iterator &, multiplier_t)

  Helper functions for manipulation of multipliers.
- template<typename T >
  bool cadabra::is_in (const T &val, const std::initializer_list< T > &list)

  Bare output operator for Ex objects, mainly to provide a simple way to generate debugging output.

7.1.2 Function Documentation

7.1.2.1 void cadabra::do_list (const Ex & *tr*, Ex::iterator *it*, std::function< bool(Ex::iterator)> *f*)

Apply a function on every element of a list, or if the iterator 'it' does not point to a list, only on that single element.

Handles lists wrapped in an node as well. It is safe to remove the node pointed to by 'it' in 'f'. If your 'f' returns false, the loop is aborted immediately.

7.1.2.2 Ex::iterator cadabra::do_subtree (const Ex & *tr,* Ex::iterator *it,* std::function< Ex::iterator(Ex::iterator)> *f*)

Apply a function on every node in the tree at and below the given node, depth-first.

Return an iterator to the top node, which replaces 'it' (may be the same).

7.1.2.3 Ex::iterator cadabra::find_in_list (const Ex & *tr,* Ex::iterator *it,* std::function< Ex::iterator(Ex::iterator)> *f*)

Returns an iterator to the first element for which 'f' does not return tr.end().

7.1.2.4 Ex::iterator cadabra::find_in_subtree (const Ex & *tr,* Ex::iterator *it,* std::function< bool(Ex::iterator)> *f,* bool *including_head* = `true`)

Returns an iterator to the first element for which 'f' returns 'true', or 'tr.end()'.

7.1.2.5 template<typename T > bool cadabra::is_in (const T & *val,* const std::initializer_list< T > & *list*)

Bare output operator for Ex objects, mainly to provide a simple way to generate debugging output.

Does not do any fancy formatting; just prints a nested list representation. For more fancy output, look at DisplayTeX, DisplaySympy and DisplayTerminal.

7.1.2.6 Ex cadabra::make_list (Ex *el*)

Ensure that the tree is a list, even if it contains only a single element.

7.1.2.7 void cadabra::multiply (rset_t::iterator & , multiplier_t)

Helper functions for manipulation of multipliers.

7.2 Cleanup

7.2.1 Module Description

Functionality to clean up expressions so that they satisfy the requirements of a consistent expression tree.

Classes to handle printing of expressions in various formats.

There are several different requirements for printing, depending on whether output is to the console, the graphical notebook, or to a different computer algebra system.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Classes

- class cadabra::DisplayBase

  Base class for all display classes.

7.3 Expression comparison

7.3.1 Module Description

Functions to compare and match expressions stored in exptree objects.

These can involve wildcards and dummy indices, and matching may depend on the properties attached to symbols. This is an important bit of functionality used widely throughout the Cadabra core. All substitution and component evaluation logic depends on this module.

Classes

- class cadabra::Ex_comparator

  A generic tree comparison class which will take into account index contractions and will also keep track of a replacement list for all types of cadabra wildcards.

Functions

- int cadabra::subtree_compare (const Properties *, Ex::iterator one, Ex::iterator two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2, bool literal_wildcards=false)

  Basic building block subtree comparison function for tensors without dummy indices, which determines whether two tensor subtrees are equal up to the names of indices.

7.3.2 Function Documentation

7.3.2.1 int cadabra::subtree_compare (const Properties * , Ex::iterator *one*, Ex::iterator *two*, int *mod_prel* = `-2`, bool *checksets* = `true`, int *compare_multiplier* = `-2`, bool *literal_wildcards* = `false`)

Basic building block subtree comparison function for tensors without dummy indices, which determines whether two tensor subtrees are equal up to the names of indices.

This uses NO property information whatsoever; when indices are compared, they are simply compared based on their name, not on the index set they may belong to. In most cases, this is NOT what you want.

In MOST cases, the use of Ex_comparator below is recommended instead, as it can do much more complicated matching and will also keep track of the relation between symbols in the pattern and symbols in the object to be matched.

Examples:

```
A_m                       equals  A_n                       up to index names
\diff{A*B_g*\diff{C}_k}_m equals  \diff{A*B_h*\diff{C}_r}_s up to index names
```

return	meaning
0	structure equal, and all indices the same name
1	structure equal, index names of one < two
-1	structure equal, index names of one > two
2	structure different, one < two
-2	structure different, one > two

Parameters

mod_prel	see below
checksets	ignored FIXME: remove
literal_wildcards	whether to treat wildcard names as ordinary names.

The mod_prel variable determines whether parent relations are taken into account when comparing: FIXME: now that subtree_compare does not use properties anymore, a lot can be simplified.

```
  -3: require that parent relations match, unless indexpos=free.
  -2: require that parent relations match (even when indexpos = free)
  -1: do not require that parent relations match
 >=0: do not require parent relations to match up to and including this level
```

Similar logic holds for the compare_multiplier parameter.

7.4 Python Core

7.4.1 Module Description

Python wrapper around the C++ core module, making the entire functionality of Cadabra accessible from Python.

Classes

- class BaseProperty

  Helper class to ensure that all Python property objects derive from the same base class.
- class Property< T >

  Property is a templated wrapper around a C++ property object.

Functions

- void pull_in (std::shared_ptr< cadabra::Ex > ex)

  Replace any objects of the form '@(...)' in the expression tree by the python expression '...' if it exists.
- bool __eq__Ex_Ex (const cadabra::Ex &, const cadabra::Ex &)

  Comparison operator for Ex objects in Python.
- bool __eq__Ex_int (const cadabra::Ex &, int)

  Comparison operator for Ex objects in Python.
- std::string Ex_str_ (const cadabra::Ex &)

  Generate the Python str() and repr() representation of the Ex object.
- std::string Ex_latex_ (const cadabra::Ex &)

  The Python 'print' function always calls the 'str' member on objects to be printed.
- cadabra::Ex operator+ (const cadabra::Ex &ex1, const cadabra::Ex &ex2)

  Add two expressions, adding a top-level node if required.
- cadabra::Ex operator- (const cadabra::Ex &ex1, const cadabra::Ex &ex2)

  Subtract two expressions, adding a top-level node if required.
- cadabra::Kernel * create_scope ()

  Setup of kernels in current scope, callable from Python.
- void inject_defaults (cadabra::Kernel *)

  *Inject properties directly into the Kernel, even if the kernel is not yet on the Python stack (needed when we create a new local scope: in this case we create the kernel and pass it back to be turned into local **cdbkernel** by Python, but we want to populate the kernel with defaults before we hand it back).*
- cadabra::Kernel * get_kernel_from_scope ()

  Get a pointer to the currently visible kernel.
- void call_post_process (cadabra::Kernel &, cadabra::Ex &ex)

  Run the post-process Python function (if defined) on the given expression.

7.4.2 Function Documentation

7.4.2.1 bool __eq__Ex_Ex (const cadabra::Ex & , const cadabra::Ex &)

Comparison operator for Ex objects in Python.

Since comparison operators need a Properties object, we cannot have such operator== things in C++, but we can in Python since we can get the kernel in the current scope.

7.4.2.2 bool __eq__Ex_int (const cadabra::Ex & , int)

Comparison operator for Ex objects in Python.

See __eq__Ex_Ex for more.

7.4.2.3 void call_post_process (cadabra::Kernel & , cadabra::Ex & *ex*)

Run the post-process Python function (if defined) on the given expression.

7.4.2.4 cadabra::Kernel* create_scope ( )

Setup of kernels in current scope, callable from Python.

When the decision was made to graft Cadabra onto Python, a choice had to be made about how Python variable scope would influence the visibility of Cadabra properties. It clearly makes sense to be able to declare properties which only hold inside a particular function. However Cadabra expressions and properties do not directly correspond to Python objects. Rather, declaring a property is more like a function call into the Cadabra module, which leaves its imprint on the state of the C++ part but does not change anything on the Python side, as you typically do not assign the created property to a Python symbol. Therefore, properties do not naturally inherit Python's scoping rules.{This is different from e.g.∼SymPy, in which mathematical objects are always in one-to-one correspondence with a Python object.} A more fundamental problem is that properties can be attached to patterns, and those patterns can involve more than just the symbols which one passes into a function.

In order to not burden the user, properties are therefore by default global variables, stored in a single global Cadabra object |__cdbkernel__| which is initialised at import of the Cadabra module. If you add new properties inside a function scope, these will go into this already existing {global} property list by default. If you want to create a local scope for your computations, create a new |__cdbkernel__| as in {verbatim} def fun(): **cdbkernel** = cadabra.↩ create_scope(); [your code here] {verbatim} Now computations will not see the global properties at all. If you want to import the global properties, use instead {verbatim} def fun(): **cdbkernel** = cadabra.create_scope_from_global() [your code here] {verbatim} It is crucial that the |__cdbkernel__| symbol is referenced from within Python and visible to the bytecompiler, because it is not possible to create new variables on the local stack at runtime. Internally, the second version above fetches, at runtime, the |__cdbkernel__| from the globals stack, copies all properties in there into a new kernel, and returns the latter.

Both versions above do populate the newly created kernel with Cadabra's default properties. If you want a completely clean slate (for e.g.∼testing purposes, or because you really do not want default rules for sums and products), use {verbatim} def fun(): **cdbkernel** = cadabra.create_empty_scope() [your code here] {verbatim} Note that in all these cases, changes to properties remain local and do not leak into the global property list.

All Cadabra algorithms, when called from Python, will first look for a kernel on the locals stack (i.e.∼what |locals()| produces). If there is no kernel available locally, they will then revert to using the global kernel.

7.4.2.5 std::string Ex_latex_ (const cadabra::Ex &)

The Python 'print' function always calls the 'str' member on objects to be printed.

This one is required to produce output which looks readable but is also still valid input. In order to produce proper LaTeX output, this is therefore not the right function to use, because Cadabra only reads a restricted subset of LaTeX (for instance, we output spacing commands like '\,' but do not accept it on input). So we have a separate *latex*() member on each object, which internally uses DisplayTeX to do the actual printing.

7.4.2.6 std::string Ex_str_ (const cadabra::Ex &)

Generate the Python str() and repr() representation of the Ex object.

7.4.2.7 cadabra::Kernel* get_kernel_from_scope ( )

Get a pointer to the currently visible kernel.

7.4.2.8 void inject_defaults (cadabra::Kernel *)

Inject properties directly into the Kernel, even if the kernel is not yet on the Python stack (needed when we create a new local scope: in this case we create the kernel and pass it back to be turned into local **cdbkernel** by Python, but we want to populate the kernel with defaults before we hand it back).

7.4.2.9 cadabra::Ex operator+ (const cadabra::Ex & *ex1,* const cadabra::Ex & *ex2*)

Add two expressions, adding a top-level node if required.

7.4.2.10 cadabra::Ex operator- (const cadabra::Ex & *ex1,* const cadabra::Ex & *ex2*)

Subtract two expressions, adding a top-level node if required.

7.4.2.11 void pull_in (std::shared_ptr< cadabra::Ex > *ex*)

Replace any objects of the form '@(...)' in the expression tree by the python expression '...' if it exists.

Rename dummies to avoid clashes.

7.5 Scalar algebra

7.5.1 Module Description

Functionality to make Cadabra interface with the SymPy computer algebra system.

Functions

- boost::python::object Ex_to_Sympy (const cadabra::Ex &)

  Convert a Cadabra 'Ex' to a Sympy expression.
- cadabra::Ex * sympy::map_sympy (const cadabra::Kernel &, cadabra::Ex &, const std::vector< std::string > &wrap, const std::string &args, const std::string &method)

  Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place.
- cadabra::Ex::iterator sympy::apply (const cadabra::Kernel &, cadabra::Ex &, cadabra::Ex::iterator &, const std::vector< std::string > &wrap, const std::string &args, const std::string &method)

  Functionality to act with Sympy functions on (parts of) Cadabra Ex expressions and read the result back into the same Ex.
- cadabra::Ex sympy::invert_matrix (const cadabra::Kernel &, cadabra::Ex &ex, cadabra::Ex &rules)

  Use Sympy to invert a matrix, given a set of rules determining its sparse components.

7.5.2 Function Documentation

7.5.2.1 cadabra::Ex::iterator sympy::apply (const cadabra::Kernel & , cadabra::Ex & , cadabra::Ex::iterator & , const std::vector< std::string > & *wrap,* const std::string & *args,* const std::string & *method*)

Functionality to act with Sympy functions on (parts of) Cadabra Ex expressions and read the result back into the same Ex.

This duplicates some of the logic in PythonCdb.hh, in particular make_Ex_from_string, but it is best to keep these two completely separate.

7.5.2.2 boost::python::object Ex_to_Sympy (const cadabra::Ex &)

Convert a Cadabra 'Ex' to a Sympy expression.

This first converts the Cadabra expression to a string, and then reads that back in by calling sympy.parsing.sympy↩_parser.parse_expr. Is mapped to a '*sympy*()' function on each Ex object. When you feed an Ex object to a Sympy function, the Ex gets converted to a Sympy object in 'sympy.sympify' because the latter attempts to call **sympy** on every object that you feed it.

7.5.2.3 Ex sympy::invert_matrix (const cadabra::Kernel & *kernel,* cadabra::Ex & *ex,* cadabra::Ex & *rules*)

Use Sympy to invert a matrix, given a set of rules determining its sparse components.

Will return a set of Cadabra rules for the inverse matrix.

7.5.2.4 cadabra::Ex* sympy::map_sympy (const cadabra::Kernel & , cadabra::Ex & , const std::vector< std::string > & *wrap,* const std::string & *args,* const std::string & *method*)

Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place.

This is a higher-level function than 'apply' below.

7.6 Algorithms

7.6.1 Module Description

Containing all algorithms acting on symbolic expression trees.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Classes

- class cadabra::canonicalise

  Canonicalise the index structure of a tensorial expression.
- class cadabra::collect_components

  Collect terms inside a sum, merging their substitution rules.
- class cadabra::collect_factors

  Collect symbolically equal factors in a product.
- class cadabra::collect_terms

  Collect symbolically equal terms in a sum.
- class cadabra::complete

  Complete a set of coordinate rules so that they also cover related tensors.
- class cadabra::decompose_product

  Decompose a product of tensors by applying Young projectors.
- class cadabra::distribute

  Distribute factors over a sum, that is, $(A+B)C \to AC+BC$.
- class cadabra::einsteinify
- class cadabra::factor_in
- class cadabra::factor_out
- class cadabra::integrate_by_parts

  Integrate by parts away from the indicated derivative object.
- class cadabra::map_sympy

  Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place.
- class split

  Split an expression into terms or factors.
- class cadabra::substitute

  Generic substitution algorithm.
- class cadabra::vary

  Generic variational derivative algorithm.

7.7 Properties

7.7.1 Module Description

Containing all properties that can be attached to symbolic expressions.

Classes

- class cadabra::Accent

  Turns a symbol into an accent.
- class cadabra::DependsInherit
- class cadabra::EpsilonTensor
- class cadabra::Integer

  Property indicating that a symbolic object always takes integer values.

7.8 Client-Server

7.8.1 Module Description

Base classes (independent of any GUI toolkit) to write Cadabra clients and servers.

All clients use two threads. One is the main thread and runs most of the logic in the DocumentThread object. The other one is a thread which communicates with the server process; code for this thread is in the ComputeThread object. One typically subclasses the former into a full-fledged object driving the graphical user interface.

All functionality that deals with GUI updates is isolated in GUIBase. A client should derive from this abstract base class and implement the methods there. Clients should also derive from DocumentThread, but there are no abstract virtual members to implemented from this base class.

All actual updates to the document stored in the DocumentThread::dtree member are made on the GUI thread as well. The compute thread merely puts requests to change the document onto a stack, and then requests that the GUI updates the document.

In order to enable undo/redo, all updates to DocumentThread::dtree are made by putting objects derived from ActionBase onto the action stack. See DocumentThread::queue_action for details. These ActionBase objects (among which ActionAddCell, ActionRemoveCell, ActionInsertText, etc) get executed on the GUI thread, and contain the actual logic to perform updates to the document and related GUI elements.

Classes

- class cadabra::ActionBase

  All actions derive from the ActionBase object, which defines the interface they need to implement.
- class cadabra::ActionAddCell

  Add a cell to the notebook.
- class cadabra::ActionPositionCursor

  Position the cursor relative to the indicated cell.
- class cadabra::ActionSetRunStatus

  Update the running status of the indicated cell.
- class cadabra::ActionRemoveCell

  Remove a cell and all its child cells from the document.
- class cadabra::ActionSplitCell

  Split a cell into two separate cells, at the point of the cursor.
- class cadabra::ActionInsertText

  Add a text string (can be just a single character) at the point of the cursor.
- class cadabra::ActionEraseText

  Remove a text string starting at the indicated position, and with the indicated length, from the indicated cell.
- class cadabra::ComputeThread

  Base class which talks to the server and sends Action objects back to the DocumentThread.
- class cadabra::DataCell

  DataCells are the basic building blocks for a document.
- class cadabra::DocumentThread

  A base class with all the logic to manipulate a Cadabra notebook document.
- class cadabra::GUIBase

  Abstract base class with methods that need to be implemented by any GUI.
- class Server

  Object representing a Cadabra server, capable of receiving messages on a websocket, running Python code, and sending output back to the client.

7.9 Front-end

7.9.1 Module Description

All code which implements graphical front-end notebook interfaces.

Modules

- Gtk+

  Notebook user interface implemented using gtkmm.
- OS-X

  Notebook interface implemented using the native OS-X Cocoa toolkit.

Classes

- class cadabra::TeXEngine

  TeXEngine is used to convert LaTeX strings into PNG images.

7.10 Gtk+

7.10.1 Module Description

Notebook user interface implemented using gtkmm.

Classes

- class cadabra::CodeInput

  A text cell editor widget with support for editing Python and LaTeX.
- class cadabra::ImageView

  An image viewing widget.
- class cadabra::NotebookWindow

  Each notebook has one main window which controls it.
- class cadabra::VisualCell

  Structure holding a pointer to one of the possible GUI widgets that can appear in a document.

7.11 OS-X

7.11.1 Module Description

Notebook interface implemented using the native OS-X Cocoa toolkit.

Classes

- class cadabra_osx::NotebookWindow

  Objective-C++ class implementing DocumentThread and providing an OS-X notebook interface.

Chapter 8

Namespace Documentation

8.1 cadabra Namespace Reference

Class to display expressions in a format that Sympy can parse.

Namespaces

- symbols

Classes

- class Accent

  Turns a symbol into an accent.
- class ActionAddCell

  Add a cell to the notebook.
- class ActionBase

  All actions derive from the ActionBase object, which defines the interface they need to implement.
- class ActionEraseText

  Remove a text string starting at the indicated position, and with the indicated length, from the indicated cell.
- class ActionInsertText

  Add a text string (can be just a single character) at the point of the cursor.
- class ActionPositionCursor

  Position the cursor relative to the indicated cell.
- class ActionRemoveCell

  Remove a cell and all its child cells from the document.
- class ActionSetRunStatus

  Update the running status of the indicated cell.
- class ActionSplitCell

  Split a cell into two separate cells, at the point of the cursor.
- class Algorithm

  Base class for all algorithms, containing generic routines and in particular the logic for index classification.
- class AntiCommuting
- class AntiSymmetric
- class canonicalise

Canonicalise the index structure of a tensorial expression.

- class CodeInput

  A text cell editor widget with support for editing Python and LaTeX.
- class collect_components

  Collect terms inside a sum, merging their substitution rules.
- class collect_factors

  Collect symbolically equal factors in a product.
- class collect_terms

  Collect symbolically equal terms in a sum.
- class combine
- class Commuting
- class CommutingAsProduct
- class CommutingAsSum
- class CommutingBehaviour
- class complete

  Complete a set of coordinate rules so that they also cover related tensors.
- class ComputeThread

  Base class which talks to the server and sends Action objects back to the DocumentThread.
- class Coordinate
- class DAntiSymmetric
- class DataCell

  DataCells are the basic building blocks for a document.
- class decompose_product

  Decompose a product of tensors by applying Young projectors.
- class Depends
- class DependsBase
- class DependsInherit
- class Derivative
- class Diagonal
- class DifferentialForm
- class DifferentialFormBase
- class DiracBar
- class DisplayBase

  Base class for all display classes.
- class DisplaySympy
- class DisplayTerminal
- class DisplayTeX
- class Distributable
- class distribute

  Distribute factors over a sum, that is, $(A+B)C \to AC + BC$.
- class DocumentThread

  A base class with all the logic to manipulate a Cadabra notebook document.
- class drop_keep_weight
- class drop_weight
- class einsteinify
- class eliminate_converter
- class eliminate_kronecker
- class eliminate_metric
- class epsilon_to_delta
- class EpsilonTensor
- class evaluate
- class Ex

  Basic storage class for symbolic mathemematical expressions.

- class Ex_comparator

  A generic tree comparison class which will take into account index contractions and will also keep track of a replacement list for all types of cadabra wildcards.
- class Ex_is_equivalent

  Basic comparison operator for tree iterators, so we can use them as keys in maps.
- class Ex_is_less
- class exchange
- class expand
- class expand_delta
- class expand_diracbar
- class expand_power
- class ExteriorDerivative
- class factor_in
- class factor_out
- class fierz
- class FilledTableau
- class flatten_product
- class flatten_sum
- class GammaMatrix
- class GammaTraceless
- class GUIBase

  Abstract base class with methods that need to be implemented by any GUI.
- class ImageView

  An image viewing widget.
- class ImaginaryI
- class ImplicitIndex
- class index_iterator

  An iterator which iterates over indices even if they are at lower levels, i.e.
- class IndexInherit
- class indexsort
- class Indices
- class Inherit

  If a property X derives from Inherit<Y>, and get<Y> is called on an object which has an X property (but no Y property), then the get<Y> will look at the non-index child of the object to see if that has a Y property.
- class Integer

  Property indicating that a symbolic object always takes integer values.
- class integrate_by_parts

  Integrate by parts away from the indicated derivative object.
- class InverseMetric
- class join_gamma
- class keep_terms
- class keep_weight
- class Kernel
- class keyval_t

  Arguments to properties get parsed into a keyval_t structure.
- class KroneckerDelta
- class labelled_property
- class LaTeXForm
- class list_property

  Something cannot be both a list property and a normal property at the same time, so we can safely inherit without virtual.
- class lr_tensor
- class map_sympy

Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place.

- class Matrix
- class Metric
- class NonCommuting
- class NotebookCanvas
- class NotebookWindow

  Each notebook has one main window which controls it.
- class nset_it_less

  Compare two nset iterators by comparing the strings to which they point.
- class NumericalFlat

  Property indicating that an operator is numerically flat, so that numerical factors in the argument can be taken outside.
- class order
- class Parser
- class PartialDerivative
- class pattern
- class product_rule
- class Properties

  Class holding a collection of properties attached to expressions.
- class property

  Base class for all properties, handling argument parsing and defining the interface.
- class PropertyInherit

  PropertyInherit is like Inherit<T> for all properties.
- class reduce_delta
- class rename_dummies
- class replace_match
- class rewrite_indices
- class RiemannTensor
- class SatisfiesBianchi
- class SelfAntiCommuting
- class SelfCommuting
- class SelfCommutingBehaviour
- class SelfNonCommuting
- class sort_product
- class sort_spinors
- class sort_sum
- class SortOrder
- class Spinor
- class split_gamma
- class split_index
- class str_node

  Elementary building block for a mathematical expression.
- class substitute

  Generic substitution algorithm.
- class sym
- class Symbol
- class Symmetric
- class tab_basics
- class Tableau
- class TableauBase
- class TableauSymmetry
- class take_match
- class TerminalStream
- class TeXEngine

TeXEngine is used to convert LaTeX strings into PNG images.

- class TeXView

  TeXView is a widget which knows how to turn a string into a LaTeX-rendered image and display that.
- class Traceless
- class tree_equal_obj
- class tree_exact_equal_mod_prel_obj
- class tree_exact_equal_obj
- class tree_exact_less_for_indexmap_obj

  Compare for indexmap_t.
- class tree_exact_less_mod_prel_obj
- class tree_exact_less_no_wildcards_mod_prel_obj
- class tree_exact_less_no_wildcards_obj

  Compare two trees exactly, treat wildcard names as ordinary names.
- class tree_exact_less_obj

  Compare two trees exactly, i.e. including exact index names.
- class tree_less_modprel_obj
- class tree_less_obj

  Compare two trees by pattern logic, i.e. modulo index names.
- class unwrap
- class vary

  Generic variational derivative algorithm.
- class VisualCell

  Structure holding a pointer to one of the possible GUI widgets that can appear in a document.
- class Weight
- class WeightBase
- class WeightInherit
- class WeylTensor
- class young_project
- class young_project_product
- class young_project_tensor

Typedefs

- typedef void(∗ dispatcher_t) (const Kernel &k, Ex &, Ex::iterator &it)
- typedef mpq_class multiplier_t
- typedef std::set< std::string > nset_t
- typedef std::set< multiplier_t > rset_t
- typedef uintptr_t hashval_t
- typedef tree< DataCell > DTree

Functions

- std::string escape_quotes (const std::string &)
- std::string cdb2python (const std::string &)
- std::string convert_line (const std::string &, std::string &lhs, std::string &rhs, std::string &indent)
- void cleanup_dispatch (const Kernel &k, Ex &, Ex::iterator &it)

  Central cleanup dispatch routine, which calls the other cleanup functions defined later.
- void check_index_consistency (const Kernel &k, Ex &tr, Ex::iterator it)

  Generic tool to check for index inconsistencies independent of running any algorithm.
- bool cleanup_productlike (const Kernel &k, Ex &, Ex::iterator &it)

  Individual node cleanup routines.

- bool cleanup_sumlike (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool push_down_multiplier (const Kernel &k, Ex &tr, Ex::iterator it)

  Given a node with a non-unit multiplier, push this multiplier down the tree if the node is not allowed to have a non-unit multiplier.
- bool cleanup_components (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cleanup_partialderivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cleanup_derivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cleanup_numericalflat (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cleanup_diagonal (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cleanup_kronecker (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cleanup_exterior_derivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- void cleanup_dispatch_deep (const Kernel &k, Ex &, dispatcher_t disp=&cleanup_dispatch)

  More general cleanup of an entire tree.
- void cleanup_dispatch_deep (const Kernel &k, Ex &tr, Ex::iterator &top, dispatcher_t dispatch)
- int subtree_compare (const Properties ∗, Ex::iterator one, Ex::iterator two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2, bool literal_wildcards=false)

  Basic building block subtree comparison function for tensors without dummy indices, which determines whether two tensor subtrees are equal up to the names of indices.
- bool tree_less (const Properties ∗, const Ex &one, const Ex &two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2)

  Various comparison functions, some exact, some with pattern logic.
- bool tree_equal (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier)
- bool tree_exact_less (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool tree_exact_equal (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool subtree_less (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier)
- bool subtree_equal (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier)
- bool subtree_exact_less (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool subtree_exact_equal (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- const char ∗ unichar (kunichar c)
- bool operator< (const exchange::tensor_type_t &one, const exchange::tensor_type_t &two)
- void do_list (const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f)

  Apply a function on every element of a list, or if the iterator 'it' does not point to a list, only on that single element.
- int list_size (const Ex &tr, Ex::iterator it)

  For lists as defined above for 'do_list', return their size (in case you really need to know the size before iterating over the elements).
- Ex::iterator do_subtree (const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f)

  Apply a function on every node in the tree at and below the given node, depth-first.
- Ex::iterator find_in_subtree (const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f, bool including↩ _head=true)

  Returns an iterator to the first element for which 'f' returns 'true', or 'tr.end()'.
- Ex::iterator find_in_list (const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f)

  Returns an iterator to the first element for which 'f' does not return tr.end().
- Ex make_list (Ex el)

  Ensure that the tree is a list, even if it contains only a single element.
- void pre_clean_dispatch (const Kernel &kernel, Ex &ex, Ex::iterator &it)
- void pre_clean_dispatch_deep (const Kernel &k, Ex &tr)
- void cleanup_updown (const Kernel &k, Ex &, Ex::iterator &it)

Cleanup for individual node types.

- void cleanup_rational (const Kernel &k, Ex &tr, Ex::iterator &st)
- void cleanup_frac (const Kernel &k, Ex &tr, Ex::iterator &st)
- void cleanup_sub (const Kernel &k, Ex &tr, Ex::iterator &it)
- void cleanup_indexbracket (const Kernel &k, Ex &, Ex::iterator &it)

Convert parser output which indicates an indexbracket to an actual indexbracket node.

- long to_long (multiplier_t mul)
- std::string to_string (long num)
- void multiply (rset_t::iterator &, multiplier_t)

Helper functions for manipulation of multipliers.

- void add (rset_t::iterator &num, multiplier_t fac)
- void zero (rset_t::iterator &num)
- void one (rset_t::iterator &num)
- void flip_sign (rset_t::iterator &num)
- void half (rset_t::iterator &num)
- template<typename T >
  bool is_in (const T &val, const std::initializer_list< T > &list)

Bare output operator for Ex objects, mainly to provide a simple way to generate debugging output.

- std::string JSON_serialise (const DTree &)

Serialise a document into .cj format, which is a JSON version of the document tree.

- void JSON_recurse (const DTree &, DTree::iterator, Json::Value &)
- void JSON_deserialise (const std::string &, DTree &)

Load a document from .cj format, i.e. the inverse of the above.

- void JSON_in_recurse (DTree &doc, DTree::iterator loc, const Json::Value &cells)
- std::string export_as_HTML (const DTree &doc, bool for_embedding=false, std::string title="")

Export a document to a single self-contained HTML file containing inline CSS.

- void HTML_recurse (const DTree &doc, DTree::iterator it, std::ostringstream &str, const std::string &preamble_string, bool for_embedding=false, std::string title="")
- std::string latex_to_html (const std::string &)

Convert various LaTeX constructions to HTML-with-Mathjax, e.g.

- std::string export_as_LaTeX (const DTree &doc)

Export a document to a single self-contained LaTeX file.

- void LaTeX_recurse (const DTree &doc, DTree::iterator it, std::ostringstream &str, const std::string &preamble_string)
- std::string export_as_python (const DTree &doc)

Export a document to a python-like file (converting text cells to comments and python cells to python code, dropping output cells).

- void python_recurse (const DTree &doc, DTree::iterator it, std::ostringstream &str)

Variables

- nset_t name_set
- rset_t rat_set

8.1.1 Detailed Description

Class to display expressions in a format that Sympy can parse.

Evaluate a tensorial expression to components, performing all dummy index sums.

Handle the entire preclean stage, which turns a parsed expression into an expression which satisfies the Cadabra conventions.

Parser module, which takes the string output of the preprocessor.hh module and turns it into an Ex expression tree.

Cadabra kernel that keeps all state information that needs to be passed around to algorithms and properties.

Functions to handle the exchange properties of two or more symbols in a product.

Class to handle display of expressions using LaTeX notation.

Class to display expressions to the terminal.

Will throw an exception if a Cadabra Ex object cannot be understood by Sympy. Can also convert expressions back to Cadabra notation. This class contains the printing logic that you see in action when you call the '**sympy**' method on an Ex object.

This is used in the PythonCdb bridge to provide 'str' type python printing for Ex objects. The cadabra2 command line application uses this class for all its output. This class does not necessarily produce output which is readable by sympy; for that, use the DisplaySympy class (as an example, partial derivatives will display using a LaTeX notation using the present class, but be printed as 'sympy.diff' by the DisplaySympy class).

This is a very non-local bit of logic, in the sense that it will try to look around the property assignments in order to figure out the best way to print any given object. See the DisplayTeX::dispatch method to see how this works in more detail.

This module is only concerned with the exchange properties of tensors as a whole, not with index permutation symmetries (which are handled in the canonicalise class of algebra.cc).

At the moment only stores property information, but could also store global settings and the like at some later stage.

The output of preprocessor.hh is assumed to be valid and consistent, so the code here is rather simple.

This does not involve the property system.

- All numerical multipliers in a product on the product node, no multiplier on a sum node.
- Any '\frac' nodes with a purely numerical denominator should be rewritten as a rational multiplier for the numerator node.
- Any \sub nodes get converted to \sum nodes.

  See tests/components.cdb for basic samples.

  Components nodes have the following structure:

  _{m} // index names/types _{n} // last child // list of index values r t [value] // value of tensor for above values of indices ... ...

  ex:= A_{m n} (A_{m n} + B_{m n}); crds:= m -> { t, r, , }; vals:= { A_{t t} -> r, A_{r r} -> r**2, A_{t } -> t, B_{t t} -> 1 }; evaluate(ex, crds, vals); tst:= r * (r + 1) + r**2 * r**2 + t*t - @(ex);

  The algorithm performs the following steps. First, all free and dummy indices are classified using members of Algorithm (maybe later?). For all indices, the 'crds' argument to the algorithm is scanned for a

```
m -> { a,b,c }
```

type pattern.

Do a dept-first scan until the first sum or product node.

In a sum node, all terms must have the same index structure, so we can now make a map from the subtree pattern to all possible index values. For each such index value set, lookup if there is a replacement pattern, if not replace by zero. This map can be reused later for equal sub-expressions (here it is important that a normal tensor expression can be used as a pattern immediately, though I would search both on explicit iterator (for things that we have just seen) and on pattern (for things that occur again later in the tree)).

In a product node, find all dummies and free indices. Create a set of all index value sets, e.g.

{m, n, p} -> { {t, t, t}, {r,t,t}, {t,r,t}, ... }

For each such index value set, lookup replacement rules.

KEY: quick way to lookup, for a given set of index values on a (found) pattern/pattern pointed to by iterator, whether and what is the value of the pattern.

8.1.2 Typedef Documentation

8.1.2.1 typedef void(* cadabra::dispatcher_t) (const Kernel &k, Ex &, Ex::iterator &it)

8.1.2.2 typedef tree<DataCell> cadabra::DTree

8.1.2.3 typedef uintptr_t cadabra::hashval_t

8.1.2.4 typedef mpq_class cadabra::multiplier_t

8.1.2.5 typedef std::set<std::string> cadabra::nset_t

8.1.2.6 typedef std::set<multiplier_t> cadabra::rset_t

8.1.3 Function Documentation

8.1.3.1 void cadabra::add (rset_t::iterator & *num,* multiplier_t *fac*)

8.1.3.2 std::string cadabra::cdb2python (const std::string & *blk*)

8.1.3.3 void cadabra::check_index_consistency (const Kernel & *k,* Ex & *tr,* Ex::iterator *it*)

Generic tool to check for index inconsistencies independent of running any algorithm.

8.1.3.4 bool cadabra::cleanup_components (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.5 bool cadabra::cleanup_derivative (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.6 bool cadabra::cleanup_diagonal (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.7 void cadabra::cleanup_dispatch (const Kernel & *k,* Ex & , Ex::iterator & *it*)

Central cleanup dispatch routine, which calls the other cleanup functions defined later.

These cleanup routines do NOT use normal cadabra algorithms; they are completely independent of them to prevent circular dependence or infinite recursion.

These algorithms clean up the tree at the current node and the first layer of child nodes, but do NOT descend deeper down the tree, UNLESS that would leave the tree in an inconsistent state. An example is acting at the top node of {4}{{a}{b}}, which would push the 4 to the multiplier of the sum, but that is not allowed, so it needs to go further down. Sibling nodes of 'it' remain untouched as well.

8.1.3.8 void cadabra::cleanup_dispatch_deep (const Kernel & *k,* Ex & , dispatcher_t *disp* = &cleanup_dispatch)

More general cleanup of an entire tree.

Walks depth-first along the entire tree and call cleanup_dispatch at every node.

8.1.3.9 void cadabra::cleanup_dispatch_deep (const Kernel & *k,* Ex & *tr,* Ex::iterator & *top,* dispatcher_t *dispatch*)

8.1.3.10 bool cadabra::cleanup_exterior_derivative (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.11 void cadabra::cleanup_frac (const Kernel & *k,* Ex & *tr,* Ex::iterator & *st*)

8.1.3.12 void cadabra::cleanup_indexbracket (const Kernel & *k,* Ex & , Ex::iterator & *it*)

Convert parser output which indicates an indexbracket to an actual indexbracket node.

8.1.3.13 bool cadabra::cleanup_kronecker (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.14 bool cadabra::cleanup_numericalflat (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.15 bool cadabra::cleanup_partialderivative (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.16 bool cadabra::cleanup_productlike (const Kernel & *k,* Ex & , Ex::iterator & *it*)

Individual node cleanup routines.

Do not call these yourself.

Once more, these algorithms clean up the tree at the current node and the first layer of child nodes, but do NOT descend deeper down the tree, except when that is necessary to ensure that the tree remains consistent. As with any algorithms, the iterator pointing to the starting node may be changed, but these functions are not allowed to modify anything except the node and nodes below (in particular, they will leave sibling nodes untouched).

8.1.3.17 void cadabra::cleanup_rational (const Kernel & *k,* Ex & *tr,* Ex::iterator & *st*)

8.1.3.18 void cadabra::cleanup_sub (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.19 bool cadabra::cleanup_sumlike (const Kernel & *k,* Ex & *tr,* Ex::iterator & *it*)

8.1.3.20 void cadabra::cleanup_updown (const Kernel & *k,* Ex & , Ex::iterator & *it*)

Cleanup for individual node types.

These are not needed during any later stages of the manipulation of an expression (and are hence defined here, not in 'Cleanup.hh').

8.1.3.21 std::string cadabra::convert_line (const std::string & *line,* std::string & *lhs,* std::string & *rhs,* std::string & *indent*)

8.1.3.22 std::string cadabra::escape_quotes (const std::string & *line*)

8.1.3.23 std::string cadabra::export_as_HTML (const DTree & *doc,* bool *for_embedding* = `false`, std::string *title* = " ")

Export a document to a single self-contained HTML file containing inline CSS.

8.1.3.24 std::string cadabra::export_as_LaTeX (const DTree & *doc*)

Export a document to a single self-contained LaTeX file.

8.1.3.25 std::string cadabra::export_as_python (const DTree & *doc*)

Export a document to a python-like file (converting text cells to comments and python cells to python code, dropping output cells).

8.1.3.26 void cadabra::flip_sign (rset_t::iterator & *num*)

8.1.3.27 void cadabra::half (rset_t::iterator & *num*)

8.1.3.28 void cadabra::HTML_recurse (const DTree & *doc,* DTree::iterator *it,* std::ostringstream & *str,* const std::string & *preamble_string,* bool *for_embedding* = `false`, std::string *title* = " ")

8.1.3.29 void cadabra::JSON_deserialise (const std::string & *cj,* DTree & *doc*)

Load a document from .cj format, i.e. the inverse of the above.

8.1.3.30 void cadabra::JSON_in_recurse (DTree & *doc,* DTree::iterator *loc,* const Json::Value & *cells*)

8.1.3.31 void cadabra::JSON_recurse (const DTree & *doc,* DTree::iterator *it,* Json::Value & *json*)

8.1.3.32 std::string cadabra::JSON_serialise (const DTree & *doc*)

Serialise a document into .cj format, which is a JSON version of the document tree.

8.1.3.33 void cadabra::LaTeX_recurse (const DTree & *doc,* DTree::iterator *it,* std::ostringstream & *str,* const std::string & *preamble_string*)

8.1.3.34 std::string cadabra::latex_to_html (const std::string & *str*)

Convert various LaTeX constructions to HTML-with-Mathjax, e.g.

8.1.3.35 int cadabra::list_size (const Ex & *tr,* Ex::iterator *it*)

For lists as defined above for 'do_list', return their size (in case you really need to know the size before iterating over the elements).

8.1.3.36 void cadabra::one (rset_t::iterator & *num*)

8.1.3.37 bool cadabra::operator< (const exchange::tensor_type_t & *one,* const exchange::tensor_type_t & *two*)

8.1.3.38 void cadabra::pre_clean_dispatch (const Kernel & *kernel,* Ex & *ex,* Ex::iterator & *it*)

8.1.3.39 void cadabra::pre_clean_dispatch_deep (const Kernel & *k,* Ex & *tr*)

8.1.3.40 bool cadabra::push_down_multiplier (const Kernel & *k,* Ex & *tr,* Ex::iterator *it*)

Given a node with a non-unit multiplier, push this multiplier down the tree if the node is not allowed to have a non-unit multiplier.

This is a recursive procedure as the node onto which the multiplier gets pushed may itself not allow for a non-unit multiplier. Note that some nodes disallow non-unit multipliers on their children, but that should be handled individually (see cleanup of product nodes for an example).

8.1.3.41 void cadabra::python_recurse (const DTree & *doc,* DTree::iterator *it,* std::ostringstream & *str*)

8.1.3.42 bool cadabra::subtree_equal (const Properties * *properties,* Ex::iterator *one,* Ex::iterator *two,* int *mod_prel,* bool *checksets,* int *compare_multiplier*)

8.1.3.43 bool cadabra::subtree_exact_equal (const Properties * *properties,* Ex::iterator *one,* Ex::iterator *two,* int *mod_prel,* bool *checksets,* int *compare_multiplier,* bool *literal_wildcards*)

8.1.3.44 bool cadabra::subtree_exact_less (const Properties * *properties,* Ex::iterator *one,* Ex::iterator *two,* int *mod_prel,* bool *checksets,* int *compare_multiplier,* bool *literal_wildcards*)

8.1.3.45 bool cadabra::subtree_less (const Properties * *properties,* Ex::iterator *one,* Ex::iterator *two,* int *mod_prel,* bool *checksets,* int *compare_multiplier*)

8.1.3.46 long cadabra::to_long (multiplier_t *mul*)

8.1.3.47 std::string cadabra::to_string (long *num*)

8.1.3.48 bool cadabra::tree_equal (const Properties * *properties,* const Ex & *one,* const Ex & *two,* int *mod_prel,* bool *checksets,* int *compare_multiplier*)

8.1.3.49 bool cadabra::tree_exact_equal (const Properties * *properties,* const Ex & *one,* const Ex & *two,* int *mod_prel,* bool *checksets,* int *compare_multiplier,* bool *literal_wildcards*)

8.1.3.50 bool cadabra::tree_exact_less (const Properties * *properties,* const Ex & *one,* const Ex & *two,* int *mod_prel,* bool *checksets,* int *compare_multiplier,* bool *literal_wildcards*)

8.1.3.51 bool cadabra::tree_less (const Properties * *properties,* const Ex & *one,* const Ex & *two,* int *mod_prel,* bool *checksets,* int *compare_multiplier*)

Various comparison functions, some exact, some with pattern logic.

8.1.3.52 const char * cadabra::unichar (kunichar *c*)

8.1.3.53 void cadabra::zero (rset_t::iterator & *num*)

8.1.4 Variable Documentation

8.1.4.1 nset_t cadabra::name_set

8.1.4.2 rset_t cadabra::rat_set

8.2 cadabra2_defaults Namespace Reference

Classes

- class Server
- class Sympy

Functions

- def display (obj, delay_send=False)
- def _displayhook (arg)
- def post_process (__cdbkernel__, ex)

Variables

- __cdbkernel__ = cadabra2.__cdbkernel__
- sympy
- string mopen = "\\begin{dmath*}{}"
- string mclose = "\\end{dmath*}"
- server
- bool have_matplotlib = True
- remember_display_hook = sys.displayhook
- displayhook

8.2.1 Function Documentation

8.2.1.1 def cadabra2_defaults._displayhook (*arg*) [private]

8.2.1.2 def cadabra2_defaults.display (*obj, delay_send* = `False`)

8.2.1.3 def cadabra2_defaults.post_process (*__cdbkernel__, ex*)

8.2.2 Variable Documentation

8.2.2.1 cadabra2_defaults.__cdbkernel__ = cadabra2.__cdbkernel__ [private]

8.2.2.2 cadabra2_defaults.displayhook

8.2.2.3 bool cadabra2_defaults.have_matplotlib = True

8.2.2.4 string cadabra2_defaults.mclose = "\\end{dmath*}"

8.2.2.5 string cadabra2_defaults.mopen = "\\begin{dmath*}{}"

8.2.2.6 cadabra2_defaults.remember_display_hook = sys.displayhook

8.2.2.7 cadabra2_defaults.server

8.2.2.8 cadabra2_defaults.sympy

8.3 cadabra::symbols Namespace Reference

Variables

- const std::set< std::string > greek

8.3.1 Variable Documentation

8.3.1.1 const std::set< std::string > cadabra::symbols::greek

8.4 cadabra_osx Namespace Reference

Classes

- class NotebookWindow

  Objective-C++ class implementing DocumentThread and providing an OS-X notebook interface.

8.5 combin Namespace Reference

Classes

- class combinations
- class combinations_base
- class symm_helper
- class symm_val_helper
- class symmetriser

Typedefs

- typedef std::vector< unsigned int > range_t
- typedef std::vector< range_t > range_vector_t
- typedef std::vector< int > weights_t

Functions

- unsigned long factorial (unsigned int x)
- long vector_sum (const std::vector< int > &)

  sum of elements
- unsigned long vector_prod (const std::vector< unsigned int > &)

  product of elements
- unsigned long vector_prod_fact (const std::vector< unsigned int > &)

  product of factorials of elements
- bool operator== (const std::vector< unsigned int > &, const std::vector< unsigned int > &)
- long hash (const std::vector< unsigned int > &)

  compute a hash value for a vector of unsigned ints
- int determine_intersection_ranges (const range_vector_t &prod, const range_vector_t &indv, range_vector_t &target)
- template<class iterator1 , class iterator2 >
  int ordersign (iterator1 b1, iterator1 e1, iterator2 b2, iterator2 e2, int stepsize=1)
- template<class iterator1 >
  int ordersign (iterator1 b1, iterator1 e1)
- template<class T >
  T fact (T x)
- template<class T >
  std::ostream & operator<< (std::ostream &str, const symmetriser< T > &sym)

8.5.1 Typedef Documentation

8.5.1.1 typedef std::vector<unsigned int> combin::range_t

8.5.1.2 typedef std::vector<range_t> combin::range_vector_t

8.5.1.3 typedef std::vector<int> combin::weights_t

8.5.2 Function Documentation

8.5.2.1 int combin::determine_intersection_ranges (const range_vector_t & *prod,* const range_vector_t & *indv,* combin::range_vector_t & *target*)

8.5.2.2 template<class T > T combin::fact (T *x*)

8.5.2.3 unsigned long combin::factorial (unsigned int *x*)

8.5.2.4 long combin::hash (const std::vector< unsigned int > & *one*)

compute a hash value for a vector of unsigned ints

8.5.2.5 template<class T > std::ostream & combin::operator<< (std::ostream & *str,* const symmetriser< T > & *sym*)

8.5.2.6 bool combin::operator== (const std::vector< unsigned int > & *one,* const std::vector< unsigned int > & *two*)

8.5.2.7 template<class iterator1 , class iterator2 > int combin::ordersign (iterator1 *b1,* iterator1 *e1,* iterator2 *b2,* iterator2 *e2,* int *stepsize* = 1)

8.5.2.8 template<class iterator1 > int combin::ordersign (iterator1 *b1,* iterator1 *e1*)

8.5.2.9 unsigned long combin::vector_prod (const std::vector< unsigned int > & *v*)

product of elements

8.5.2.10 unsigned long combin::vector_prod_fact (const std::vector< unsigned int > & *v*)

product of factorials of elements

8.5.2.11 long combin::vector_sum (const std::vector< int > & *v*)

sum of elements

8.6 echokernel Namespace Reference

Classes

- class EchoKernel

Variables

- kernel_class

8.6.1 Variable Documentation

8.6.1.1 echokernel.kernel_class

8.7 snoop Namespace Reference

Classes

- class Flush
- class Snoop

  Logging class with functionality to send log information to a remote server using a websocket connection.
- class SnoopImpl

Variables

- Flush flush
- Snoop log
- const char info [ ] ="info"
- const char warn [ ] ="warning"
- const char error [ ] ="error"
- const char fatal [ ] ="fatal"
- const char email [ ] ="email"

8.7.1 Variable Documentation

8.7.1.1 const char snoop::email[ ] ="email"

8.7.1.2 const char snoop::error[ ] ="error"

8.7.1.3 const char snoop::fatal[ ] ="fatal"

8.7.1.4 snoop::Flush snoop::flush

8.7.1.5 const char snoop::info[ ] ="info"

8.7.1.6 snoop::Snoop snoop::log

8.7.1.7 const char snoop::warn[ ] ="warning"

8.8 sympy Namespace Reference

Functions

- cadabra::Ex * map_sympy (const cadabra::Kernel &, cadabra::Ex &, const std::vector< std::string > &wrap, const std::string &args, const std::string &method)

  Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place.
- cadabra::Ex::iterator apply (const cadabra::Kernel &, cadabra::Ex &, cadabra::Ex::iterator &, const std↩ ::vector< std::string > &wrap, const std::string &args, const std::string &method)

  Functionality to act with Sympy functions on (parts of) Cadabra Ex expressions and read the result back into the same Ex.
- cadabra::Ex invert_matrix (const cadabra::Kernel &, cadabra::Ex &ex, cadabra::Ex &rules)

  Use Sympy to invert a matrix, given a set of rules determining its sparse components.

8.9 yngtab Namespace Reference

Generic Young tableaux routines.

Classes

- class filled_tableau
- class tableau
- class tableau_base
- class tableaux

Typedefs

- typedef filled_tableau< std::pair< int, int > > keeptrack_tab_t

Functions

- std::ostream & operator<< (std::ostream &str, const tableau &tab)
- template<>
  void add_box (tableau &tab1, unsigned int row1, const tableau &tab2, unsigned int row2, unsigned int col2)
- bool legal_box (const std::vector< std::pair< int, int > > &prev, const std::vector< std::pair< int, int > > &ths, int colpos, int trypos)
- template<class T , class OutputIterator >
  void LR_tensor (const tableaux< T > &tabs1, const T &tab2, unsigned int maxrows, OutputIterator out, bool alltabs=false)
- template<class T1 , class T2 >
  void add_box (T1 &tab1, unsigned int row1, const T2 &tab2, unsigned int row2, unsigned int col2)
- template<class T1 >
  void add_box (T1 &tab1, unsigned int row1, const tableau &tab2, unsigned int row2, unsigned int col2)

- template<class Tab , class OutputIterator >
  void LR_add_box (const Tab &tab2, Tab &newtab, unsigned int currow2, unsigned int curcol2, unsigned int startrow, unsigned int maxrows, OutputIterator outit, keeptrack_tab_t &Ycurrent, bool alltabs)
- template<class Tab , class OutputIterator >
  void LR_tensor (const Tab &tab1, const Tab &tab2, unsigned int maxrows, OutputIterator outit, bool alltabs=false)
- template<class T , class OutputIterator >
  void LR_tensor (const tableaux< T > &, bool symmetric, unsigned int maxrows, OutputIterator outit)
- template<class T >
  std::ostream & operator<< (std::ostream &, const tableaux< T > &)
- template<class T >
  std::ostream & operator<< (std::ostream &, const filled_tableau< T > &)

8.9.1 Detailed Description

Generic Young tableaux routines.

8.9.2 Typedef Documentation

8.9.2.1 typedef filled_tableau<std::pair<int, int> > yngtab::keeptrack_tab_t

8.9.3 Function Documentation

8.9.3.1 template<> void yngtab::add_box (tableau & *tab1,* unsigned int *row1,* const tableau & *tab2,* unsigned int *row2,* unsigned int *col2*)

8.9.3.2 template<class T1 , class T2 > void yngtab::add_box (T1 & *tab1,* unsigned int *row1,* const T2 & *tab2,* unsigned int *row2,* unsigned int *col2*)

8.9.3.3 template<class T1 > void yngtab::add_box (T1 & *tab1,* unsigned int *row1,* const tableau & *tab2,* unsigned int *row2,* unsigned int *col2*)

8.9.3.4 bool yngtab::legal_box (const std::vector< std::pair< int, int > > & *prev,* const std::vector< std::pair< int, int > > & *ths,* int *colpos,* int *trypos*)

8.9.3.5 template<class Tab , class OutputIterator > void yngtab::LR_add_box (const Tab & *tab2,* Tab & *newtab,* unsigned int *currow2,* unsigned int *curcol2,* unsigned int *startrow,* unsigned int *maxrows,* OutputIterator *outit,* keeptrack_tab_t & *Ycurrent,* bool *alltabs*)

8.9.3.6 template<class T , class OutputIterator > void yngtab::LR_tensor (const tableaux< T > & *tabs1,* const T & *tab2,* unsigned int *maxrows,* OutputIterator *out,* bool *alltabs* = `false`)

8.9.3.7 template<class Tab , class OutputIterator > void yngtab::LR_tensor (const Tab & *tab1,* const Tab & *tab2,* unsigned int *maxrows,* OutputIterator *outit,* bool *alltabs* = `false`)

8.9.3.8 template<class T , class OutputIterator > void yngtab::LR_tensor (const tableaux< T > & , bool *symmetric,* unsigned int *maxrows,* OutputIterator *outit*)

8.9.3.9 std::ostream & yngtab::operator<< (std::ostream & *str,* const tableau & *tab*)

8.9.3.10 template<class T > std::ostream & yngtab::operator<< (std::ostream & *str,* const tableaux< T > & *tabs*)

8.9.3.11 template<class T > std::ostream & yngtab::operator<< (std::ostream & *str,* const filled_tableau< T > & *tab*)

Chapter 9

Class Documentation

9.1 cadabra::Accent Class Reference

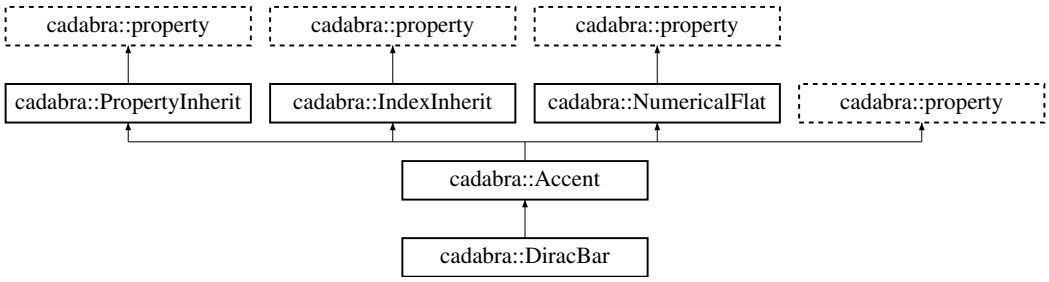
9.1.1 Description

Turns a symbol into an accent.

Accented objects inherit all properties and indices from the objects which they wrap. Here is an example with inherited coordinate dependence: {screen}{1,2,3,4,5,6} {#}::Accent. {#}::PartialDerivative. A::Depends(). (A {A} B): !(%): !(%); (A) * A * B + A * ({A}) * B; {screen} Without the accent property, |{A}| object would be a completely independent object, unrelated to |A|.

`#include <Accent.hh>`

Inheritance diagram for cadabra::Accent:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.1.2 Member Function Documentation

9.1.2.1 std::string Accent::name ( ) const [virtual]

Implements cadabra::property.

Reimplemented in cadabra::DiracBar.

The documentation for this class was generated from the following files:

- core/properties/Accent.hh
- core/properties/Accent.cc

9.2 preprocessor::accu_t Class Reference

Public Member Functions

- accu_t ()
- void erase ()

Public Attributes

- bool head_is_generated
- std::string accu
- unsigned int order
- std::vector< std::string > parts
- unsigned int bracket
- bool is_index

9.2.1 Constructor & Destructor Documentation

9.2.1.1 preprocessor::accu_t::accu_t ( )

9.2.2 Member Function Documentation

9.2.2.1 void preprocessor::accu_t::erase ( )

9.2.3 Member Data Documentation

9.2.3.1 std::string preprocessor::accu_t::accu

9.2.3.2 unsigned int preprocessor::accu_t::bracket

9.2.3.3 bool preprocessor::accu_t::head_is_generated

9.2.3.4 bool preprocessor::accu_t::is_index

9.2.3.5 unsigned int preprocessor::accu_t::order

9.2.3.6 std::vector<std::string> preprocessor::accu_t::parts

The documentation for this class was generated from the following files:

- core/PreProcessor.hh
- core/PreProcessor.cc

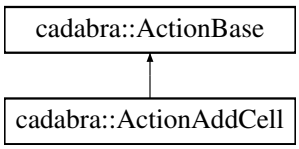
9.3 cadabra::ActionAddCell Class Reference

9.3.1 Description

Add a cell to the notebook.

`#include <Actions.hh>`

Inheritance diagram for cadabra::ActionAddCell:



Public Types

- enum Position { Position::before, Position::after, Position::child }

Public Member Functions

- ActionAddCell (DataCell, DTree::iterator ref_, Position pos_)
- virtual void execute (DocumentThread &, GUIBase &) override

  Perform the action.
- virtual void revert (DocumentThread &, GUIBase &) override

  Revert the change to the DTree document and the GUI.

Private Attributes

- DataCell newcell
- DTree::iterator ref
- DTree::iterator newref
- Position pos
- int child_num

9.3.2 Member Enumeration Documentation

9.3.2.1 enum cadabra::ActionAddCell::Position [strong]

Enumerator

before

after

child

9.3.3 Constructor & Destructor Documentation

9.3.3.1 ActionAddCell::ActionAddCell (DataCell *cell*, DTree::iterator *ref_*, Position *pos_*)

9.3.4 Member Function Documentation

9.3.4.1 void ActionAddCell::execute (DocumentThread & , GUIBase &) [override], [virtual]

Perform the action.

This should update both the document tree data structure and the GUI. The latter is updated by calling relevant methods on the GUIBase object passed in.

Implements cadabra::ActionBase.

9.3.4.2 void ActionAddCell::revert (DocumentThread & , GUIBase &) [override], [virtual]

Revert the change to the DTree document and the GUI.

Implements cadabra::ActionBase.

9.3.5 Member Data Documentation

9.3.5.1 int cadabra::ActionAddCell::child_num [private]

9.3.5.2 DataCell cadabra::ActionAddCell::newcell [private]

9.3.5.3 DTree::iterator cadabra::ActionAddCell::newref [private]

9.3.5.4 Position cadabra::ActionAddCell::pos [private]

9.3.5.5 DTree::iterator cadabra::ActionAddCell::ref [private]

The documentation for this class was generated from the following files:

- client_server/Actions.hh
- client_server/Actions.cc

9.4 cadabra::ActionBase Class Reference

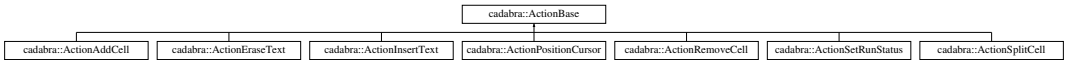
9.4.1 Description

All actions derive from the ActionBase object, which defines the interface they need to implement.

These objects are used to pass (user) action instructions around. They can be stored in undo/redo stacks. All actions run on the GUI thread. The update_gui members typically call members of the GUIBase class. Action objects are allowed to modify the DTree document doc, since they essentially contain code which is part of the DocumentThread object.

```
#include <Actions.hh>
```

Inheritance diagram for cadabra::ActionBase:



Public Member Functions

- virtual void execute (DocumentThread &, GUIBase &)=0

  Perform the action.
- virtual void revert (DocumentThread &, GUIBase &)=0

  Revert the change to the DTree document and the GUI.
- virtual bool undoable () const

  Can this action be undone?

9.4.2 Member Function Documentation

9.4.2.1 virtual void cadabra::ActionBase::execute (DocumentThread & , GUIBase &) [pure virtual]

Perform the action.

This should update both the document tree data structure and the GUI. The latter is updated by calling relevant methods on the GUIBase object passed in.

Implemented in cadabra::ActionEraseText, cadabra::ActionInsertText, cadabra::ActionSplitCell, cadabra::Action↩RemoveCell, cadabra::ActionSetRunStatus, cadabra::ActionPositionCursor, and cadabra::ActionAddCell.

9.4.2.2 virtual void cadabra::ActionBase::revert (DocumentThread & , GUIBase &) [pure virtual]

Revert the change to the DTree document and the GUI.

Implemented in cadabra::ActionEraseText, cadabra::ActionInsertText, cadabra::ActionSplitCell, cadabra::Action↩RemoveCell, cadabra::ActionSetRunStatus, cadabra::ActionPositionCursor, and cadabra::ActionAddCell.

9.4.2.3 bool ActionBase::undoable ( ) const [virtual]

Can this action be undone?

Reimplemented in cadabra::ActionSetRunStatus.

The documentation for this class was generated from the following files:

- client_server/Actions.hh
- client_server/Actions.cc

9.5 cadabra::ActionEraseText Class Reference

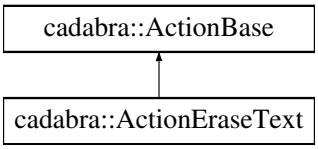
9.5.1 Description

Remove a text string starting at the indicated position, and with the indicated length, from the indicated cell.

This action is assumed to be triggered from a user change to the GUI cells, so will not update the GUI itself, only the underlying DTree. However, the revert method will need to update the GUI representation.

```
#include <Actions.hh>
```

Inheritance diagram for cadabra::ActionEraseText:



Public Member Functions

- ActionEraseText (DTree::iterator, int, int)
- virtual void execute (DocumentThread &, GUIBase &) override

  Perform the action.
- virtual void revert (DocumentThread &, GUIBase &) override

  Revert the change to the DTree document and the GUI.

Private Attributes

- DTree::iterator this_cell
- int from_pos
- int to_pos
- std::string removed_text

9.5.2 Constructor & Destructor Documentation

9.5.2.1 ActionEraseText::ActionEraseText (DTree::iterator *ref_*, int *start*, int *end*)

9.5.3 Member Function Documentation

9.5.3.1 void ActionEraseText::execute (DocumentThread &, GUIBase &) [override], [virtual]

Perform the action.

This should update both the document tree data structure and the GUI. The latter is updated by calling relevant methods on the GUIBase object passed in.

Implements cadabra::ActionBase.

9.5.3.2 void ActionEraseText::revert (DocumentThread &, GUIBase &) [override], [virtual]

Revert the change to the DTree document and the GUI.

Implements cadabra::ActionBase.

9.5.4 Member Data Documentation

9.5.4.1 int cadabra::ActionEraseText::from_pos [private]

9.5.4.2 std::string cadabra::ActionEraseText::removed_text [private]

9.5.4.3 DTree::iterator cadabra::ActionEraseText::this_cell [private]

9.5.4.4 int cadabra::ActionEraseText::to_pos [private]

The documentation for this class was generated from the following files:

- client_server/Actions.hh
- client_server/Actions.cc

9.6 cadabra::ActionInsertText Class Reference

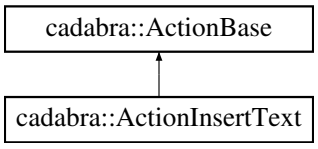
9.6.1 Description

Add a text string (can be just a single character) at the point of the cursor.

This action is assumed to be triggered from a user change to the GUI cells, so will not update the GUI itself, only the underlying DTree. However, the revert method will need to update the GUI representation.

```
#include <Actions.hh>
```

Inheritance diagram for cadabra::ActionInsertText:



Public Member Functions

- ActionInsertText (DTree::iterator, int pos, const std::string &)
- virtual void execute (DocumentThread &, GUIBase &) override

  Perform the action.
- virtual void revert (DocumentThread &, GUIBase &) override

  Revert the change to the DTree document and the GUI.

Private Attributes

- DTree::iterator this_cell
- int insert_pos
- std::string text

9.6.2 Constructor & Destructor Documentation

9.6.2.1 ActionInsertText::ActionInsertText (DTree::iterator *ref_*, int *pos*, const std::string & *content*)

9.6.3 Member Function Documentation

9.6.3.1 void ActionInsertText::execute (DocumentThread & , GUIBase &) [override], [virtual]

Perform the action.

This should update both the document tree data structure and the GUI. The latter is updated by calling relevant methods on the GUIBase object passed in.

Implements cadabra::ActionBase.

9.6.3.2 void ActionInsertText::revert (DocumentThread & , GUIBase &) [override], [virtual]

Revert the change to the DTree document and the GUI.

Implements cadabra::ActionBase.

9.6.4 Member Data Documentation

9.6.4.1 int cadabra::ActionInsertText::insert_pos [private]

9.6.4.2 std::string cadabra::ActionInsertText::text [private]

9.6.4.3 DTree::iterator cadabra::ActionInsertText::this_cell [private]

The documentation for this class was generated from the following files:

- client_server/Actions.hh
- client_server/Actions.cc

9.7 cadabra::ActionPositionCursor Class Reference

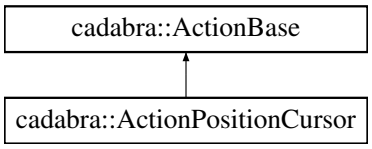
9.7.1 Description

Position the cursor relative to the indicated cell.

If position is 'next' and there is no input cell following the indicated one, create a new one.

`#include <Actions.hh>`

Inheritance diagram for cadabra::ActionPositionCursor:



Public Types

- enum Position { Position::in, Position::next, Position::previous }

Public Member Functions

- ActionPositionCursor (DTree::iterator ref_, Position pos_)
- virtual void execute (DocumentThread &, GUIBase &) override

  Perform the action.
- virtual void revert (DocumentThread &, GUIBase &) override

  Revert the change to the DTree document and the GUI.

Private Attributes

- bool needed_new_cell
- DTree::iterator ref
- DTree::iterator newref
- Position pos

9.7.2 Member Enumeration Documentation

9.7.2.1 enum cadabra::ActionPositionCursor::Position [strong]

Enumerator

in

next

previous

9.7.3 Constructor & Destructor Documentation

9.7.3.1 ActionPositionCursor::ActionPositionCursor (DTree::iterator *ref_*, Position *pos_*)

9.7.4 Member Function Documentation

9.7.4.1 void ActionPositionCursor::execute (DocumentThread & , GUIBase &) [override], [virtual]

Perform the action.

This should update both the document tree data structure and the GUI. The latter is updated by calling relevant methods on the GUIBase object passed in.

Implements cadabra::ActionBase.

9.7.4.2 void ActionPositionCursor::revert (DocumentThread & , GUIBase &) [override], [virtual]

Revert the change to the DTree document and the GUI.

Implements cadabra::ActionBase.

9.7.5 Member Data Documentation

9.7.5.1 bool cadabra::ActionPositionCursor::needed_new_cell [private]

9.7.5.2 DTree::iterator cadabra::ActionPositionCursor::newref [private]

9.7.5.3 Position cadabra::ActionPositionCursor::pos [private]

9.7.5.4 DTree::iterator cadabra::ActionPositionCursor::ref [private]

The documentation for this class was generated from the following files:

- client_server/Actions.hh
- client_server/Actions.cc

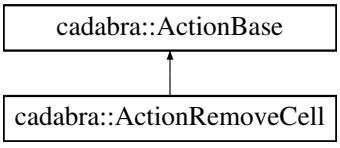
9.8 cadabra::ActionRemoveCell Class Reference

9.8.1 Description

Remove a cell and all its child cells from the document.

`#include <Actions.hh>`

Inheritance diagram for cadabra::ActionRemoveCell:



Public Member Functions

- ActionRemoveCell (DTree::iterator ref_)
- ~ActionRemoveCell ()
- virtual void execute (DocumentThread &, GUIBase &) override

  Perform the action.
- virtual void revert (DocumentThread &, GUIBase &) override

  Revert the change to the DTree document and the GUI.

Private Attributes

- DTree removed_tree
- DTree::iterator reference_parent_cell
- DTree::iterator this_cell
- size_t reference_child_index

9.8.2 Constructor & Destructor Documentation

9.8.2.1 ActionRemoveCell::ActionRemoveCell (DTree::iterator *ref_*)

9.8.2.2 ActionRemoveCell::~ActionRemoveCell ( )

9.8.3 Member Function Documentation

9.8.3.1 void ActionRemoveCell::execute (DocumentThread & , GUIBase &) [override], [virtual]

Perform the action.

This should update both the document tree data structure and the GUI. The latter is updated by calling relevant methods on the GUIBase object passed in.

Implements cadabra::ActionBase.

9.8.3.2 void ActionRemoveCell::revert (DocumentThread & , GUIBase &) [override], [virtual]

Revert the change to the DTree document and the GUI.

Implements cadabra::ActionBase.

9.8.4 Member Data Documentation

9.8.4.1 size_t cadabra::ActionRemoveCell::reference_child_index [private]

9.8.4.2 DTree::iterator cadabra::ActionRemoveCell::reference_parent_cell [private]

9.8.4.3 DTree cadabra::ActionRemoveCell::removed_tree [private]

9.8.4.4 DTree::iterator cadabra::ActionRemoveCell::this_cell [private]

The documentation for this class was generated from the following files:

- client_server/Actions.hh
- client_server/Actions.cc

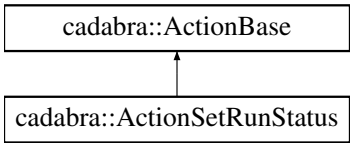
9.9 cadabra::ActionSetRunStatus Class Reference

9.9.1 Description

Update the running status of the indicated cell.

`#include <Actions.hh>`

Inheritance diagram for cadabra::ActionSetRunStatus:



Public Member Functions

- ActionSetRunStatus (DTree::iterator ref_, bool running)
- virtual void execute (DocumentThread &, GUIBase &) override

  Perform the action.
- virtual void revert (DocumentThread &, GUIBase &) override

  Revert the change to the DTree document and the GUI.
- virtual bool undoable () const override

  Can this action be undone?

Private Attributes

- DTree::iterator this_cell
- bool was_running_
- bool new_running_

9.9.2 Constructor & Destructor Documentation

9.9.2.1 ActionSetRunStatus::ActionSetRunStatus (DTree::iterator *ref_*, bool *running*)

9.9.3 Member Function Documentation

9.9.3.1 void ActionSetRunStatus::execute (DocumentThread &, GUIBase &) [override], [virtual]

Perform the action.

This should update both the document tree data structure and the GUI. The latter is updated by calling relevant methods on the GUIBase object passed in.

Implements cadabra::ActionBase.

9.9.3.2 void ActionSetRunStatus::revert (DocumentThread & , GUIBase &) [override], [virtual]

Revert the change to the DTree document and the GUI.

Implements cadabra::ActionBase.

9.9.3.3 bool ActionSetRunStatus::undoable ( ) const [override], [virtual]

Can this action be undone?

Reimplemented from cadabra::ActionBase.

9.9.4 Member Data Documentation

9.9.4.1 bool cadabra::ActionSetRunStatus::new_running_ [private]

9.9.4.2 DTree::iterator cadabra::ActionSetRunStatus::this_cell [private]

9.9.4.3 bool cadabra::ActionSetRunStatus::was_running_ [private]

The documentation for this class was generated from the following files:

- client_server/Actions.hh
- client_server/Actions.cc

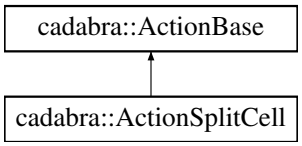
9.10 cadabra::ActionSplitCell Class Reference

9.10.1 Description

Split a cell into two separate cells, at the point of the cursor.

`#include <Actions.hh>`

Inheritance diagram for cadabra::ActionSplitCell:



Public Member Functions

- ActionSplitCell (DTree::iterator ref_)
- ~ActionSplitCell ()
- virtual void execute (DocumentThread &, GUIBase &) override
  Perform the action.
- virtual void revert (DocumentThread &, GUIBase &) override
  Revert the change to the DTree document and the GUI.

Private Attributes

- DTree::iterator this_cell
- DTree::iterator newref

9.10.2 Constructor & Destructor Documentation

9.10.2.1 ActionSplitCell::ActionSplitCell (DTree::iterator *ref_*)

9.10.2.2 ActionSplitCell::~ActionSplitCell ( )

9.10.3 Member Function Documentation

9.10.3.1 void ActionSplitCell::execute (DocumentThread & , GUIBase &) [override], [virtual]

Perform the action.

This should update both the document tree data structure and the GUI. The latter is updated by calling relevant methods on the GUIBase object passed in.

Implements cadabra::ActionBase.

9.10.3.2 void ActionSplitCell::revert (DocumentThread & , GUIBase &) [override], [virtual]

Revert the change to the DTree document and the GUI.

Implements cadabra::ActionBase.

9.10.4 Member Data Documentation

9.10.4.1 DTree::iterator cadabra::ActionSplitCell::newref [private]

9.10.4.2 DTree::iterator cadabra::ActionSplitCell::this_cell [private]

The documentation for this class was generated from the following files:

- client_server/Actions.hh
- client_server/Actions.cc

9.11 cadabra::Algorithm Class Reference

9.11.1 Description

Base class for all algorithms, containing generic routines and in particular the logic for index classification.

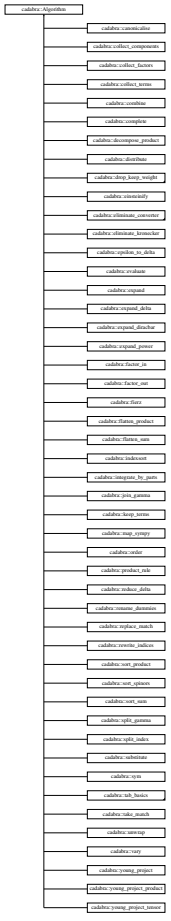
Also contains static algorithms acting on Ex objects which require property information and can therefore not be a member of Ex.

In order to implement a new algorithm, subclass Algorithm and implement the abstract members Algorithm::can_apply and Algorithm::apply (see there for further documentation). The general logic is that the implementation of Algorithm::apply(iterator&) is not allowed to make the node pointed at by the iterator invalid. If the algorithm makes the node vanish, it should indicate so by setting its multiplier to zero; the calling logic will then take care of cleaning up the subtree at the node.

The algorithm is, however, allowed to change the node itself or replace it with another one, as long as it updates the iterator.

```
#include <Algorithm.hh>
```

Inheritance diagram for cadabra::Algorithm:



Public Types

- typedef Ex::iterator iterator
- typedef Ex::post_order_iterator post_order_iterator
- typedef Ex::sibling_iterator sibling_iterator
- typedef Ex::result_t result_t
- typedef std::multimap< Ex, Ex::iterator, tree_exact_less_for_indexmap_obj > index_map_t

  A map from a pattern to the position where it occurs in the tree.
- typedef std::map< Ex::iterator, int, Ex::iterator_base_less > index_position_map_t

  A map from the position of each index to the sequential index.

Public Member Functions

- Algorithm (const Kernel &, Ex &)

  Initialise the algorithm with a reference to the expression tree, but do not yet do anything with this tree.
- virtual ~Algorithm ()
- void set_progress_monitor (ProgressMonitor *)

  Provide the algorithm with a ProgressMonitor object on which to register (nested) progress information, to be reported out-of-band to a client.
- result_t apply_generic (bool deep=true, bool repeat=false, unsigned int depth=0)

  The main entry points for running algorithms, which traverse the tree post-order ('child before parent').
- result_t apply_generic (iterator &, bool deep, bool repeat, unsigned int depth)
- result_t apply_pre_order (bool repeat=false)

  Apply algorithm with alternative traversal: starting from the top node, traverse the tree pre-order ('parent before child') and once the algorithm acts at a given node, do not traverse the subtree below anymore.
- bool check_consistency (iterator) const

  Given an expression top node, check index consistency.
- bool check_index_consistency (iterator) const
- bool check_degree_consistency (iterator) const

  Given an expression top node, check differential form degree consistency.
- void report_progress (const std::string &, int todo, int done, int count=2)
- index_iterator begin_index (iterator it) const
- index_iterator end_index (iterator it) const
- unsigned int number_of_indices (iterator it)
- bool rename_replacement_dummies (iterator, bool still_inside_algo=false)

Static Public Member Functions

- static unsigned int number_of_indices (const Properties &, iterator it)
- static unsigned int number_of_direct_indices (iterator it)

Public Attributes

- bool interrupted
- unsigned int number_of_calls
- unsigned int number_of_modifications
- bool suppress_normal_output
- bool discard_command_node
- Stopwatch index_sw
- Stopwatch get_dummy_sw
- Stopwatch report_progress_stopwatch

Protected Types

- typedef std::pair< sibling_iterator, sibling_iterator > range_t

  Finding objects in sets.
- typedef std::vector< range_t > range_vector_t

Protected Member Functions

- virtual bool can_apply (iterator)=0
- virtual result_t apply (iterator &)=0
- int index_parity (iterator) const
- bool contains (sibling_iterator from, sibling_iterator to, sibling_iterator arg)
- void find_argument_lists (range_vector_t &, bool only_comma_lists=true) const
- template<class Iter >
  range_vector_t::iterator find_arg_superset (range_vector_t &, Iter st, Iter nd)
- range_vector_t::iterator find_arg_superset (range_vector_t &, sibling_iterator it)
- unsigned int locate_single_object (Ex::iterator obj_to_find, Ex::iterator st, Ex::iterator nd, std::vector< unsigned int > &store)
- bool locate_object_set (const Ex &objs, Ex::iterator st, Ex::iterator nd, std::vector< unsigned int > &store)
- bool is_termlike (iterator)

  Determine structure (version-2)
- bool is_factorlike (iterator)
- bool is_single_term (iterator)

  Take a single non-product node in a sum and wrap it in a product node, so it can be handled on the same footing as a proper product.
- bool is_nonprod_factor_in_prod (iterator)
- bool prod_wrap_single_term (iterator &)
- bool prod_unwrap_single_term (iterator &)
- void force_prod_wrap (iterator &)

  Wrap a term in a product, irrespective of its parent (it usually makes more sense to call the safer prod_wrap_single_term above).
- bool separated_by_derivative (iterator, iterator, iterator check_dependence) const

  Figure out whether two objects (commonly indices) are separated by a derivative operator, as in

  $$\partial_a A_b C^b$$

  .
- void pushup_multiplier (iterator)
- template<class BinaryPredicate >
  unsigned int intersection_number (sibling_iterator, sibling_iterator, sibling_iterator, sibling_iterator, BinaryPredicate) const

  Determine the number of elements in the first range which also occur in the second range.
- void node_zero (iterator)
- void node_one (iterator)
- void node_integer (iterator, int)
- void fill_index_position_map (iterator, const index_map_t &, index_position_map_t &) const

  Index manipulation and classification.
- void fill_map (index_map_t &, sibling_iterator, sibling_iterator) const
- void print_classify_indices (std::ostream &, iterator) const
- void determine_intersection (index_map_t &one, index_map_t &two, index_map_t &target, bool move_out=false) const
- void classify_add_index (iterator it, index_map_t &ind_free, index_map_t &ind_dummy) const
- void classify_indices_up (iterator, index_map_t &ind_free, index_map_t &ind_dummy) const
- void classify_indices (iterator, index_map_t &ind_free, index_map_t &ind_dummy) const

- int max_numbered_name_one (const std::string &nm, const index_map_t *one) const
- int max_numbered_name (const std::string &, const index_map_t *m1, const index_map_t *m2=0, const index_map_t *m3=0, const index_map_t *m4=0, const index_map_t *m5=0) const
- Ex get_dummy (const list_property *, const index_map_t *m1, const index_map_t *m2=0, const index_map_t *m3=0, const index_map_t *m4=0, const index_map_t *m5=0) const
- Ex get_dummy (const list_property *, iterator) const
- Ex get_dummy (const list_property *, iterator, iterator) const
- bool index_in_set (Ex, const index_map_t *) const
- index_map_t::iterator find_modulo_parent_rel (iterator it, index_map_t &imap) const

  Find an index in the set, not taking into account index position.

Static Protected Member Functions

- static bool less_without_numbers (nset_t::iterator, nset_t::iterator)
- static bool equal_without_numbers (nset_t::iterator, nset_t::iterator)
- static bool compare_ (const str_node &, const str_node &)

Protected Attributes

- const Kernel & kernel
- Ex & tr
- ProgressMonitor * pm

Private Member Functions

- result_t apply_once (Ex::iterator &it)
- result_t apply_deep (Ex::iterator &it)
- void propagate_zeroes (post_order_iterator &, const iterator &)

  Given a node with zero multiplier, propagate this zero upwards in the tree.
- void dumpmap (std::ostream &, const index_map_t &) const

9.11.2 Member Typedef Documentation

9.11.2.1 typedef std::multimap<Ex, Ex::iterator, tree_exact_less_for_indexmap_obj> cadabra::Algorithm::index_map_t

A map from a pattern to the position where it occurs in the tree.

The comparator is such that we store indices exactly, apart from their multiplicative factor. This means that the index in A_{n} and in A_{-n} are stored in the same way, and one needs to lookup the expression in the tree to find this multiplier. See basic.cdb test 26 for an example that uses this.

9.11.2.2 typedef std::map<Ex::iterator, int, Ex::iterator_base_less> cadabra::Algorithm::index_position_map_t

A map from the position of each index to the sequential index.

9.11.2.3 typedef Ex::iterator cadabra::Algorithm::iterator

9.11.2.4 typedef Ex::post_order_iterator cadabra::Algorithm::post_order_iterator

9.11.2.5 typedef std::pair<sibling_iterator, sibling_iterator> cadabra::Algorithm::range_t [protected]

Finding objects in sets.

9.11.2.6 typedef std::vector<range_t> cadabra::Algorithm::range_vector_t [protected]

9.11.2.7 typedef Ex::result_t cadabra::Algorithm::result_t

9.11.2.8 typedef Ex::sibling_iterator cadabra::Algorithm::sibling_iterator

9.11.3 Constructor & Destructor Documentation

9.11.3.1 Algorithm::Algorithm (const Kernel & *k*, Ex & *tr_*)

Initialise the algorithm with a reference to the expression tree, but do not yet do anything with this tree.

Algorithms are not typically allowed to mess with the settings in the Kernel, so it is passed const.

9.11.3.2 Algorithm::~Algorithm ( ) [virtual]

9.11.4 Member Function Documentation

9.11.4.1 virtual result_t cadabra::Algorithm::apply (iterator &) [protected], [pure virtual]

Implemented in cadabra::evaluate, cadabra::keep_weight, cadabra::substitute, cadabra::drop_weight, cadabra::complete, cadabra::decompose_product, cadabra::distribute, cadabra::integrate_by_parts, cadabra::map_sympy, cadabra::vary, cadabra::canonicalise, cadabra::collect_components, cadabra::collect_factors, cadabra::collect_terms, cadabra::einsteinify, cadabra::factor_in, cadabra::factor_out, cadabra::fierz, cadabra::split_index, cadabra::young_project, cadabra::epsilon_to_delta, cadabra::split_gamma, cadabra::young_project_tensor, cadabra::combine, cadabra::eliminate_kronecker, cadabra::eliminate_converter, cadabra::expand, cadabra::expand_delta, cadabra::expand_diracbar, cadabra::expand_power, cadabra::flatten_product, cadabra::join_gamma, cadabra::keep_terms, cadabra::order, cadabra::reduce_delta, cadabra::rename_dummies, cadabra::replace_match, cadabra::rewrite_indices, cadabra::sort_product, cadabra::sort_spinors, cadabra::sort_sum, cadabra::sym, cadabra::take_match, cadabra::young_project_product, cadabra::indexsort, cadabra::lr_tensor, cadabra::flatten_sum, cadabra::product_rule, and cadabra::unwrap.

9.11.4.2 Algorithm::result_t Algorithm::apply_deep (Ex::iterator & *it*) [private]

9.11.4.3 Algorithm::result_t Algorithm::apply_generic (bool *deep* = true, bool *repeat* = false, unsigned int *depth* = 0)

The main entry points for running algorithms, which traverse the tree post-order ('child before parent').

The 'deep' flag indicates whether sub-expressions should be acted on too. The 'repeat' flag indicates whether the algorithm should be applied until the expression no longer changes. The 'depth' flag, if not equal to -1, indicates the depth in the tree where the algorithm should start applying.

9.11.4.4 Algorithm::result_t Algorithm::apply_generic (Ex::iterator & *it,* bool *deep,* bool *repeat,* unsigned int *depth*)

9.11.4.5 Algorithm::result_t Algorithm::apply_once (Ex::iterator & *it*) [private]

9.11.4.6 Algorithm::result_t Algorithm::apply_pre_order (bool *repeat* = false)

Apply algorithm with alternative traversal: starting from the top node, traverse the tree pre-order ('parent before child') and once the algorithm acts at a given node, do not traverse the subtree below anymore.

9.11.4.7 index_iterator Algorithm::begin_index (iterator *it*) const

9.11.4.8 virtual bool cadabra::Algorithm::can_apply (iterator) [protected], [pure virtual]

Implemented in cadabra::evaluate, cadabra::substitute, cadabra::complete, cadabra::decompose_product, cadabra::distribute, cadabra::integrate_by_parts, cadabra::map_sympy, cadabra::vary, cadabra::canonicalise, cadabra::collect_components, cadabra::collect_factors, cadabra::collect_terms, cadabra::drop_keep_weight, cadabra::einsteinify, cadabra::factor_in, cadabra::factor_out, cadabra::fierz, cadabra::split_index, cadabra::young_project, cadabra::epsilon_to_delta, cadabra::split_gamma, cadabra::young_project_tensor, cadabra::combine, cadabra::eliminate_kronecker, cadabra::eliminate_converter, cadabra::expand, cadabra::expand_delta, cadabra::expand_diracbar, cadabra::expand_power, cadabra::flatten_product, cadabra::join_gamma, cadabra::keep_terms, cadabra::order, cadabra::reduce_delta, cadabra::rename_dummies, cadabra::replace_match, cadabra::rewrite_indices, cadabra::sort_product, cadabra::sort_spinors, cadabra::sort_sum, cadabra::sym, cadabra::take_match, cadabra::young_project_product, cadabra::indexsort, cadabra::lr_tensor, cadabra::flatten_sum, cadabra::product_rule, and cadabra::unwrap.

9.11.4.9 bool Algorithm::check_consistency (iterator *it*) const

Given an expression top node, check index consistency.

9.11.4.10 bool Algorithm::check_degree_consistency (iterator *it*) const

Given an expression top node, check differential form degree consistency.

9.11.4.11 bool Algorithm::check_index_consistency (iterator *it*) const

9.11.4.12 void Algorithm::classify_add_index (iterator *it,* index_map_t & *ind_free,* index_map_t & *ind_dummy*) const [protected]

9.11.4.13 void Algorithm::classify_indices (iterator *it,* index_map_t & *ind_free,* index_map_t & *ind_dummy*) const [protected]

9.11.4.14 void Algorithm::classify_indices_up (iterator *it,* index_map_t & *ind_free,* index_map_t & *ind_dummy*) const [protected]

9.11.4.15 bool Algorithm::compare_ (const str_node & *one,* const str_node & *two*) [static], [protected]

9.11.4.16 bool Algorithm::contains (sibling_iterator *from,* sibling_iterator *to,* sibling_iterator *arg*) [protected]

9.11.4.17 void Algorithm::determine_intersection (index_map_t & *one,* index_map_t & *two,* index_map_t & *target,* bool *move_out* = false) const [protected]

9.11.4.18 void Algorithm::dumpmap (std::ostream & *str,* const index_map_t & *mp*) const [private]

9.11.4.19 index_iterator Algorithm::end_index (iterator *it*) const

9.11.4.20 bool Algorithm::equal_without_numbers (nset_t::iterator *it1,* nset_t::iterator *it2*) [static], [protected]

9.11.4.21 void Algorithm::fill_index_position_map (iterator *prodnode,* const index_map_t & *im,* index_position_map_t & *ipm*) const [protected]

Index manipulation and classification.

Routines to find and classify all indices in an expression, taking into account sums and products. Note that dummy indices do not always come in pairs, for instance in expressions like a_{m n} (b^{n p} + q^{n p}) . Similarly, free indices can appear multiple times, as in a_{m} + b_{m} .

9.11.4.22 void Algorithm::fill_map (index_map_t & *mp,* sibling_iterator *st,* sibling_iterator *nd*) const [protected]

9.11.4.23 template<class Iter > Algorithm::range_vector_t::iterator Algorithm::find_arg_superset (range_vector_t & *ran,* Iter *st,* Iter *nd*) [protected]

9.11.4.24 Algorithm::range_vector_t::iterator Algorithm::find_arg_superset (range_vector_t & *ran,* sibling_iterator *it*) [protected]

9.11.4.25 void cadabra::Algorithm::find_argument_lists (range_vector_t & , bool *only_comma_lists* = true) const [protected]

9.11.4.26 Algorithm::index_map_t::iterator Algorithm::find_modulo_parent_rel (iterator *it,* index_map_t & *imap*) const [protected]

Find an index in the set, not taking into account index position.

9.11.4.27 void Algorithm::force_prod_wrap (iterator & *it*) [protected]

Wrap a term in a product, irrespective of its parent (it usually makes more sense to call the safer prod_wrap_↩ single_term above).

9.11.4.28 **Ex Algorithm::get_dummy (const list_property * *dums,* const index_map_t * *m1,* const index_map_t * *m2* = 0, const index_map_t * *m3* = 0, const index_map_t * *m4* = 0, const index_map_t * *m5* = 0) const** [protected]

9.11.4.29 **Ex Algorithm::get_dummy (const list_property * *dums,* iterator *it*) const** [protected]

9.11.4.30 **Ex Algorithm::get_dummy (const list_property * *dums,* iterator *it1,* iterator *it2*) const** [protected]

9.11.4.31 **bool Algorithm::index_in_set (Ex *ex,* const index_map_t * *im*) const** [protected]

9.11.4.32 **int Algorithm::index_parity (iterator *it*) const** [protected]

9.11.4.33 **template<class BinaryPredicate > unsigned int Algorithm::intersection_number (sibling_iterator *from1,* sibling_iterator *to1,* sibling_iterator *from2,* sibling_iterator *to2,* BinaryPredicate *fun*) const** [protected]

Determine the number of elements in the first range which also occur in the second range.

9.11.4.34 **bool Algorithm::is_factorlike (iterator *it*)** [protected]

9.11.4.35 **bool Algorithm::is_nonprod_factor_in_prod (iterator *it*)** [protected]

9.11.4.36 **bool Algorithm::is_single_term (iterator *it*)** [protected]

Take a single non-product node in a sum and wrap it in a product node, so it can be handled on the same footing as a proper product.

9.11.4.37 **bool Algorithm::is_termlike (iterator *it*)** [protected]

Determine structure (version-2)

9.11.4.38 **bool Algorithm::less_without_numbers (nset_t::iterator *it1,* nset_t::iterator *it2*)** [static], [protected]

9.11.4.39 **bool Algorithm::locate_object_set (const Ex & *objs,* Ex::iterator *st,* Ex::iterator *nd,* std::vector< unsigned int > & *store*)** [protected]

9.11.4.40 **unsigned int Algorithm::locate_single_object (Ex::iterator *obj_to_find,* Ex::iterator *st,* Ex::iterator *nd,* std::vector< unsigned int > & *store*)** [protected]

9.11.4.41 **int Algorithm::max_numbered_name (const std::string & *nm,* const index_map_t * *m1,* const index_map_t * *m2* = 0, const index_map_t * *m3* = 0, const index_map_t * *m4* = 0, const index_map_t * *m5* = 0) const** [protected]

9.11.4.42 **int Algorithm::max_numbered_name_one (const std::string & *nm,* const index_map_t * *one*) const** [protected]

9.11.4.43 void Algorithm::node_integer (iterator *it,* int *num*) [protected]

9.11.4.44 void Algorithm::node_one (iterator *it*) [protected]

9.11.4.45 void Algorithm::node_zero (iterator *it*) [protected]

9.11.4.46 unsigned int Algorithm::number_of_direct_indices (iterator *it*) [static]

9.11.4.47 unsigned int Algorithm::number_of_indices (iterator *it*)

9.11.4.48 unsigned int Algorithm::number_of_indices (const Properties & *pr,* iterator *it*) [static]

9.11.4.49 void Algorithm::print_classify_indices (std::ostream & *str,* iterator *st*) const [protected]

9.11.4.50 bool Algorithm::prod_unwrap_single_term (iterator & *it*) [protected]

9.11.4.51 bool Algorithm::prod_wrap_single_term (iterator & *it*) [protected]

9.11.4.52 void Algorithm::propagate_zeroes (post_order_iterator & *it,* const iterator & *topnode*) [private]

Given a node with zero multiplier, propagate this zero upwards in the tree.

Changes the iterator so that it points to the next node in a post_order traversal (post_order: children first, then node). The second node is the topmost node, beyond which this routine is not allowed to touch the tree (i.e. the 2nd iterator will always remain valid).

9.11.4.53 void Algorithm::pushup_multiplier (iterator *it*) [protected]

9.11.4.54 bool Algorithm::rename_replacement_dummies (iterator *two,* bool *still_inside_algo* = false)

9.11.4.55 void Algorithm::report_progress (const std::string & *str,* int *todo,* int *done,* int *count* = 2)

9.11.4.56 bool Algorithm::separated_by_derivative (iterator *i1,* iterator *i2,* iterator *check_dependence*) const [protected]

Figure out whether two objects (commonly indices) are separated by a derivative operator, as in

$$\partial_a A_b C^b$$

.

If the last iterator is pointing to a valid node, check whether it is independent of the derivative (using the Depends property).

9.11.4.57 void Algorithm::set_progress_monitor (ProgressMonitor * *pm_*)

Provide the algorithm with a ProgressMonitor object on which to register (nested) progress information, to be reported out-of-band to a client.

9.11.5 Member Data Documentation

9.11.5.1 bool cadabra::Algorithm::discard_command_node

9.11.5.2 Stopwatch cadabra::Algorithm::get_dummy_sw [mutable]

9.11.5.3 Stopwatch cadabra::Algorithm::index_sw [mutable]

9.11.5.4 bool cadabra::Algorithm::interrupted

9.11.5.5 const Kernel& cadabra::Algorithm::kernel [protected]

9.11.5.6 unsigned int cadabra::Algorithm::number_of_calls

9.11.5.7 unsigned int cadabra::Algorithm::number_of_modifications

9.11.5.8 ProgressMonitor* cadabra::Algorithm::pm [protected]

9.11.5.9 Stopwatch cadabra::Algorithm::report_progress_stopwatch [mutable]

9.11.5.10 bool cadabra::Algorithm::suppress_normal_output

9.11.5.11 Ex& cadabra::Algorithm::tr [protected]

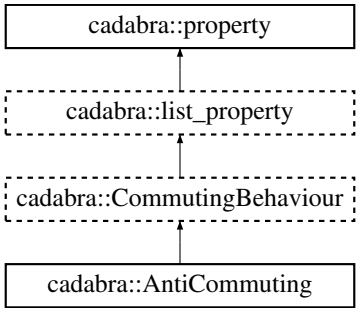
The documentation for this class was generated from the following files:

- core/Algorithm.hh
- core/Algorithm.cc

9.12 cadabra::AntiCommuting Class Reference

`#include <AntiCommuting.hh>`

Inheritance diagram for cadabra::AntiCommuting:



Public Member Functions

- virtual std::string name () const
- virtual int sign () const

Additional Inherited Members

9.12.1 Member Function Documentation

9.12.1.1 std::string AntiCommuting::name ( ) const [virtual]

Implements cadabra::property.

9.12.1.2 virtual int cadabra::AntiCommuting::sign ( ) const [inline], [virtual]

Implements cadabra::CommutingBehaviour.

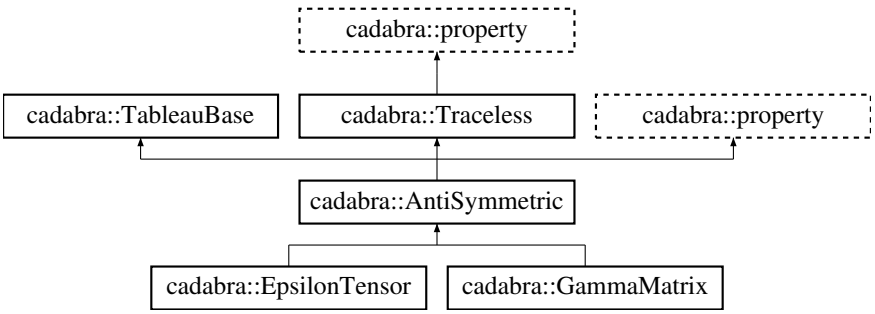
The documentation for this class was generated from the following files:

- core/properties/AntiCommuting.hh
- core/properties/AntiCommuting.cc

9.13 cadabra::AntiSymmetric Class Reference

```
#include <AntiSymmetric.hh>
```

Inheritance diagram for cadabra::AntiSymmetric:



Public Member Functions

- virtual ~AntiSymmetric ()
- virtual std::string name () const override
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const override
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const override

Additional Inherited Members

9.13.1 Constructor & Destructor Documentation

9.13.1.1 virtual cadabra::AntiSymmetric::~AntiSymmetric ( ) [inline], [virtual]

9.13.2 Member Function Documentation

9.13.2.1 TableauBase::tab_t AntiSymmetric::get_tab (const Properties & *pr,* Ex & *tr,* Ex::iterator *it,* unsigned int *num*) const [override], [virtual]

Implements cadabra::TableauBase.

9.13.2.2 std::string AntiSymmetric::name ( ) const [override], [virtual]

Implements cadabra::property.

Reimplemented in cadabra::EpsilonTensor, and cadabra::GammaMatrix.

9.13.2.3 unsigned int AntiSymmetric::size (const Properties & , Ex & , Ex::iterator) const [override], [virtual]

Implements cadabra::TableauBase.

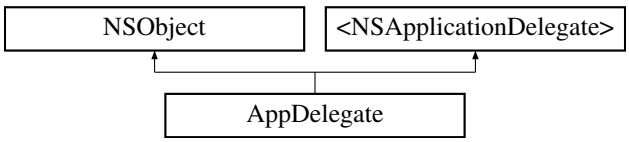
The documentation for this class was generated from the following files:

- core/properties/AntiSymmetric.hh
- core/properties/AntiSymmetric.cc

9.14 AppDelegate Class Reference

```
#import <AppDelegate.h>
```

Inheritance diagram for AppDelegate:



The documentation for this class was generated from the following file:

- frontend/osx/Cadabra/Cadabra/AppDelegate.h

9.15 AppDelegate() Category Reference

Properties

- IBOutlet NSWindow * window
- IBOutlet NotebookController * notebookController

9.15.1 Property Documentation

9.15.1.1 - (IBOutlet NotebookController*) notebookController [read], [write], [nonatomic], [strong]

9.15.1.2 - (IBOutlet NSWindow*) window [read], [write], [atomic], [weak]

The documentation for this category was generated from the following file:

- frontend/osx/Cadabra/Cadabra/AppDelegate.mm

9.16 snoop::Snoop::AppEntry Class Reference

9.16.1 Description

C++ representation of an run entry.

`#include <Snoop.hh>`

Public Member Functions

- AppEntry ()
- AppEntry (const std::string &uuid_, uint64_t create_millis_, uint64_t receive_millis_, pid_t pid_, const std::string &ip_address_, const std::string &machine_id_, const std::string &app_name_, const std::string &app_version_, const std::string &user_id_, int server_status_)
- std::string to_json (bool human_readable) const
- void from_json (const Json::Value &)

Public Attributes

- int id
- std::string uuid
- uint64_t create_millis
- uint64_t receive_millis
- pid_t pid
- std::string ip_address
- std::string machine_id
- std::string app_name
- std::string app_version
- std::string user_id
- int server_status
- bool connected

9.16.2 Constructor & Destructor Documentation

9.16.2.1 Snoop::AppEntry::AppEntry ( )

9.16.2.2 Snoop::AppEntry::AppEntry (const std::string & *uuid_*, uint64_t *create_millis_*, uint64_t *receive_millis_*, pid_t *pid_*, const std::string & *ip_address_*, const std::string & *machine_id_*, const std::string & *app_name_*, const std::string & *app_version_*, const std::string & *user_id_*, int *server_status_*)

9.16.3 Member Function Documentation

9.16.3.1 void Snoop::AppEntry::from_json (const Json::Value & *entry*)

9.16.3.2 std::string Snoop::AppEntry::to_json (bool *human_readable*) const

9.16.4 Member Data Documentation

9.16.4.1 std::string snoop::Snoop::AppEntry::app_name

9.16.4.2 std::string snoop::Snoop::AppEntry::app_version

9.16.4.3 bool snoop::Snoop::AppEntry::connected

9.16.4.4 uint64_t snoop::Snoop::AppEntry::create_millis

9.16.4.5 int snoop::Snoop::AppEntry::id

9.16.4.6 std::string snoop::Snoop::AppEntry::ip_address

9.16.4.7 std::string snoop::Snoop::AppEntry::machine_id

9.16.4.8 pid_t snoop::Snoop::AppEntry::pid

9.16.4.9 uint64_t snoop::Snoop::AppEntry::receive_millis

9.16.4.10 int snoop::Snoop::AppEntry::server_status

9.16.4.11 std::string snoop::Snoop::AppEntry::user_id

9.16.4.12 std::string snoop::Snoop::AppEntry::uuid

The documentation for this class was generated from the following files:

- client_server/Snoop.hh
- client_server/Snoop.cc

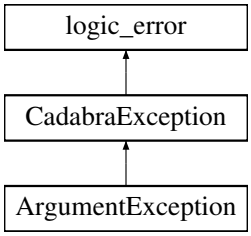
9.17 ArgumentException Class Reference

9.17.1 Description

Exception thrown when arguments to an algorithm or property are not correct.

`#include <Exceptions.hh>`

Inheritance diagram for ArgumentException:



Public Member Functions

- ArgumentException (std::string="")
- std::string py_what () const

9.17.2 Constructor & Destructor Documentation

9.17.2.1 ArgumentException::ArgumentException (std::string *s* = " ")

9.17.3 Member Function Documentation

9.17.3.1 std::string ArgumentException::py_what ( ) const

The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

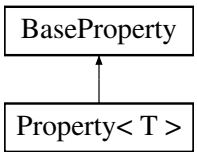
9.18 BaseProperty Class Reference

9.18.1 Description

Helper class to ensure that all Python property objects derive from the same base class.

`#include <PythonCdb.hh>`

Inheritance diagram for BaseProperty:



The documentation for this class was generated from the following file:

- core/PythonCdb.hh

9.19 ProgressMonitor::Block Class Reference

Public Member Functions

- Block ()

Public Attributes

- std::string name
- std::chrono::milliseconds started
- int step
- int total_steps

9.19.1 Constructor & Destructor Documentation

9.19.1.1 ProgressMonitor::Block::Block ( )

9.19.2 Member Data Documentation

9.19.2.1 std::string ProgressMonitor::Block::name

9.19.2.2 std::chrono::milliseconds ProgressMonitor::Block::started

9.19.2.3 int ProgressMonitor::Block::step

9.19.2.4 int ProgressMonitor::Block::total_steps

The documentation for this class was generated from the following files:

- core/ProgressMonitor.hh
- core/ProgressMonitor.cc

9.20 Server::Block Class Reference

Public Member Functions

- Block (websocketpp::connection_hdl, const std::string &, uint64_t id)

Public Attributes

- websocketpp::connection_hdl hdl
- std::string input
- std::string output
- std::string error
- uint64_t cell_id

9.20.1 Constructor & Destructor Documentation

9.20.1.1 Server::Block::Block (websocketpp::connection_hdl *h*, const std::string & *str*, uint64_t *id*)

9.20.2 Member Data Documentation

9.20.2.1 uint64_t Server::Block::cell_id

9.20.2.2 std::string Server::Block::error

9.20.2.3 websocketpp::connection_hdl Server::Block::hdl

9.20.2.4 std::string Server::Block::input

9.20.2.5 std::string Server::Block::output

The documentation for this class was generated from the following files:

- client_server/Server.hh
- client_server/Server.cc

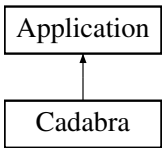
9.21 Cadabra Class Reference

9.21.1 Description

The Cadabra notebook application.

`#include <Cadabra.hh>`

Inheritance diagram for Cadabra:



Public Member Functions

- bool open_help (const std::string &filename, const std::string &title)

Static Public Member Functions

- static Glib::RefPtr< Cadabra > create (int, char **)

Protected Member Functions

- Cadabra (int, char **)
- virtual ~Cadabra ()
- void on_activate () override
- void on_open (const Gio::Application::type_vec_files &files, const Glib::ustring &hint) override

Private Attributes

- cadabra::ComputeThread compute
- std::thread compute_thread

9.21.2 Constructor & Destructor Documentation

9.21.2.1 Cadabra::Cadabra (int *argc,* char ** *argv*) [protected]

9.21.2.2 Cadabra::~Cadabra ( ) [protected], [virtual]

9.21.3 Member Function Documentation

9.21.3.1 Glib::RefPtr< Cadabra > Cadabra::create (int *argc,* char ** *argv*) [static]

9.21.3.2 void Cadabra::on_activate ( ) [override], [protected]

9.21.3.3 void Cadabra::on_open (const Gio::Application::type_vec_files & *files,* const Glib::ustring & *hint*) [override], [protected]

9.21.3.4 bool Cadabra::open_help (const std::string & *filename,* const std::string & *title*)

9.21.4 Member Data Documentation

9.21.4.1 cadabra::ComputeThread Cadabra::compute [private]

9.21.4.2 std::thread Cadabra::compute_thread [private]

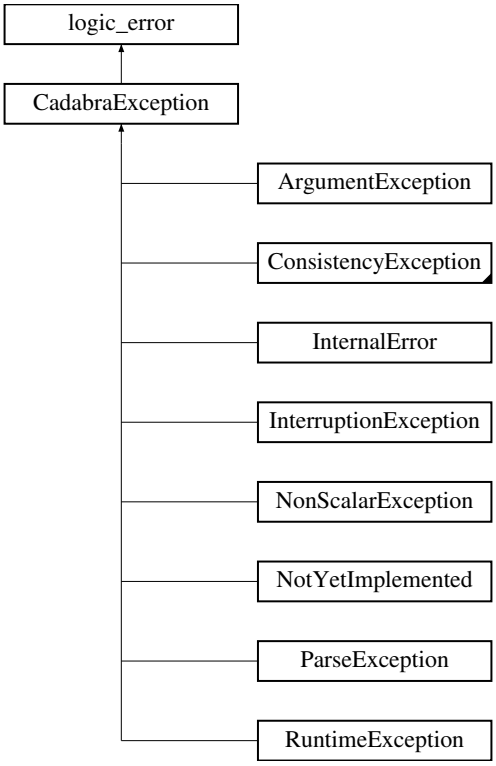
The documentation for this class was generated from the following files:

- frontend/gtkmm/Cadabra.hh
- frontend/gtkmm/Cadabra.cc

9.22 CadabraException Class Reference

`#include <Exceptions.hh>`

Inheritance diagram for CadabraException:



Public Member Functions

- CadabraException (std::string)

9.22.1 Constructor & Destructor Documentation

9.22.1.1 CadabraException::CadabraException (std::string *s*)

The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

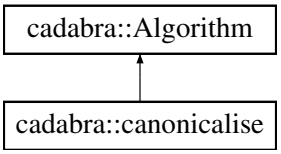
9.23 cadabra::canonicalise Class Reference

9.23.1 Description

Canonicalise the index structure of a tensorial expression.

`#include <canonicalise.hh>`

Inheritance diagram for cadabra::canonicalise:



Public Member Functions

- canonicalise (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Public Attributes

- std::vector< std::vector< int > > generating_set
- bool reuse_generating_set

Private Member Functions

- bool remove_traceless_traces (iterator &)
- bool remove_vanishing_numericals (iterator &)
- bool only_one_on_derivative (iterator index1, iterator index2) const
- std::string get_index_set_name (iterator) const
- Indices::position_t position_type (iterator) const

Additional Inherited Members

9.23.2 Constructor & Destructor Documentation

9.23.2.1 canonicalise::canonicalise (const Kernel & *k,* Ex & *tr*)

9.23.3 Member Function Documentation

9.23.3.1 Algorithm::result_t canonicalise::apply (iterator & *it*) [virtual]

Implements cadabra::Algorithm.

9.23.3.2 bool canonicalise::can_apply (iterator *it*) [virtual]

Implements cadabra::Algorithm.

9.23.3.3 std::string canonicalise::get_index_set_name (iterator *it*) const [private]

9.23.3.4 bool canonicalise::only_one_on_derivative (iterator *index1*, iterator *index2*) const [private]

9.23.3.5 Indices::position_t canonicalise::position_type (iterator *it*) const [private]

9.23.3.6 bool canonicalise::remove_traceless_traces (iterator & *it*) [private]

9.23.3.7 bool canonicalise::remove_vanishing_numericals (iterator & *it*) [private]

9.23.4 Member Data Documentation

9.23.4.1 std::vector<std::vector<int> > cadabra::canonicalise::generating_set

9.23.4.2 bool cadabra::canonicalise::reuse_generating_set

The documentation for this class was generated from the following files:

- core/algorithms/canonicalise.hh
- core/algorithms/canonicalise.cc

9.24 Server::CatchOutput Class Reference

9.24.1 Description

Python output catching.

We implement this in a C++ class because we want to be able to catch each line of output separately, and perhaps add additional functionality to send out-of-band messages to the client later.

```
#include <Server.hh>
```

Public Member Functions

- CatchOutput ()
- CatchOutput (const CatchOutput &)
- void write (const std::string &txt)
- void clear ()
- std::string str () const

Private Attributes

- std::string collect

9.24.2 Constructor & Destructor Documentation

9.24.2.1 Server::CatchOutput::CatchOutput ( )

9.24.2.2 Server::CatchOutput::CatchOutput (const CatchOutput &)

9.24.3 Member Function Documentation

9.24.3.1 void Server::CatchOutput::clear ( )

9.24.3.2 std::string Server::CatchOutput::str ( ) const

9.24.3.3 void Server::CatchOutput::write (const std::string & *txt*)

9.24.4 Member Data Documentation

9.24.4.1 std::string Server::CatchOutput::collect [private]

The documentation for this class was generated from the following files:

- client_server/Server.hh
- client_server/Server.cc

9.25 cadabra::CodeInput Class Reference

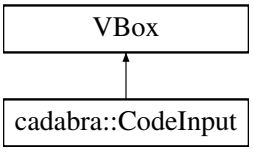
9.25.1 Description

A text cell editor widget with support for editing Python and LaTeX.

CodeInput is essentially a TextView with some additional i/o logic.

```
#include <CodeInput.hh>
```

Inheritance diagram for cadabra::CodeInput:



Classes

- class exp_input_tv

  The actual text widget used by CodeInput.

Public Member Functions

- CodeInput (DTree::iterator, Glib::RefPtr< Gtk::TextBuffer >, double scale, int font_step)

  Initialise with existing TextBuffer and a pointer to the Datacell corresponding to this CodeInput widget.
- CodeInput (DTree::iterator, const std::string &, double scale, int font_step)

  Initialise with a new TextBuffer (to be created by CodeInput), filling it with the content of the given string.
- void set_font_size (int num)

  Set the font size, 0 being default, negative smaller, positive larger.
- bool handle_button_press (GdkEventButton *)

  Handle mouse buttons.
- void handle_insert (const Gtk::TextIter &pos, const Glib::ustring &text, int bytes)

  Handle an insert event, which can consist of one or more inserted characters.
- void handle_erase (const Gtk::TextIter &start, const Gtk::TextIter &end)

  Handle an erase event.
- void update_buffer ()

  Ensure that the visual representation matches the DTree cell.
- void slice_cell (std::string &before, std::string &after)

  Return two strings corresponding to the text before and after the current cursor position.

Public Attributes

- Glib::RefPtr< Gtk::TextBuffer > buffer

  We cannot edit the content of the DataCell directly, because Gtk needs a Gtk::TextBuffer.
- exp_input_tv edit

Private Member Functions

- void init (int font_step)

9.25.2 Constructor & Destructor Documentation

9.25.2.1 CodeInput::CodeInput (DTree::iterator *it,* Glib::RefPtr< Gtk::TextBuffer > *tb,* double *scale,* int *font_step*)

Initialise with existing TextBuffer and a pointer to the Datacell corresponding to this CodeInput widget.

CodeInput is not allowed to modify this DataCell directly, but can read properties from it (e.g. in order to know when to display a 'busy' indicator). The scale parameter refers to hdpi scaling.

9.25.2.2 CodeInput::CodeInput (DTree::iterator *it,* const std::string & *txt,* double *scale,* int *font_step*)

Initialise with a new TextBuffer (to be created by CodeInput), filling it with the content of the given string.

9.25.3 Member Function Documentation

9.25.3.1 bool CodeInput::handle_button_press (GdkEventButton * *button*)

Handle mouse buttons.

9.25.3.2 void CodeInput::handle_erase (const Gtk::TextIter & *start,* const Gtk::TextIter & *end*)

Handle an erase event.

This function will just massage that data and then feed it through to the notebook window class by emitting a signal on content_erase (done like this to separate DTree modification from the widget).

9.25.3.3 void CodeInput::handle_insert (const Gtk::TextIter & *pos,* const Glib::ustring & *text,* int *bytes*)

Handle an insert event, which can consist of one or more inserted characters.

This function will just massage that data and then feed it through to the notebook window class by emitting a signal on content_insert (done like this to separate DTree modification from the widget).

9.25.3.4 void CodeInput::init (int *font_step*) [private]

9.25.3.5 void CodeInput::set_font_size (int *num*)

Set the font size, 0 being default, negative smaller, positive larger.

9.25.3.6 void CodeInput::slice_cell (std::string & *before,* std::string & *after*)

Return two strings corresponding to the text before and after the current cursor position.

9.25.3.7 void CodeInput::update_buffer ( )

Ensure that the visual representation matches the DTree cell.

9.25.4 Member Data Documentation

9.25.4.1 Glib::RefPtr<Gtk::TextBuffer> cadabra::CodeInput::buffer

We cannot edit the content of the DataCell directly, because Gtk needs a Gtk::TextBuffer.

However, the CodeInput widgets corresponding to a single DataCell all share their TextBuffer.

9.25.4.2 exp_input_tv cadabra::CodeInput::edit

The documentation for this class was generated from the following files:

- frontend/gtkmm/CodeInput.hh
- frontend/gtkmm/CodeInput.cc

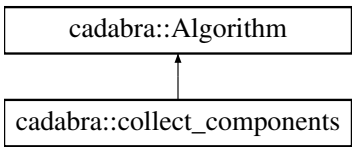
9.26 cadabra::collect_components Class Reference

9.26.1 Description

Collect terms inside a sum, merging their substitution rules.

`#include <collect_components.hh>`

Inheritance diagram for cadabra::collect_components:



Public Member Functions

- collect_components (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.26.2 Constructor & Destructor Documentation

9.26.2.1 collect_components::collect_components (const Kernel & *k*, Ex & *tr*)

9.26.3 Member Function Documentation

9.26.3.1 Algorithm::result_t collect_components::apply (iterator & *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.26.3.2 bool collect_components::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

The documentation for this class was generated from the following files:

- core/algorithms/collect_components.hh
- core/algorithms/collect_components.cc

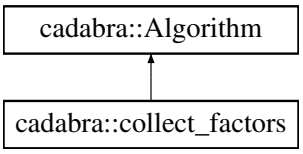
9.27 cadabra::collect_factors Class Reference

9.27.1 Description

Collect symbolically equal factors in a product.

`#include <collect_factors.hh>`

Inheritance diagram for cadabra::collect_factors:



Public Member Functions

- collect_factors (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Private Types

- typedef std::multimap< hashval_t, sibling_iterator > factor_hash_t
- typedef factor_hash_t::iterator factor_hash_iterator_t

Private Member Functions

- void fill_hash_map (iterator)

Private Attributes

- factor_hash_t factor_hash

Additional Inherited Members

9.27.2 Member Typedef Documentation

9.27.2.1 typedef factor_hash_t::iterator cadabra::collect_factors::factor_hash_iterator_t [private]

9.27.2.2 typedef std::multimap<hashval_t, sibling_iterator> cadabra::collect_factors::factor_hash_t [private]

9.27.3 Constructor & Destructor Documentation

9.27.3.1 collect_factors::collect_factors (const Kernel & *k*, Ex & *e*)

9.27.4 Member Function Documentation

9.27.4.1 Algorithm::result_t collect_factors::apply (iterator & *st*) [virtual]

Implements cadabra::Algorithm.

9.27.4.2 bool collect_factors::can_apply (iterator *it*) [virtual]

Implements cadabra::Algorithm.

9.27.4.3 void collect_factors::fill_hash_map (iterator *it*) [private]

9.27.5 Member Data Documentation

9.27.5.1 factor_hash_t cadabra::collect_factors::factor_hash [private]

The documentation for this class was generated from the following files:

- core/algorithms/collect_factors.hh
- core/algorithms/collect_factors.cc

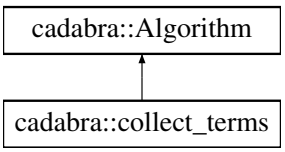
9.28 cadabra::collect_terms Class Reference

9.28.1 Description

Collect symbolically equal terms in a sum.

`#include <collect_terms.hh>`

Inheritance diagram for cadabra::collect_terms:



Public Member Functions

- collect_terms (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)
- void fill_hash_map (iterator)
- void fill_hash_map (sibling_iterator, sibling_iterator)

Private Types

- typedef std::multimap< hashval_t, sibling_iterator > term_hash_t
- typedef term_hash_t::iterator term_hash_iterator_t

Private Member Functions

- result_t collect_from_hash_map ()
- void remove_zeroed_terms (sibling_iterator, sibling_iterator)

Private Attributes

- term_hash_t term_hash

Additional Inherited Members

9.28.2 Member Typedef Documentation

9.28.2.1 typedef term_hash_t::iterator cadabra::collect_terms::term_hash_iterator_t [private]

9.28.2.2 typedef std::multimap<hashval_t, sibling_iterator> cadabra::collect_terms::term_hash_t [private]

9.28.3 Constructor & Destructor Documentation

9.28.3.1 collect_terms::collect_terms (const Kernel & *k*, Ex & *tr*)

9.28.4 Member Function Documentation

9.28.4.1 Algorithm::result_t collect_terms::apply (iterator & *st*) [virtual]

Implements cadabra::Algorithm.

9.28.4.2 bool collect_terms::can_apply (iterator *st*) [virtual]

Implements cadabra::Algorithm.

9.28.4.3 Algorithm::result_t collect_terms::collect_from_hash_map ( ) [private]

9.28.4.4 void collect_terms::fill_hash_map (iterator *it*)

9.28.4.5 void collect_terms::fill_hash_map (sibling_iterator *sib*, sibling_iterator *end*)

9.28.4.6 void collect_terms::remove_zeroed_terms (sibling_iterator *from*, sibling_iterator *to*) [private]

9.28.5 Member Data Documentation

9.28.5.1 term_hash_t cadabra::collect_terms::term_hash [private]

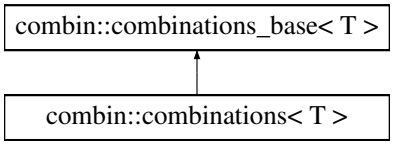
The documentation for this class was generated from the following files:

- core/algorithms/collect_terms.hh
- core/algorithms/collect_terms.cc

9.29 combin::combinations< T > Class Template Reference

`#include <Combinatorics.hh>`

Inheritance diagram for combin::combinations< T >:



Public Types

- typedef std::vector< std::vector< T > > permuted_sets_t
- typedef permuted_sets_t::const_iterator const_iterator

Public Member Functions

- combinations ()
- combinations (const std::vector< T > &)
- virtual ~combinations ()
- virtual void clear ()
- virtual void clear_results ()
- const std::vector< T > & operator[ ] (unsigned int) const
- int ordersign (unsigned int) const
- unsigned int size () const
- unsigned int multiplier (unsigned int) const

Protected Member Functions

- virtual void vector_generated (const std::vector< unsigned int > &)

Private Attributes

- permuted_sets_t storage

Additional Inherited Members

9.29.1 Member Typedef Documentation

9.29.1.1 template<class T> typedef permuted_sets_t::const_iterator combin::combinations< T >::const_iterator

9.29.1.2 template<class T> typedef std::vector<std::vector<T> > combin::combinations< T >::permuted_sets_t

9.29.2 Constructor & Destructor Documentation

9.29.2.1 template<class T > combin::combinations< T >::combinations ( )

9.29.2.2 template<class T> combin::combinations< T >::combinations (const std::vector< T > & *oa*)

9.29.2.3 template<class T > combin::combinations< T >::~combinations ( ) [virtual]

9.29.3 Member Function Documentation

9.29.3.1 template<class T > void combin::combinations< T >::clear ( ) [virtual]

Reimplemented from combin::combinations_base< T >.

9.29.3.2 template<class T > void combin::combinations< T >::clear_results ( ) [virtual]

Reimplemented from combin::combinations_base< T >.

9.29.3.3 template<class T > unsigned int combin::combinations< T >::multiplier (unsigned int *num*) const

9.29.3.4 template<class T > const std::vector< T > & combin::combinations< T >::operator[ ] (unsigned int *i*) const

9.29.3.5 template<class T > int combin::combinations< T >::ordersign (unsigned int *num*) const

9.29.3.6 template<class T > unsigned int combin::combinations< T >::size ( ) const

9.29.3.7 template<class T > void combin::combinations< T >::vector_generated (const std::vector< unsigned int > & *toadd*) [protected], [virtual]

Implements combin::combinations_base< T >.

9.29.4 Member Data Documentation

9.29.4.1 template<class T> permuted_sets_t combin::combinations< T >::storage [private]

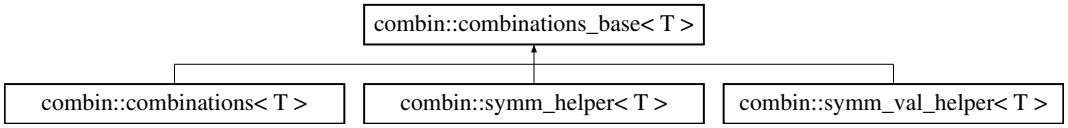
The documentation for this class was generated from the following file:

- core/Combinatorics.hh

9.30 combin::combinations_base< T > Class Template Reference

`#include <Combinatorics.hh>`

Inheritance diagram for combin::combinations_base< T >:



Public Types

- enum weight_cond { weight_equals, weight_less, weight_greater }

Public Member Functions

- combinations_base ()
- combinations_base (const std::vector< T > &)
- virtual ~combinations_base ()
- void permute (long start=-1, long end=-1)
- virtual void clear ()
- virtual void clear_results ()
- unsigned int sum_of_sublengths () const
- void set_unit_sublengths ()
- unsigned int multiplier (const std::vector< T > &) const
- unsigned int total_permutations () const

Public Attributes

- unsigned int block_length
- std::vector< unsigned int > sublengths
- range_vector_t input_asym
- std::vector< T > original
- bool multiple_pick
- std::vector< weights_t > weights
- std::vector< int > max_weights
- std::vector< weight_cond > weight_conditions
- unsigned int sub_problem_blocksize

Protected Member Functions

- virtual void vector_generated (const std::vector< unsigned int > &)=0
- virtual bool entry_accepted (unsigned int current) const

Protected Attributes

- std::vector< unsigned int > temparr
- long start_
- long end_
- long vector_generated_called_
- std::vector< int > current_weight

Private Member Functions

- bool is_allowed_by_weight_constraints (unsigned int i)
- bool final_weight_constraints_check () const
- void update_weights (unsigned int i)
- void restore_weights (unsigned int i)
- void nextstep (unsigned int current, unsigned int fromalgehad, unsigned int groupindex, std::vector< bool > algehad)

9.30.1 Member Enumeration Documentation

9.30.1.1 template<class T> enum combin::combinations_base::weight_cond

Enumerator

weight_equals

weight_less

weight_greater

9.30.2 Constructor & Destructor Documentation

9.30.2.1 template<class T > combin::combinations_base< T >::combinations_base ( )

9.30.2.2 template<class T> combin::combinations_base< T >::combinations_base (const std::vector< T > & *oa*)

9.30.2.3 template<class T > combin::combinations_base< T >::~combinations_base ( ) [virtual]

9.30.3 Member Function Documentation

9.30.3.1 template<class T > void combin::combinations_base< T >::clear ( ) [virtual]

Reimplemented in combin::symm_val_helper< T >, combin::symm_val_helper< unsigned int >, combin::symm_helper< T >, combin::symm_helper< unsigned int >, combin::combinations< T >, and combin::combinations< unsigned int >.

9.30.3.2 template<class T > void combin::combinations_base< T >::clear_results ( ) [virtual]

Reimplemented in combin::combinations< T >, and combin::combinations< unsigned int >.

9.30.3.3 template<class T > bool combin::combinations_base< T >::entry_accepted (unsigned int *current*) const [protected], [virtual]

9.30.3.4 template<class T > bool combin::combinations_base< T >::final_weight_constraints_check ( ) const [private]

9.30.3.5 template<class T > bool combin::combinations_base< T >::is_allowed_by_weight_constraints (unsigned int *i*) [private]

9.30.3.6 template<class T> unsigned int combin::combinations_base< T >::multiplier (const std::vector< T > & *stor*) const

9.30.3.7 template<class T > void combin::combinations_base< T >::nextstep (unsigned int *current,* unsigned int *fromalgehad,* unsigned int *groupindex,* std::vector< bool > *algehad*) [private]

9.30.3.8 template<class T > void combin::combinations_base< T >::permute (long *start* = -1, long *end* = -1)

9.30.3.9 template<class T > void combin::combinations_base< T >::restore_weights (unsigned int *i*) [private]

9.30.3.10 template<class T > void combin::combinations_base< T >::set_unit_sublengths ( )

9.30.3.11 template<class T > unsigned int combin::combinations_base< T >::sum_of_sublengths ( ) const

9.30.3.12 template<class T > unsigned int combin::combinations_base< T >::total_permutations ( ) const

9.30.3.13 template<class T > void combin::combinations_base< T >::update_weights (unsigned int *i*) [private]

9.30.3.14 template<class T> virtual void combin::combinations_base< T >::vector_generated (const std::vector< unsigned int > &) [protected], [pure virtual]

Implemented in combin::symm_val_helper< T >, combin::symm_val_helper< unsigned int >, combin::symm_helper< T >, combin::symm_helper< unsigned int >, combin::combinations< T >, and combin::combinations< unsigned int >.

9.30.4 Member Data Documentation

9.30.4.1 template<class T> unsigned int combin::combinations_base< T >::block_length

9.30.4.2 template<class T> std::vector<int> combin::combinations_base< T >::current_weight [protected]

9.30.4.3 template<class T> long combin::combinations_base< T >::end_ [protected]

9.30.4.4 template<class T> range_vector_t combin::combinations_base< T >::input_asym

9.30.4.5 template<class T> std::vector<int> combin::combinations_base< T >::max_weights

9.30.4.6 template<class T> bool combin::combinations_base< T >::multiple_pick

9.30.4.7 template<class T> std::vector<T> combin::combinations_base< T >::original

9.30.4.8 template<class T> long combin::combinations_base< T >::start_ [protected]

9.30.4.9 template<class T> unsigned int combin::combinations_base< T >::sub_problem_blocksize

9.30.4.10 template<class T> std::vector<unsigned int> combin::combinations_base< T >::sublengths

9.30.4.11 template<class T> std::vector<unsigned int> combin::combinations_base< T >::temparr [protected]

9.30.4.12 template<class T> long combin::combinations_base< T >::vector_generated_called_ [protected]

9.30.4.13 template<class T> std::vector<weight_cond> combin::combinations_base< T >::weight_conditions

9.30.4.14 template<class T> std::vector<weights_t> combin::combinations_base< T >::weights

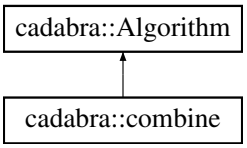
The documentation for this class was generated from the following file:

- core/Combinatorics.hh

9.31 cadabra::combine Class Reference

`#include <combine.hh>`

Inheritance diagram for cadabra::combine:



Public Member Functions

- combine (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Types

- typedef std::map< nset_t::iterator, iterator > indexlocmap_t

Private Attributes

- indexlocmap_t iloc

Additional Inherited Members

9.31.1 Member Typedef Documentation

9.31.1.1 typedef std::map<nset_t::iterator, iterator> cadabra::combine::indexlocmap_t [private]

9.31.2 Constructor & Destructor Documentation

9.31.2.1 combine::combine (const Kernel & *k*, Ex & *e*)

9.31.3 Member Function Documentation

9.31.3.1 Algorithm::result_t combine::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.31.3.2 bool combine::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.31.4 Member Data Documentation

9.31.4.1 indexlocmap_t cadabra::combine::iloc [private]

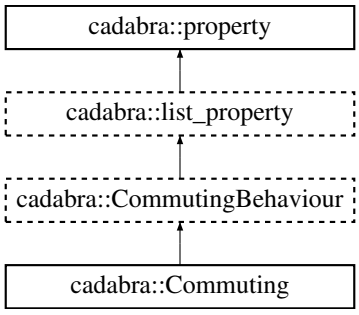
The documentation for this class was generated from the following files:

- core/algorithms/combine.hh
- core/algorithms/combine.cc

9.32 cadabra::Commuting Class Reference

`#include <Commuting.hh>`

Inheritance diagram for cadabra::Commuting:



Public Member Functions

- virtual std::string name () const
- virtual int sign () const

Additional Inherited Members

9.32.1 Member Function Documentation

9.32.1.1 std::string Commuting::name ( ) const [virtual]

Implements cadabra::property.

9.32.1.2 virtual int cadabra::Commuting::sign ( ) const [inline], [virtual]

Implements cadabra::CommutingBehaviour.

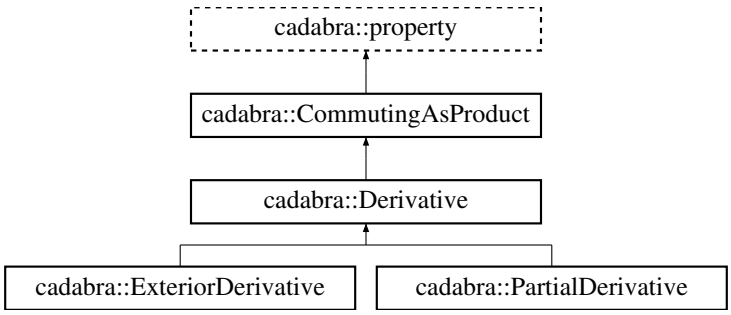
The documentation for this class was generated from the following files:

- core/properties/Commuting.hh
- core/properties/Commuting.cc

9.33 cadabra::CommutingAsProduct Class Reference

`#include <CommutingAsProduct.hh>`

Inheritance diagram for cadabra::CommutingAsProduct:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.33.1 Member Function Documentation

9.33.1.1 std::string CommutingAsProduct::name ( ) const [virtual]

Implements cadabra::property.

Reimplemented in cadabra::Derivative, cadabra::ExteriorDerivative, and cadabra::PartialDerivative.

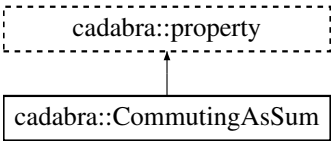
The documentation for this class was generated from the following files:

- core/properties/CommutingAsProduct.hh
- core/properties/CommutingAsProduct.cc

9.34 cadabra::CommutingAsSum Class Reference

`#include <CommutingAsSum.hh>`

Inheritance diagram for cadabra::CommutingAsSum:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.34.1 Member Function Documentation

9.34.1.1 std::string CommutingAsSum::name ( ) const [virtual]

Implements cadabra::property.

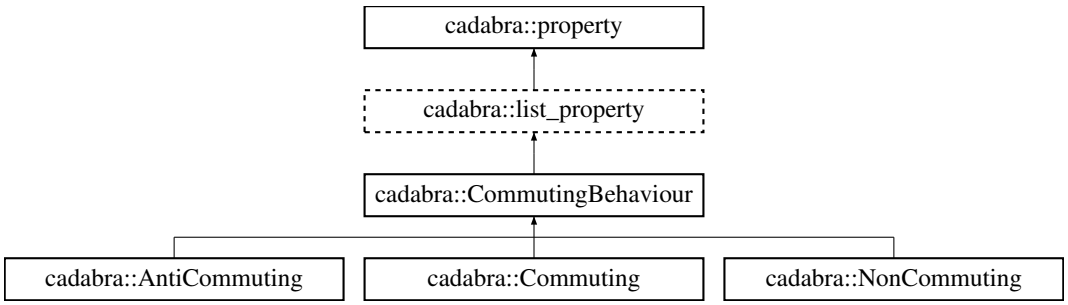
The documentation for this class was generated from the following files:

- core/properties/CommutingAsSum.hh
- core/properties/CommutingAsSum.cc

9.35 cadabra::CommutingBehaviour Class Reference

`#include <CommutingBehaviour.hh>`

Inheritance diagram for cadabra::CommutingBehaviour:



Public Member Functions

- virtual int sign () const =0
- virtual match_t equals (const property ∗) const

Additional Inherited Members

9.35.1 Member Function Documentation

9.35.1.1 property::match_t CommutingBehaviour::equals (const property ∗) const [virtual]

Reimplemented from cadabra::property.

9.35.1.2 virtual int cadabra::CommutingBehaviour::sign ( ) const [pure virtual]

Implemented in cadabra::AntiCommuting, cadabra::Commuting, and cadabra::NonCommuting.

The documentation for this class was generated from the following files:

- core/properties/CommutingBehaviour.hh
- core/properties/CommutingBehaviour.cc

9.36 cadabra::complete Class Reference

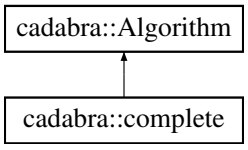
9.36.1 Description

Complete a set of coordinate rules so that they also cover related tensors.

At present this only inverts metric rules, but could do more related rules, or expanded to cover symmetry.

`#include <complete.hh>`

Inheritance diagram for cadabra::complete:



Public Member Functions

- complete (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Attributes

- Ex goal

Additional Inherited Members

9.36.2 Constructor & Destructor Documentation

9.36.2.1 complete::complete (const Kernel & *k*, Ex & *e*, Ex & *g*)

9.36.3 Member Function Documentation

9.36.3.1 Algorithm::result_t complete::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.36.3.2 bool complete::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.36.4 Member Data Documentation

9.36.4.1 Ex cadabra::complete::goal [private]

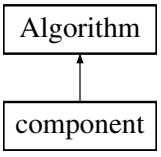
The documentation for this class was generated from the following files:

- core/algorithms/complete.hh
- core/algorithms/complete.cc

9.37 component Class Reference

`#include <component.hh>`

Inheritance diagram for component:



Public Member Functions

- component (const Kernel &, Ex &ex, Ex &rules)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

9.37.1 Constructor & Destructor Documentation

9.37.1.1 component::component (const Kernel & , Ex & *ex*, Ex & *rules*)

9.37.2 Member Function Documentation

9.37.2.1 virtual result_t component::apply (iterator &) [virtual]

9.37.2.2 virtual bool component::can_apply (iterator) [virtual]

The documentation for this class was generated from the following file:

- core/algorithms/component.hh

9.38 cadabra::ComputeThread Class Reference

9.38.1 Description

Base class which talks to the server and sends Action objects back to the DocumentThread.

ComputeThread is the base class which takes care of doing actual computations with the cells in a document. It handles talking to the server backend. It knows how to pass cells to the server and ask them to be executed. Results are reported back to the GUI by putting ActionBase objects onto its todo stack. ComputeThread never directly modifies the document tree.

```
#include <ComputeThread.hh>
```

Public Member Functions

- ComputeThread ()

  If the ComputeThread is constructed with a null pointer to the gui, there will be no gui updates, just DTree updates.
- ComputeThread (const ComputeThread &)=delete
- ~ComputeThread ()
- void set_master (GUIBase *, DocumentThread *)

  Determine the objects that this compute thread should be talking to.
- void run ()

  Main entry point, which will connect to the server and then start an event loop to handle communication with the server.
- void execute_cell (DTree::iterator)

  In order to execute code on the server, call the following from the GUI thread.
- void stop ()

  Stop the current cell execution on the server and remove all other cells from the run queue as well.
- void restart_kernel ()

  Restart the kernel.
- int number_of_cells_executing (void) const
- void terminate ()

  Terminate the compute thread, in preparation for shutting down the client altogether.

Private Member Functions

- void init ()
- void try_connect ()
- void try_spawn_server ()
- void on_open (websocketpp::connection_hdl hdl)
- void on_fail (websocketpp::connection_hdl hdl)
- void on_close (websocketpp::connection_hdl hdl)
- void on_message (websocketpp::connection_hdl hdl, message_ptr msg)
- DTree::iterator find_cell_by_id (DataCell::id_t, bool remove)
- void all_cells_nonrunning ()

  Set all cells to be non-running (e.g.

Private Attributes

- GUIBase * gui
- DocumentThread * docthread
- std::thread::id gui_thread_id
- std::map< DataCell::id_t, DTree::iterator > running_cells
- WSClient wsclient
- bool connection_is_open
- bool restarting_kernel
- WSClient::connection_ptr connection
- websocketpp::connection_hdl our_connection_hdl
- Glib::Pid server_pid
- int server_stdout
- int server_stderr
- unsigned short port

9.38.2 Constructor & Destructor Documentation

9.38.2.1 ComputeThread::ComputeThread ( )

If the ComputeThread is constructed with a null pointer to the gui, there will be no gui updates, just DTree updates.

9.38.2.2 cadabra::ComputeThread::ComputeThread (const ComputeThread &) [delete]

9.38.2.3 ComputeThread::~ComputeThread ( )

9.38.3 Member Function Documentation

9.38.3.1 void ComputeThread::all_cells_nonrunning ( ) [private]

Set all cells to be non-running (e.g.

after a kernel failure) and report the status of each cell to the GUI.

9.38.3.2 void ComputeThread::execute_cell (DTree::iterator *it*)

In order to execute code on the server, call the following from the GUI thread.

This method returns as soon as the request has been put on the network queue. If no communication with the server is necessary, this returns immediately. The ComputeThread will report the result of the computation/processing by adding actions to the DocumentThread owned pending_actions stack, by calling queue_action. It will never modify the cell directly, and will also never modify any other cells in the document tree.

9.38.3.3 DTree::iterator ComputeThread::find_cell_by_id (DataCell::id_t *id,* bool *remove*) [private]

9.38.3.4 void ComputeThread::init ( ) [private]

9.38.3.5 int ComputeThread::number_of_cells_executing (void) const

9.38.3.6 void ComputeThread::on_close (websocketpp::connection_hdl *hdl*) [private]

9.38.3.7 void ComputeThread::on_fail (websocketpp::connection_hdl *hdl*) [private]

9.38.3.8 void ComputeThread::on_message (websocketpp::connection_hdl *hdl,* message_ptr *msg*) [private]

9.38.3.9 void ComputeThread::on_open (websocketpp::connection_hdl *hdl*) [private]

9.38.3.10 void ComputeThread::restart_kernel ( )

Restart the kernel.

9.38.3.11 void ComputeThread::run ( )

Main entry point, which will connect to the server and then start an event loop to handle communication with the server.

Only terminates when the connection drops, so run your GUI on a different thread.

9.38.3.12 void ComputeThread::set_master (GUIBase * *b,* DocumentThread * *d*)

Determine the objects that this compute thread should be talking to.

9.38.3.13 void ComputeThread::stop ( )

Stop the current cell execution on the server and remove all other cells from the run queue as well.

9.38.3.14 void ComputeThread::terminate ( )

Terminate the compute thread, in preparation for shutting down the client altogether.

9.38.3.15 void ComputeThread::try_connect ( ) [private]

9.38.3.16 void ComputeThread::try_spawn_server ( ) [private]

9.38.4 Member Data Documentation

9.38.4.1 WSClient::connection_ptr cadabra::ComputeThread::connection [private]

9.38.4.2 bool cadabra::ComputeThread::connection_is_open [private]

9.38.4.3 DocumentThread* cadabra::ComputeThread::docthread [private]

9.38.4.4 GUIBase* cadabra::ComputeThread::gui [private]

9.38.4.5 std::thread::id cadabra::ComputeThread::gui_thread_id [private]

9.38.4.6 websocketpp::connection_hdl cadabra::ComputeThread::our_connection_hdl [private]

9.38.4.7 unsigned short cadabra::ComputeThread::port [private]

9.38.4.8 bool cadabra::ComputeThread::restarting_kernel [private]

9.38.4.9 std::map<DataCell::id_t, DTree::iterator> cadabra::ComputeThread::running_cells [private]

9.38.4.10 Glib::Pid cadabra::ComputeThread::server_pid [private]

9.38.4.11 int cadabra::ComputeThread::server_stderr [private]

9.38.4.12 int cadabra::ComputeThread::server_stdout [private]

9.38.4.13 WSClient cadabra::ComputeThread::wsclient [private]

The documentation for this class was generated from the following files:

- client_server/ComputeThread.hh
- client_server/ComputeThread.cc

9.39 Server::Connection Class Reference

Public Member Functions

- Connection ()

Public Attributes

- websocketpp::connection_hdl hdl
- boost::uuids::uuid uuid

9.39.1 Constructor & Destructor Documentation

9.39.1.1 Server::Connection::Connection ( )

9.39.2 Member Data Documentation

9.39.2.1 websocketpp::connection_hdl Server::Connection::hdl

9.39.2.2 boost::uuids::uuid Server::Connection::uuid

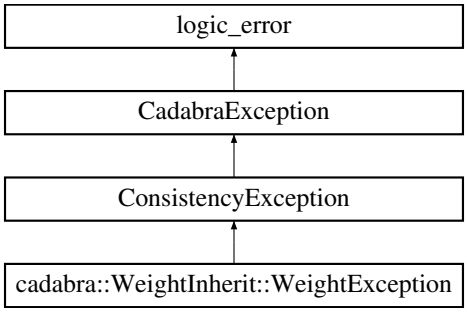
The documentation for this class was generated from the following files:

- client_server/Server.hh
- client_server/Server.cc

9.40 ConsistencyException Class Reference

`#include <Exceptions.hh>`

Inheritance diagram for ConsistencyException:



Public Member Functions

- ConsistencyException (std::string)

9.40.1 Constructor & Destructor Documentation

9.40.1.1 ConsistencyException::ConsistencyException (std::string *s*)

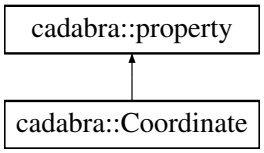
The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

9.41 cadabra::Coordinate Class Reference

`#include <Coordinate.hh>`

Inheritance diagram for cadabra::Coordinate:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.41.1 Member Function Documentation

9.41.1.1 std::string Coordinate::name ( ) const [virtual]

Implements cadabra::property.

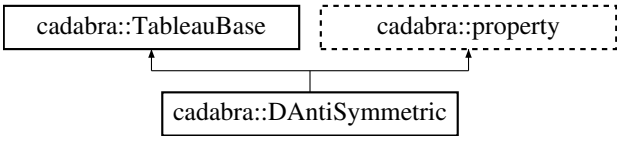
The documentation for this class was generated from the following files:

- core/properties/Coordinate.hh
- core/properties/Coordinate.cc

9.42 cadabra::DAntiSymmetric Class Reference

`#include <DAntiSymmetric.hh>`

Inheritance diagram for cadabra::DAntiSymmetric:



Public Member Functions

- virtual ~DAntiSymmetric ()
- virtual std::string name () const override
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const override
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const override

Additional Inherited Members

9.42.1 Constructor & Destructor Documentation

9.42.1.1 virtual cadabra::DAntiSymmetric::~DAntiSymmetric ( ) [inline], [virtual]

9.42.2 Member Function Documentation

9.42.2.1 TableauBase::tab_t DAntiSymmetric::get_tab (const Properties & *properties,* Ex & *tr,* Ex::iterator *it,* unsigned int *num*) const [override], [virtual]

Implements cadabra::TableauBase.

9.42.2.2 std::string DAntiSymmetric::name ( ) const [override], [virtual]

Implements cadabra::property.

9.42.2.3 unsigned int DAntiSymmetric::size (const Properties & , Ex & , Ex::iterator) const [override], [virtual]

Implements cadabra::TableauBase.

The documentation for this class was generated from the following files:

- core/properties/DAntiSymmetric.hh
- core/properties/DAntiSymmetric.cc

9.43 cadabra::DataCell Class Reference

9.43.1 Description

DataCells are the basic building blocks for a document.

They represent visual units in the notebook interface. They are stored in a tree inside the client, and can be transmitted over the wire between server and client in JSON format (see the documentation of the cadabra::JSON_serialise and cadabra::JSON_deserialise methods below for details on this representation). The notebook user interface reads these cells and construct corresponding graphical output for them.

The cadabra.display method of the cadabra python library knows how to turn various Python objects into the corresponding JSON representation of a DataCell.

```
#include <DataCell.hh>
```

Classes

- class id_t

  Each cell is identified by a serial number 'id' which is used to keep track of it across network calls, and a bool indicating whether the client or the server has created this cell.

Public Types

- enum CellType {
  CellType::document, CellType::python, CellType::latex, CellType::output,
  CellType::verbatim, CellType::latex_view, CellType::image_png, CellType::error }

  Cells are labelled with the data type of its contents, which is stored in a textural representation but may need processing for proper display.

Public Member Functions

- DataCell (CellType t=CellType::python, const std::string &str="", bool hidden=false)

  Standard constructor, generates a new unique id for this DataCell.
- DataCell (id_t, CellType t=CellType::python, const std::string &str="", bool hidden=false)

  Initialise a cell with an already determined id (it is the caller's responsibility to ensure that this id does not clash with any other DataCell's id).
- DataCell (const DataCell &)

  Copy constructor; preserves all information including id.
- id_t id () const

Public Attributes

- CellType cell_type
- std::string textbuf

  Textual representation of the cell content.
- bool hidden

  Flag indicating whether this cell should be hidden from view.
- bool sensitive
- bool running

  Indicator whether this cell is currently being evaluated by the server.

Private Attributes

- id_t serial_number

Static Private Attributes

- static std::mutex serial_mutex
- static uint64_t max_serial_number =0

9.43.2 Member Enumeration Documentation

9.43.2.1 enum cadabra::DataCell::CellType [strong]

Cells are labelled with the data type of its contents, which is stored in a textural representation but may need processing for proper display.

Enumerator

document head node, only one per document

python input : editor cell for code in python

latex input : editor cell for code in latex

output output: cell showing python stdout, verbatim

verbatim output: cell showing other verbatim output

latex_view output: cell showing LaTeX text formatted using a LaTeX

image_png output: cell showing a base64 encoded PNG image

error output: cell showing LaTeX text for errors

9.43.3 Constructor & Destructor Documentation

9.43.3.1 DataCell::DataCell (CellType *t* = CellType::python, const std::string & *str* = " ", bool *hidden* = `false`)

Standard constructor, generates a new unique id for this DataCell.

9.43.3.2 DataCell::DataCell (id_t *id_*, CellType *t* = CellType::python, const std::string & *str* = " ", bool *hidden* = `false`)

Initialise a cell with an already determined id (it is the caller's responsibility to ensure that this id does not clash with any other DataCell's id).

9.43.3.3 DataCell::DataCell (const DataCell & *other*)

Copy constructor; preserves all information including id.

9.43.4 Member Function Documentation

9.43.4.1 DataCell::id_t DataCell::id ( ) const

9.43.5 Member Data Documentation

9.43.5.1 CellType cadabra::DataCell::cell_type

9.43.5.2 bool cadabra::DataCell::hidden

Flag indicating whether this cell should be hidden from view.

The GUI should have a way to bring the cells back into view, typically by clicking on the output cell corresponding to the input cell.

9.43.5.3 uint64_t DataCell::max_serial_number =0 `[static]`, `[private]`

9.43.5.4 bool cadabra::DataCell::running

Indicator whether this cell is currently being evaluated by the server.

Currently only has a meaning for cells of type 'python'. This flag is set/reset using the ActionSetRunStatus action.

9.43.5.5 bool cadabra::DataCell::sensitive

9.43.5.6 std::mutex DataCell::serial_mutex [static], [private]

9.43.5.7 id_t cadabra::DataCell::serial_number [private]

9.43.5.8 std::string cadabra::DataCell::textbuf

Textual representation of the cell content.

For e.g. latex cells it is a bit of a waste to store this representation both in the input and in the output cell. However, this gives us the flexibility to do manipulations on the input (e.g. resolving equation references) before feeding it to LaTeX.

The documentation for this class was generated from the following files:

- client_server/DataCell.hh
- client_server/DataCell.cc

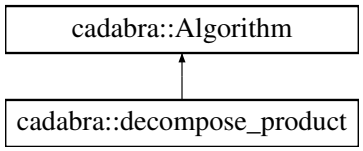
9.44 cadabra::decompose_product Class Reference

9.44.1 Description

Decompose a product of tensors by applying Young projectors.

`#include <decompose_product.hh>`

Inheritance diagram for cadabra::decompose_product:



Public Member Functions

- decompose_product (const Kernel &, Ex &tr)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Types

- typedef young_project::name_tab_t sibtab_t
- typedef yngtab::tableaux< sibtab_t > sibtabs_t
- typedef young_project::pos_tab_t numtab_t
- typedef yngtab::tableaux< numtab_t > numtabs_t

Private Member Functions

- const Indices * indices_equivalent (iterator it) const

  Test that all indices on the product are equivalent, that is, have the same Indices property attached to them.
- void fill_asym_ranges (TableauBase::tab_t &tab, int offset, combin::range_vector_t &)
- void project_onto_initial_symmetries (Ex &rep, iterator rr, young_project &yp, const TableauBase *tt, iterator ff, int offset, const TableauBase::tab_t &thetab, bool remove_traces)

Private Attributes

- iterator f1
- iterator f2
- const TableauBase * t1
- const TableauBase * t2
- TableauBase::tab_t t1tab
- TableauBase::tab_t t2tab
- const Indices * ind1
- const Indices * ind2
- unsigned int dim
- yngtab::filled_tableau< iterator > nt1
- yngtab::filled_tableau< iterator > nt2
- combin::range_vector_t asym_ranges

Additional Inherited Members

9.44.2 Member Typedef Documentation

9.44.2.1 typedef young_project::pos_tab_t cadabra::decompose_product::numtab_t [private]

9.44.2.2 typedef yngtab::tableaux<numtab_t> cadabra::decompose_product::numtabs_t [private]

9.44.2.3 typedef young_project::name_tab_t cadabra::decompose_product::sibtab_t [private]

9.44.2.4 typedef yngtab::tableaux<sibtab_t> cadabra::decompose_product::sibtabs_t [private]

9.44.3 Constructor & Destructor Documentation

9.44.3.1 decompose_product::decompose_product (const Kernel & *k*, Ex & *tr*)

9.44.4 Member Function Documentation

9.44.4.1 Algorithm::result_t decompose_product::apply (iterator & *it*) [override], [virtual]

++ntt; ++ntt; ++ntt;

break;

Implements cadabra::Algorithm.

9.44.4.2 bool decompose_product::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.44.4.3 void decompose_product::fill_asym_ranges (TableauBase::tab_t & *tab,* int *offset,* combin::range_vector_t & *ranges*) [private]

9.44.4.4 const Indices * decompose_product::indices_equivalent (iterator *it*) const [private]

Test that all indices on the product are equivalent, that is, have the same Indices property attached to them.

Return this property.

9.44.4.5 void decompose_product::project_onto_initial_symmetries (Ex & *rep,* iterator *rr,* young_project & *yp,* const TableauBase * *tt,* iterator *ff,* int *offset,* const TableauBase::tab_t & *thetab,* bool *remove_traces*) [private]

9.44.5 Member Data Documentation

9.44.5.1 combin::range_vector_t cadabra::decompose_product::asym_ranges [private]

9.44.5.2 unsigned int cadabra::decompose_product::dim [private]

9.44.5.3 iterator cadabra::decompose_product::f1 [private]

9.44.5.4 iterator cadabra::decompose_product::f2 [private]

9.44.5.5 const Indices* cadabra::decompose_product::ind1 [private]

9.44.5.6 const Indices * cadabra::decompose_product::ind2 [private]

9.44.5.7 yngtab::filled_tableau<iterator> cadabra::decompose_product::nt1 [private]

9.44.5.8 yngtab::filled_tableau<iterator> cadabra::decompose_product::nt2 [private]

9.44.5.9 const TableauBase* cadabra::decompose_product::t1 [private]

9.44.5.10 TableauBase::tab_t cadabra::decompose_product::t1tab [private]

9.44.5.11 const TableauBase * cadabra::decompose_product::t2 [private]

9.44.5.12 TableauBase::tab_t cadabra::decompose_product::t2tab [private]

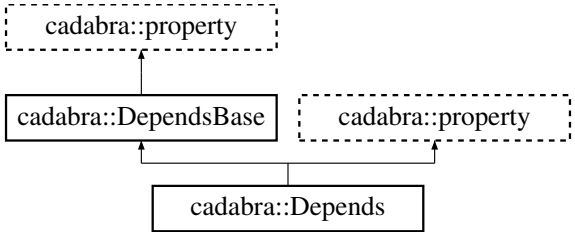
The documentation for this class was generated from the following files:

- core/algorithms/decompose_product.hh
- core/algorithms/decompose_product.cc

9.45 cadabra::Depends Class Reference

`#include <Depends.hh>`

Inheritance diagram for cadabra::Depends:



Public Member Functions

- virtual std::string name () const override
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual Ex dependencies (const Kernel &, Ex::iterator) const override

  Returns a tree of objects on which the given object depends.
- virtual std::string unnamed_argument () const override

Private Attributes

- Ex dependencies_

Additional Inherited Members

9.45.1 Member Function Documentation

9.45.1.1 Ex Depends::dependencies (const Kernel & , Ex::iterator) const [override], [virtual]

Returns a tree of objects on which the given object depends.

Implements cadabra::DependsBase.

9.45.1.2 std::string Depends::name ( ) const [override], [virtual]

Implements cadabra::property.

9.45.1.3 bool Depends::parse (const Kernel & *kernel,* keyval_t & *kv*) [override], [virtual]

Reimplemented from cadabra::property.

9.45.1.4 virtual std::string cadabra::Depends::unnamed_argument ( ) const [inline],[override],[virtual]

Reimplemented from cadabra::property.

9.45.2 Member Data Documentation

9.45.2.1 Ex cadabra::Depends::dependencies_ [private]

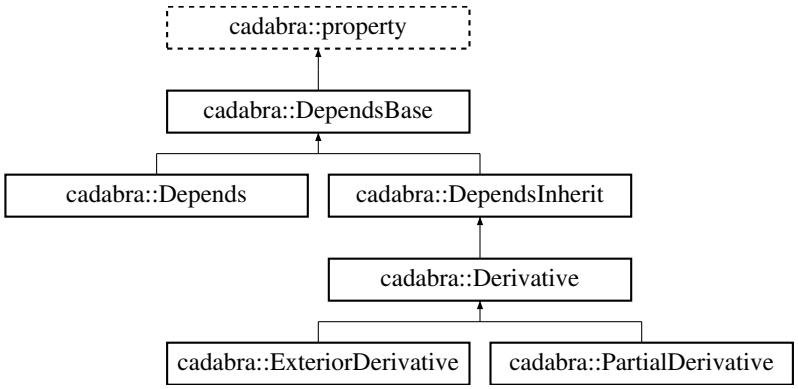
The documentation for this class was generated from the following files:

- core/properties/Depends.hh
- core/properties/Depends.cc

9.46 cadabra::DependsBase Class Reference

`#include <DependsBase.hh>`

Inheritance diagram for cadabra::DependsBase:



Public Member Functions

- virtual Ex dependencies (const Kernel &, Ex::iterator) const =0

  Returns a tree of objects on which the given object depends.

Additional Inherited Members

9.46.1 Member Function Documentation

9.46.1.1 virtual Ex cadabra::DependsBase::dependencies (const Kernel & , Ex::iterator) const [pure virtual]

Returns a tree of objects on which the given object depends.

Implemented in cadabra::DependsInherit, and cadabra::Depends.

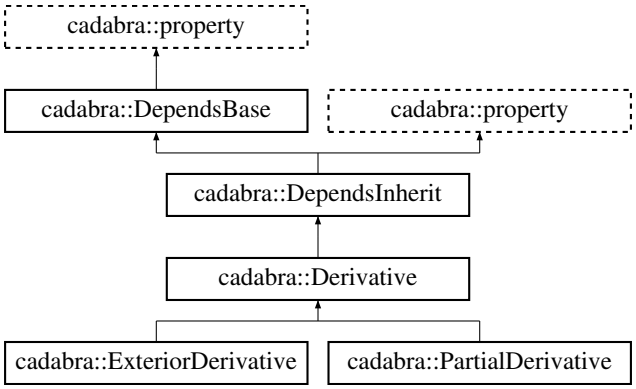
The documentation for this class was generated from the following file:

- core/properties/DependsBase.hh

9.47 cadabra::DependsInherit Class Reference

`#include <DependsInherit.hh>`

Inheritance diagram for cadabra::DependsInherit:



Public Member Functions

- virtual std::string name () const override
- virtual Ex dependencies (const Kernel &, Ex::iterator) const override

  Returns a tree of objects on which the given object depends.

Additional Inherited Members

9.47.1 Member Function Documentation

9.47.1.1 Ex DependsInherit::dependencies (const Kernel & , Ex::iterator) const [override], [virtual]

Returns a tree of objects on which the given object depends.

Implements cadabra::DependsBase.

9.47.1.2 std::string DependsInherit::name ( ) const [override], [virtual]

Implements cadabra::property.

Reimplemented in cadabra::Derivative, cadabra::ExteriorDerivative, and cadabra::PartialDerivative.

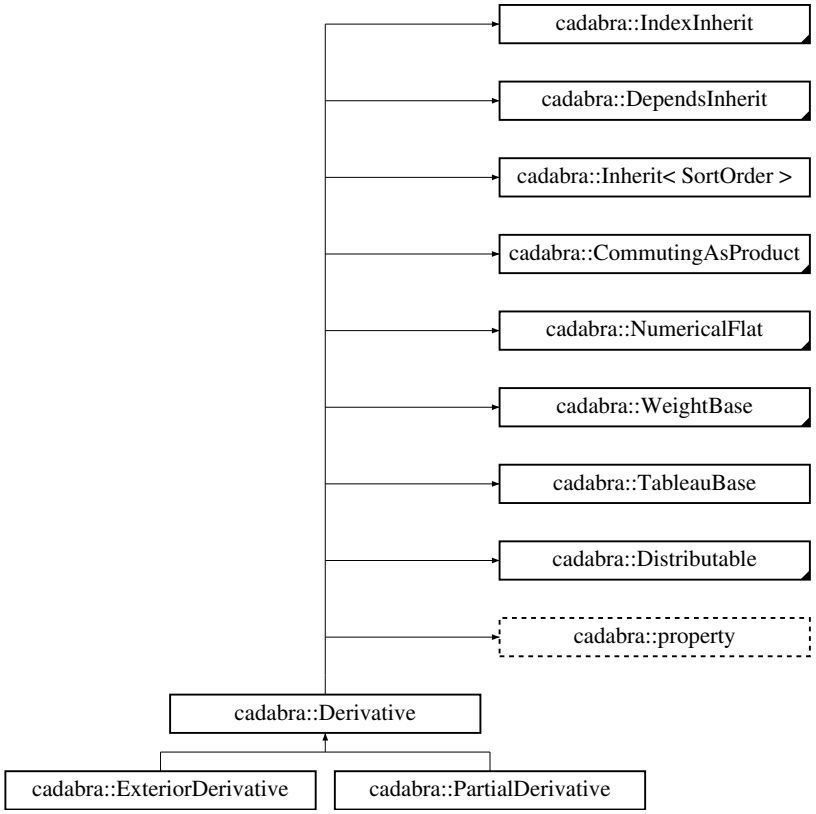
The documentation for this class was generated from the following files:

- core/properties/DependsInherit.hh
- core/properties/DependsInherit.cc

9.48 cadabra::Derivative Class Reference

`#include <Derivative.hh>`

Inheritance diagram for cadabra::Derivative:



Public Member Functions

- virtual ~Derivative ()
- virtual std::string name () const override
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const override
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const override
- virtual multiplier_t value (const Kernel &, Ex::iterator, const std::string &forcedlabel) const override

Additional Inherited Members

9.48.1 Constructor & Destructor Documentation

9.48.1.1 virtual cadabra::Derivative::~Derivative ( ) [inline], [virtual]

9.48.2 Member Function Documentation

9.48.2.1 TableauBase::tab_t Derivative::get_tab (const Properties & *properties*, Ex & *tr*, Ex::iterator *it*, unsigned int *num*) const [override], [virtual]

Implements cadabra::TableauBase.

Reimplemented in cadabra::PartialDerivative.

9.48.2.2 std::string Derivative::name ( ) const [override], [virtual]

Implements cadabra::property.

Reimplemented in cadabra::ExteriorDerivative, and cadabra::PartialDerivative.

9.48.2.3 unsigned int Derivative::size (const Properties & *properties,* Ex & *tr,* Ex::iterator *it*) const [override], [virtual]

Implements cadabra::TableauBase.

Reimplemented in cadabra::PartialDerivative.

9.48.2.4 multiplier_t Derivative::value (const Kernel & *kernel,* Ex::iterator *it,* const std::string & *forcedlabel*) const [override], [virtual]

Implements cadabra::WeightBase.

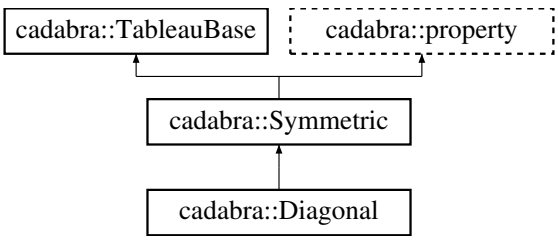
The documentation for this class was generated from the following files:

- core/properties/Derivative.hh
- core/properties/Derivative.cc

9.49 cadabra::Diagonal Class Reference

`#include <Diagonal.hh>`

Inheritance diagram for cadabra::Diagonal:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.49.1 Member Function Documentation

9.49.1.1 std::string Diagonal::name ( ) const [virtual]

Reimplemented from cadabra::Symmetric.

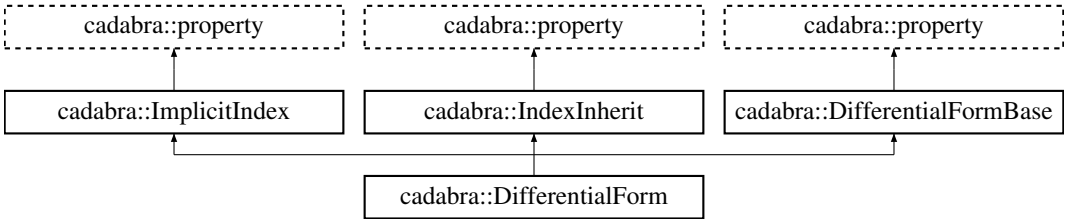
The documentation for this class was generated from the following files:

- core/properties/Diagonal.hh
- core/properties/Diagonal.cc

9.50 cadabra::DifferentialForm Class Reference

`#include <DifferentialForm.hh>`

Inheritance diagram for cadabra::DifferentialForm:



Public Member Functions

- virtual std::string name () const override
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual Ex degree (const Properties &, Ex::iterator) const override

Private Attributes

- Ex degree_

Additional Inherited Members

9.50.1 Member Function Documentation

9.50.1.1 Ex DifferentialForm::degree (const Properties & , Ex::iterator) const [override], [virtual]

Implements cadabra::DifferentialFormBase.

9.50.1.2 std::string DifferentialForm::name ( ) const [override], [virtual]

Reimplemented from cadabra::ImplicitIndex.

9.50.1.3 bool DifferentialForm::parse (const Kernel & , keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::ImplicitIndex.

9.50.2 Member Data Documentation

9.50.2.1 Ex cadabra::DifferentialForm::degree_ [private]

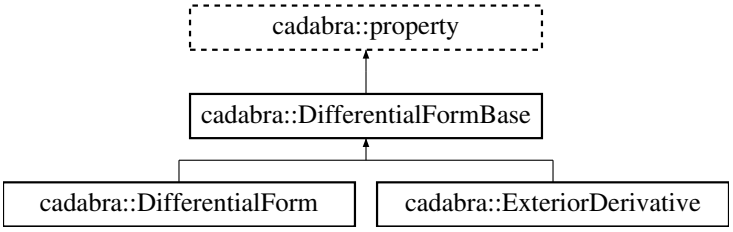
The documentation for this class was generated from the following files:

- core/properties/DifferentialForm.hh
- core/properties/DifferentialForm.cc

9.51 cadabra::DifferentialFormBase Class Reference

```
#include <DifferentialFormBase.hh>
```

Inheritance diagram for cadabra::DifferentialFormBase:



Public Member Functions

- virtual Ex degree (const Properties &, Ex::iterator) const =0

Additional Inherited Members

9.51.1 Member Function Documentation

9.51.1.1 virtual Ex cadabra::DifferentialFormBase::degree (const Properties & , Ex::iterator) const [pure virtual]

Implemented in cadabra::DifferentialForm, and cadabra::ExteriorDerivative.

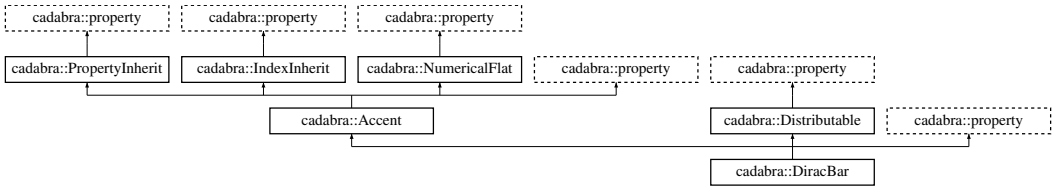
The documentation for this class was generated from the following file:

- core/properties/DifferentialFormBase.hh

9.52 cadabra::DiracBar Class Reference

`#include <DiracBar.hh>`

Inheritance diagram for cadabra::DiracBar:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.52.1 Member Function Documentation

9.52.1.1 std::string DiracBar::name ( ) const [virtual]

Implements cadabra::property.

The documentation for this class was generated from the following files:

- core/properties/DiracBar.hh
- core/properties/DiracBar.cc

9.53 cadabra::DisplayBase Class Reference

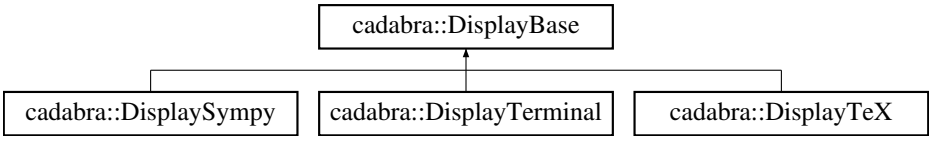
9.53.1 Description

Base class for all display classes.

A key difficulty with printing is to figure out when to print additional brackets for objects which would otherwise not render correctly. For example, a sum inside a product need brackets, but it does not need brackets when it is given as an argument to a function, because then the brackets are already there from the function call.

`#include <DisplayBase.hh>`

Inheritance diagram for cadabra::DisplayBase:



Public Member Functions

- DisplayBase (const Kernel &, const Ex &)
- void output (std::ostream &)
- void output (std::ostream &, Ex::iterator)
- virtual void dispatch (std::ostream &, Ex::iterator)=0

Protected Member Functions

- virtual bool needs_brackets (Ex::iterator it)=0

  Determine if a node needs extra brackets around it.

Protected Attributes

- const Ex & tree
- const Kernel & kernel

9.53.2 Constructor & Destructor Documentation

9.53.2.1 DisplayBase::DisplayBase (const Kernel & *k,* const Ex & *e*)

9.53.3 Member Function Documentation

9.53.3.1 virtual void cadabra::DisplayBase::dispatch (std::ostream & , Ex::iterator) [pure virtual]

Implemented in cadabra::DisplaySympy, cadabra::DisplayTeX, and cadabra::DisplayTerminal.

9.53.3.2 virtual bool cadabra::DisplayBase::needs_brackets (Ex::iterator *it*) [protected], [pure virtual]

Determine if a node needs extra brackets around it.

Uses context from the parent node if necessary. Has to be implemented in a derived class, because the answer depends on the printing method (e.g. (a+b)/c needs brackets when printed like this, but does not need brackets when printed as {a+b}{c}).

Implemented in cadabra::DisplaySympy, cadabra::DisplayTerminal, and cadabra::DisplayTeX.

9.53.3.3 void DisplayBase::output (std::ostream & *str*)

9.53.3.4 void DisplayBase::output (std::ostream & *str,* Ex::iterator *it*)

9.53.4 Member Data Documentation

9.53.4.1 const Kernel& cadabra::DisplayBase::kernel [protected]

9.53.4.2 const Ex& cadabra::DisplayBase::tree [protected]

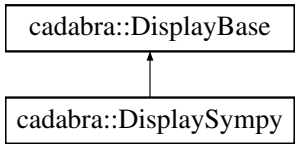
The documentation for this class was generated from the following files:

- core/DisplayBase.hh
- core/DisplayBase.cc

9.54 cadabra::DisplaySympy Class Reference

`#include <DisplaySympy.hh>`

Inheritance diagram for cadabra::DisplaySympy:



Public Member Functions

- DisplaySympy (const Kernel &, const Ex &)
- void import (Ex &)

  Rewrite the output of sympy back into a notation used by Cadabra.

Protected Member Functions

- virtual bool needs_brackets (Ex::iterator it) override

  Determine if a node needs extra brackets around it.

Private Member Functions

- void print_multiplier (std::ostream &, Ex::iterator)

  Output the expression to a sympy-readable form.
- void print_opening_bracket (std::ostream &, str_node::bracket_t)
- void print_closing_bracket (std::ostream &, str_node::bracket_t)
- void print_parent_rel (std::ostream &, str_node::parent_rel_t, bool first)
- void print_children (std::ostream &, Ex::iterator, int skip=0)
- virtual void dispatch (std::ostream &, Ex::iterator) override

  For every object encountered, dispatch will figure out the most appropriate way to convert it into a LaTeX expression.
- void print_productlike (std::ostream &, Ex::iterator, const std::string &inbetween)

  Printing members for various standard constructions, e.g.
- void print_sumlike (std::ostream &, Ex::iterator)
- void print_fraclike (std::ostream &, Ex::iterator)
- void print_commalike (std::ostream &, Ex::iterator)
- void print_arrowlike (std::ostream &, Ex::iterator)
- void print_powlike (std::ostream &, Ex::iterator)
- void print_intlike (std::ostream &, Ex::iterator)
- void print_equalitylike (std::ostream &, Ex::iterator)
- void print_components (std::ostream &, Ex::iterator)
- void print_partial (std::ostream &str, Ex::iterator it)
- void print_matrix (std::ostream &str, Ex::iterator it)
- void print_other (std::ostream &str, Ex::iterator it)
- bool children_have_brackets (Ex::iterator ch) const

Private Attributes

- std::map< std::string, std::string > symmap

  Map from Cadabra symbols to Sympy symbols.
- std::map< nset_t::iterator, Ex, nset_it_less > depsyms

  Map from symbols which have had dependencies added to an expression containing these dependencies.

Additional Inherited Members

9.54.1 Constructor & Destructor Documentation

9.54.1.1 DisplaySympy::DisplaySympy (const Kernel & *kernel,* const Ex & *e*)

9.54.2 Member Function Documentation

9.54.2.1 bool DisplaySympy::children_have_brackets (Ex::iterator *ch*) const [private]

9.54.2.2 void DisplaySympy::dispatch (std::ostream & *str,* Ex::iterator *it*) [override], [private], [virtual]

For every object encountered, dispatch will figure out the most appropriate way to convert it into a LaTeX expression.

This may be done by simply looking at the object's name (e.g. will print as a product) but may also involve looking up properties and deciding on the best course of action based on the attached properties.

Implements cadabra::DisplayBase.

9.54.2.3 void DisplaySympy::import (Ex & *ex*)

Rewrite the output of sympy back into a notation used by Cadabra.

This in particular involves converting 'sin' and friends to '' and so on, as well as converting all the greek symbols. Currently only acts node-by-node, does not do anything complicated with trees.

9.54.2.4 bool DisplaySympy::needs_brackets (Ex::iterator *it*) [override], [protected], [virtual]

Determine if a node needs extra brackets around it.

Uses context from the parent node if necessary. Has to be implemented in a derived class, because the answer depends on the printing method (e.g. (a+b)/c needs brackets when printed like this, but does not need brackets when printed as {a+b}{c}).

Implements cadabra::DisplayBase.

9.54.2.5 void DisplaySympy::print_arrowlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.6 void DisplaySympy::print_children (std::ostream & *str,* Ex::iterator *it,* int *skip* = 0) [private]

9.54.2.7 void DisplaySympy::print_closing_bracket (std::ostream & *str,* str_node::bracket_t *br*) [private]

9.54.2.8 void DisplaySympy::print_commalike (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.9 void DisplaySympy::print_components (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.10 void DisplaySympy::print_equalitylike (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.11 void DisplaySympy::print_fraclike (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.12 void DisplaySympy::print_intlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.13 void DisplaySympy::print_matrix (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.14 void DisplaySympy::print_multiplier (std::ostream & *str,* Ex::iterator *it*) [private]

Output the expression to a sympy-readable form.

For symbols which cannot be parsed by sympy, this can convert to an alternative, Such rewrites can then be undone by the 'import' member above.

9.54.2.15 void DisplaySympy::print_opening_bracket (std::ostream & *str,* str_node::bracket_t *br*) [private]

9.54.2.16 void DisplaySympy::print_other (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.17 void DisplaySympy::print_parent_rel (std::ostream & *str,* str_node::parent_rel_t *pr,* bool *first*) [private]

9.54.2.18 void DisplaySympy::print_partial (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.19 void DisplaySympy::print_powlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.2.20 void DisplaySympy::print_productlike (std::ostream & *str,* Ex::iterator *it,* const std::string & *inbetween*) [private]

Printing members for various standard constructions, e.g.

print as a list, or as a decorated symbol with super/subscripts etc. The names reflect the structure of the output, not necessarily the meaning or name of the object that is being printed.

9.54.2.21 void DisplaySympy::print_sumlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.54.3 Member Data Documentation

9.54.3.1 std::map<nset_t::iterator, Ex, nset_it_less> cadabra::DisplaySympy::depsyms [private]

Map from symbols which have had dependencies added to an expression containing these dependencies.

9.54.3.2 std::map<std::string, std::string> cadabra::DisplaySympy::symmap [private]

Map from Cadabra symbols to Sympy symbols.

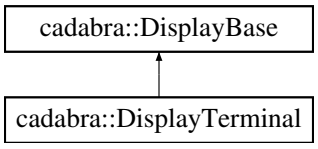
The documentation for this class was generated from the following files:

- core/DisplaySympy.hh
- core/DisplaySympy.cc

9.55 cadabra::DisplayTerminal Class Reference

`#include <DisplayTerminal.hh>`

Inheritance diagram for cadabra::DisplayTerminal:



Public Member Functions

- DisplayTerminal (const Kernel &, const Ex &)

Protected Member Functions

- virtual void dispatch (std::ostream &, Ex::iterator) override
- virtual bool needs_brackets (Ex::iterator it) override

  Determine if a node needs extra brackets around it.

Private Member Functions

- void print_multiplier (std::ostream &, Ex::iterator, int mult=1)
- void print_opening_bracket (std::ostream &, str_node::bracket_t, str_node::parent_rel_t)
- void print_closing_bracket (std::ostream &, str_node::bracket_t, str_node::parent_rel_t)
- void print_parent_rel (std::ostream &, str_node::parent_rel_t, bool first)
- void print_children (std::ostream &, Ex::iterator, int skip=0)
- void print_productlike (std::ostream &, Ex::iterator, const std::string &inbetween)

  Printing members for various standard constructions, e.g.
- void print_sumlike (std::ostream &, Ex::iterator)
- void print_fraclike (std::ostream &, Ex::iterator)
- void print_commalike (std::ostream &, Ex::iterator)
- void print_arrowlike (std::ostream &, Ex::iterator)
- void print_powlike (std::ostream &, Ex::iterator)
- void print_intlike (std::ostream &, Ex::iterator)
- void print_equalitylike (std::ostream &, Ex::iterator)
- void print_commutator (std::ostream &str, Ex::iterator it, bool comm)
- void print_components (std::ostream &, Ex::iterator)
- void print_other (std::ostream &, Ex::iterator)
- bool children_have_brackets (Ex::iterator ch) const

Private Attributes

- int bracket_level =0
- std::map< std::string, std::string > symmap

Additional Inherited Members

9.55.1 Constructor & Destructor Documentation

9.55.1.1 DisplayTerminal::DisplayTerminal (const Kernel & *k*, const Ex & *e*)

9.55.2 Member Function Documentation

9.55.2.1 bool DisplayTerminal::children_have_brackets (Ex::iterator *ch*) const [private]

9.55.2.2 void DisplayTerminal::dispatch (std::ostream & *str*, Ex::iterator *it*) [override], [protected], [virtual]

Implements cadabra::DisplayBase.

9.55.2.3 bool DisplayTerminal::needs_brackets (Ex::iterator *it*) [override], [protected], [virtual]

Determine if a node needs extra brackets around it.

Uses context from the parent node if necessary. Has to be implemented in a derived class, because the answer depends on the printing method (e.g. (a+b)/c needs brackets when printed like this, but does not need brackets when printed as {a+b}{c}).

Implements cadabra::DisplayBase.

9.55.2.4 void DisplayTerminal::print_arrowlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.2.5 void DisplayTerminal::print_children (std::ostream & *str,* Ex::iterator *it,* int *skip* = 0) [private]

9.55.2.6 void DisplayTerminal::print_closing_bracket (std::ostream & *str,* str_node::bracket_t *br,* str_node::parent_rel_t *pr*) [private]

9.55.2.7 void DisplayTerminal::print_commalike (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.2.8 void DisplayTerminal::print_commutator (std::ostream & *str,* Ex::iterator *it,* bool *comm*) [private]

9.55.2.9 void DisplayTerminal::print_components (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.2.10 void DisplayTerminal::print_equalitylike (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.2.11 void DisplayTerminal::print_fraclike (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.2.12 void DisplayTerminal::print_intlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.2.13 void DisplayTerminal::print_multiplier (std::ostream & *str,* Ex::iterator *it,* int *mult* = 1) [private]

9.55.2.14 void DisplayTerminal::print_opening_bracket (std::ostream & *str,* str_node::bracket_t *br,* str_node::parent_rel_t *pr*) [private]

9.55.2.15 void DisplayTerminal::print_other (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.2.16 void DisplayTerminal::print_parent_rel (std::ostream & *str,* str_node::parent_rel_t *pr,* bool *first*) [private]

9.55.2.17 void DisplayTerminal::print_powlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.2.18 void DisplayTerminal::print_productlike (std::ostream & *str,* Ex::iterator *it,* const std::string & *inbetween*) [private]

Printing members for various standard constructions, e.g.

print as a list, or as a decorated symbol with super/subscripts etc. The names reflect the structure of the output, not necessarily the meaning or name of the object that is being printed.

9.55.2.19 void DisplayTerminal::print_sumlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.55.3 Member Data Documentation

9.55.3.1 int cadabra::DisplayTerminal::bracket_level =0 [private]

9.55.3.2 std::map<std::string, std::string> cadabra::DisplayTerminal::symmap [private]

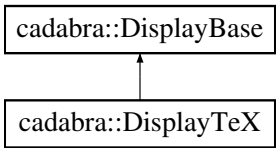
The documentation for this class was generated from the following files:

- core/DisplayTerminal.hh
- core/DisplayTerminal.cc

9.56 cadabra::DisplayTeX Class Reference

`#include <DisplayTeX.hh>`

Inheritance diagram for cadabra::DisplayTeX:



Public Member Functions

- DisplayTeX (const Kernel &, const Ex &)

Protected Member Functions

- virtual bool needs_brackets (Ex::iterator it) override

  Determine if a node needs extra brackets around it.

Private Member Functions

- void print_multiplier (std::ostream &, Ex::iterator, int mult=1)
- void print_opening_bracket (std::ostream &, str_node::bracket_t, str_node::parent_rel_t)
- void print_closing_bracket (std::ostream &, str_node::bracket_t, str_node::parent_rel_t)
- void print_parent_rel (std::ostream &, str_node::parent_rel_t, bool first)
- void print_children (std::ostream &, Ex::iterator, int skip=0)
- std::string texify (std::string) const
- virtual void dispatch (std::ostream &, Ex::iterator) override

  For every object encountered, dispatch will figure out the most appropriate way to convert it into a LaTeX expression.
- void print_productlike (std::ostream &, Ex::iterator, const std::string &inbetween)

  Printing members for various standard constructions, e.g.
- void print_sumlike (std::ostream &, Ex::iterator)
- void print_fraclike (std::ostream &, Ex::iterator)
- void print_commalike (std::ostream &, Ex::iterator)
- void print_arrowlike (std::ostream &, Ex::iterator)
- void print_powlike (std::ostream &, Ex::iterator)
- void print_intlike (std::ostream &, Ex::iterator)
- void print_equalitylike (std::ostream &, Ex::iterator)
- void print_commutator (std::ostream &str, Ex::iterator it, bool comm)
- void print_components (std::ostream &, Ex::iterator)
- void print_conditional (std::ostream &, Ex::iterator)
- void print_relation (std::ostream &, Ex::iterator)
- void print_indexbracket (std::ostream &, Ex::iterator)
- void print_wedgeproduct (std::ostream &, Ex::iterator)
- void print_tableau (std::ostream &, Ex::iterator)
- void print_ftableau (std::ostream &, Ex::iterator)
- void print_other (std::ostream &str, Ex::iterator it)
- bool children_have_brackets (Ex::iterator ch) const
- bool reads_as_operator (Ex::iterator obj, Ex::iterator arg) const

  Determine whether the indicated object 'obj', when acting on the single argument 'arg', reads as an operator, and therefore 'arg' does not need to be wrapped in brackets.

Private Attributes

- bool print_star =false
- bool tight_star =false
- bool tight_plus =false
- bool utf8_output =false
- bool latex_spacing =true
- bool latex_linefeeds =true
- bool extra_brackets_for_symbols =false
- int bracket_level =0
- std::map< std::string, std::string > symmap

  Map from Cadabra symbols to LaTeX symbols (for some typographic cleanup).
- std::set< std::string > curly_bracket_operators

  List of operators which need all arguments fed with curly brackets, not round ones (e.g.

Additional Inherited Members

9.56.1 Constructor & Destructor Documentation

9.56.1.1 DisplayTeX::DisplayTeX (const Kernel & *k,* const Ex & *e*)

9.56.2 Member Function Documentation

9.56.2.1 bool DisplayTeX::children_have_brackets (Ex::iterator *ch*) const [private]

9.56.2.2 void DisplayTeX::dispatch (std::ostream & *str,* Ex::iterator *it*) [override], [private], [virtual]

For every object encountered, dispatch will figure out the most appropriate way to convert it into a LaTeX expression.

This may be done by simply looking at the object's name (e.g. will print as a product) but may also involve looking up properties and deciding on the best course of action based on the attached properties.

Implements cadabra::DisplayBase.

9.56.2.3 bool DisplayTeX::needs_brackets (Ex::iterator *it*) [override], [protected], [virtual]

Determine if a node needs extra brackets around it.

Uses context from the parent node if necessary. Has to be implemented in a derived class, because the answer depends on the printing method (e.g. (a+b)/c needs brackets when printed like this, but does not need brackets when printed as {a+b}{c}).

Implements cadabra::DisplayBase.

9.56.2.4 void DisplayTeX::print_arrowlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.5 void DisplayTeX::print_children (std::ostream & *str,* Ex::iterator *it,* int *skip* = 0) [private]

9.56.2.6 void DisplayTeX::print_closing_bracket (std::ostream & *str,* str_node::bracket_t *br,* str_node::parent_rel_t *pr*) [private]

9.56.2.7 void DisplayTeX::print_commalike (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.8 void DisplayTeX::print_commutator (std::ostream & *str,* Ex::iterator *it,* bool *comm*) [private]

9.56.2.9 void DisplayTeX::print_components (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.10 void DisplayTeX::print_conditional (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.11 void DisplayTeX::print_equalitylike (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.12 void DisplayTeX::print_fraclike (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.13 void DisplayTeX::print_ftableau (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.14 void DisplayTeX::print_indexbracket (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.15 void DisplayTeX::print_intlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.16 void DisplayTeX::print_multiplier (std::ostream & *str,* Ex::iterator *it,* int *mult* = 1) [private]

9.56.2.17 void DisplayTeX::print_opening_bracket (std::ostream & *str,* str_node::bracket_t *br,* str_node::parent_rel_t *pr*) [private]

9.56.2.18 void DisplayTeX::print_other (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.19 void DisplayTeX::print_parent_rel (std::ostream & *str,* str_node::parent_rel_t *pr,* bool *first*) [private]

9.56.2.20 void DisplayTeX::print_powlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.21 void DisplayTeX::print_productlike (std::ostream & *str,* Ex::iterator *it,* const std::string & *inbetween*) [private]

Printing members for various standard constructions, e.g.

print as a list, or as a decorated symbol with super/subscripts etc. The names reflect the structure of the output, not necessarily the meaning or name of the object that is being printed.

9.56.2.22 void DisplayTeX::print_relation (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.23 void DisplayTeX::print_sumlike (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.24 void DisplayTeX::print_tableau (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.25 void DisplayTeX::print_wedgeproduct (std::ostream & *str,* Ex::iterator *it*) [private]

9.56.2.26 bool DisplayTeX::reads_as_operator (Ex::iterator *obj,* Ex::iterator *arg*) const [private]

Determine whether the indicated object 'obj', when acting on the single argument 'arg', reads as an operator, and therefore 'arg' does not need to be wrapped in brackets.

Example: the tree {A} can be displayed as ' A' (or rather, '{A}', instead of '(A)'.

9.56.2.27 std::string DisplayTeX::texify (std::string *str*) const [private]

9.56.3 Member Data Documentation

9.56.3.1 int cadabra::DisplayTeX::bracket_level =0 [private]

9.56.3.2 std::set<std::string> cadabra::DisplayTeX::curly_bracket_operators [private]

List of operators which need all arguments fed with curly brackets, not round ones (e.g.

).

9.56.3.3 bool cadabra::DisplayTeX::extra_brackets_for_symbols =false [private]

9.56.3.4 bool cadabra::DisplayTeX::latex_linefeeds =true [private]

9.56.3.5 bool cadabra::DisplayTeX::latex_spacing =true [private]

9.56.3.6 bool cadabra::DisplayTeX::print_star =false [private]

9.56.3.7 std::map<std::string, std::string> cadabra::DisplayTeX::symmap [private]

Map from Cadabra symbols to LaTeX symbols (for some typographic cleanup).

9.56.3.8 bool cadabra::DisplayTeX::tight_plus =false [private]

9.56.3.9 bool cadabra::DisplayTeX::tight_star =false [private]

9.56.3.10 bool cadabra::DisplayTeX::utf8_output =false [private]

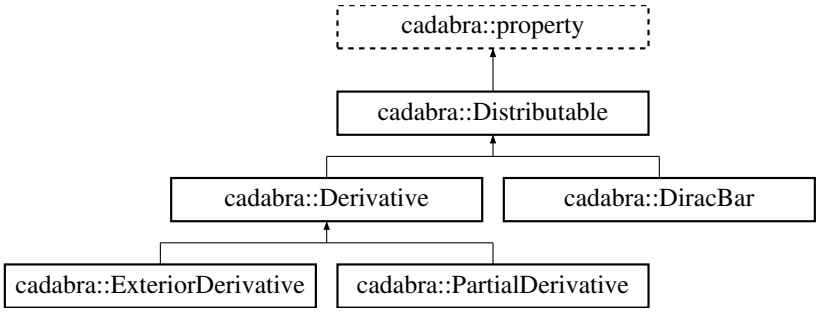
The documentation for this class was generated from the following files:

- core/DisplayTeX.hh
- core/DisplayTeX.cc

9.57 cadabra::Distributable Class Reference

`#include <Distributable.hh>`

Inheritance diagram for cadabra::Distributable:



Public Member Functions

- virtual ~Distributable ()
- virtual std::string name () const

Additional Inherited Members

9.57.1 Constructor & Destructor Documentation

9.57.1.1 virtual cadabra::Distributable::~Distributable ( ) [inline], [virtual]

9.57.2 Member Function Documentation

9.57.2.1 std::string Distributable::name ( ) const [virtual]

Implements cadabra::property.

Reimplemented in cadabra::Derivative, cadabra::ExteriorDerivative, cadabra::DiracBar, and cadabra::PartialDerivative.

The documentation for this class was generated from the following files:

- core/properties/Distributable.hh
- core/properties/Distributable.cc

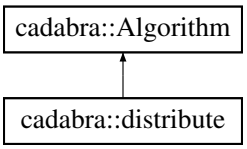
9.58 cadabra::distribute Class Reference

9.58.1 Description

Distribute factors over a sum, that is, $(A+B)C \rightarrow AC + BC$.

`#include <distribute.hh>`

Inheritance diagram for cadabra::distribute:



Public Member Functions

- distribute (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Additional Inherited Members

9.58.2 Constructor & Destructor Documentation

9.58.2.1 distribute::distribute (const Kernel & *k*, Ex & *tr*)

9.58.3 Member Function Documentation

9.58.3.1 Algorithm::result_t distribute::apply (iterator & *prod*) [virtual]

Implements cadabra::Algorithm.

9.58.3.2 bool distribute::can_apply (iterator *st*) [virtual]

Implements cadabra::Algorithm.

The documentation for this class was generated from the following files:

- core/algorithms/distribute.hh
- core/algorithms/distribute.cc

9.59 cadabra::DocumentThread Class Reference

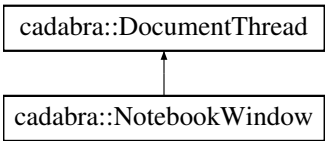
9.59.1 Description

A base class with all the logic to manipulate a Cadabra notebook document.

Relies on the various objects derived from ActionBase in order to get actual work done. All methods here will always run on the GUI thread.In order to implement a GUI, derive from both DocumentThread and GUIBase and then implement the virtual functions of the latter (those implement basic insertion/removal of notebook cells; the logic to figure out which ones and to implement the undo/redo stack is all in the GUI-agnostic DocumentThread).

`#include <DocumentThread.hh>`

Inheritance diagram for cadabra::DocumentThread:



Public Member Functions

- DocumentThread (GUIBase *)
- DocumentThread (const DocumentThread &)=delete

  It is not possible to copy-construct a DocumentThread as it holds on to resources which are not easily copied (such as GUI elements).
- void set_compute_thread (ComputeThread *)

  Let the notebook know about the ComputeThread so that it can send cells for evaluation.
- void build_visual_representation ()

  Ensure that the gui has an up-to-date representation of the dtree.
- void queue_action (std::shared_ptr< ActionBase >)

  All changes to the document should be made by submitting ActionBase derived objects to the 'queue_action' function, so that an undo stack can be kept.
- void new_document ()

  Setup an empty new document with a single Python input cell.
- void load_from_string (const std::string &)

  Load a new notebook from a JSON string.
- void undo ()

  One undo step.
- bool is_registered () const

  Determine if a user has been registered with the Cadabra log server.
- void set_user_details (const std::string &name, const std::string &email, const std::string &affiliation)

  Set user details which will be sent to the Cadabra log server.

Public Attributes

- friend ActionAddCell
- friend ActionPositionCursor
- friend ActionRemoveCell
- friend ActionSplitCell
- friend ActionSetRunStatus
- friend ActionInsertText
- friend ActionEraseText

Protected Types

- enum help_t { help_t::algorithm, help_t::property, help_t::latex, help_t::none }

  Help system.
- typedef std::stack< std::shared_ptr< ActionBase > > ActionStack

Protected Member Functions

- void process_action_queue ()

  Process the action queue.
- bool help_type_and_topic (const std::string &before, const std::string &after, help_t &help_type, std::string &help_topic) const

Protected Attributes

- GUIBase * gui
- ComputeThread * compute
- DTree doc

  The actual document tree.
- std::mutex stack_mutex

  The action undo/redo/todo stacks and logic to execute.
- ActionStack undo_stack
- ActionStack redo_stack
- std::queue< std::shared_ptr< ActionBase > > pending_actions
- bool disable_stacks
- bool registered

  Configuration options read from ~/.config/cadabra.conf.

9.59.2 Member Typedef Documentation

9.59.2.1 typedef std::stack<std::shared_ptr<ActionBase> > cadabra::DocumentThread::ActionStack [protected]

9.59.3 Member Enumeration Documentation

9.59.3.1 enum cadabra::DocumentThread::help_t [strong], [protected]

Help system.

Enumerator

algorithm

property

latex

none

9.59.4 Constructor & Destructor Documentation

9.59.4.1 DocumentThread::DocumentThread (GUIBase * *g*)

9.59.4.2 cadabra::DocumentThread::DocumentThread (const DocumentThread &) [delete]

It is not possible to copy-construct a DocumentThread as it holds on to resources which are not easily copied (such as GUI elements).

9.59.5 Member Function Documentation

9.59.5.1 void DocumentThread::build_visual_representation ( )

Ensure that the gui has an up-to-date representation of the dtree.

Traverses the entire tree so is expensive to run and should only be used when loading new documents or creating a new canvas view.

9.59.5.2 bool DocumentThread::help_type_and_topic (const std::string & *before*, const std::string & *after*, help_t & *help_type*, std::string & *help_topic*) const [protected]

9.59.5.3 bool DocumentThread::is_registered ( ) const

Determine if a user has been registered with the Cadabra log server.

9.59.5.4 void DocumentThread::load_from_string (const std::string & *json*)

Load a new notebook from a JSON string.

Should only be called from the GUI thread. Will cancel any pending operations on the existing notebook (if present) first.

9.59.5.5 void DocumentThread::new_document ( )

Setup an empty new document with a single Python input cell.

9.59.5.6 void DocumentThread::process_action_queue ( ) [protected]

Process the action queue.

It is allowed to call queue_action() above while this is running. So a running action can add more actions.

9.59.5.7 void DocumentThread::queue_action (std::shared_ptr< ActionBase > *ab*)

All changes to the document should be made by submitting ActionBase derived objects to the 'queue_action' function, so that an undo stack can be kept.

They are then processed by calling the 'process_action_queue' method (only available from this thread).

9.59.5.8 void DocumentThread::set_compute_thread (ComputeThread * *cl*)

Let the notebook know about the ComputeThread so that it can send cells for evaluation.

Notebook does NOT own this pointer.

9.59.5.9 void DocumentThread::set_user_details (const std::string & *name,* const std::string & *email,* const std::string & *affiliation*)

Set user details which will be sent to the Cadabra log server.

9.59.5.10 void DocumentThread::undo ( )

One undo step.

9.59.6 Member Data Documentation

9.59.6.1 friend cadabra::DocumentThread::ActionAddCell

9.59.6.2 friend cadabra::DocumentThread::ActionEraseText

9.59.6.3 friend cadabra::DocumentThread::ActionInsertText

9.59.6.4 friend cadabra::DocumentThread::ActionPositionCursor

9.59.6.5 friend cadabra::DocumentThread::ActionRemoveCell

9.59.6.6 friend cadabra::DocumentThread::ActionSetRunStatus

9.59.6.7 friend cadabra::DocumentThread::ActionSplitCell

9.59.6.8 ComputeThread* cadabra::DocumentThread::compute [protected]

9.59.6.9 bool cadabra::DocumentThread::disable_stacks [protected]

9.59.6.10 DTree cadabra::DocumentThread::doc [protected]

The actual document tree.

This object is only modified on the GUI thread, either directly by code in DocumentThread, or by code in the various objects derived from ActionBase. In particular, ComputeThread cannot access this tree directly.

9.59.6.11 GUIBase* cadabra::DocumentThread::gui [protected]

9.59.6.12 std::queue<std::shared_ptr<ActionBase> > cadabra::DocumentThread::pending_actions [protected]

9.59.6.13 ActionStack cadabra::DocumentThread::redo_stack [protected]

9.59.6.14 bool cadabra::DocumentThread::registered [protected]

Configuration options read from ~/.config/cadabra.conf.

9.59.6.15 std::mutex cadabra::DocumentThread::stack_mutex [protected]

The action undo/redo/todo stacks and logic to execute.

them. These stacks can be accessed from both the DocumentThread as well as the ComputeThread (the latter does it through the DocumentThread::queue_action method), so they need a mutex to access them.

9.59.6.16 ActionStack cadabra::DocumentThread::undo_stack [protected]

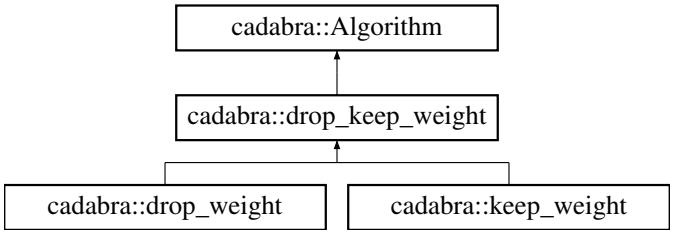
The documentation for this class was generated from the following files:

- client_server/DocumentThread.hh
- client_server/DocumentThread.cc

9.60 cadabra::drop_keep_weight Class Reference

`#include <drop_weight.hh>`

Inheritance diagram for cadabra::drop_keep_weight:



Public Member Functions

- drop_keep_weight (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- result_t do_apply (iterator &, bool keepthem)

Protected Attributes

- Ex arg
- const WeightInherit ∗ gmn
- const Weight ∗ wgh
- std::string label
- multiplier_t weight

Additional Inherited Members

9.60.1 Constructor & Destructor Documentation

9.60.1.1 drop_keep_weight::drop_keep_weight (const Kernel & *k,* Ex & *tr,* Ex & *a*)

9.60.2 Member Function Documentation

9.60.2.1 bool drop_keep_weight::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.60.2.2 Algorithm::result_t drop_keep_weight::do_apply (iterator & *it,* bool *keepthem*)

9.60.3 Member Data Documentation

9.60.3.1 Ex cadabra::drop_keep_weight::arg [protected]

9.60.3.2 const WeightInherit∗ cadabra::drop_keep_weight::gmn [protected]

9.60.3.3 std::string cadabra::drop_keep_weight::label [protected]

9.60.3.4 multiplier_t cadabra::drop_keep_weight::weight [protected]

9.60.3.5 const Weight∗ cadabra::drop_keep_weight::wgh [protected]

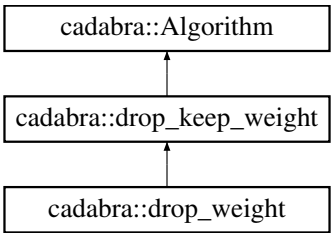
The documentation for this class was generated from the following files:

- core/algorithms/drop_weight.hh
- core/algorithms/drop_weight.cc

9.61 cadabra::drop_weight Class Reference

`#include <drop_weight.hh>`

Inheritance diagram for cadabra::drop_weight:



Public Member Functions

- drop_weight (const Kernel &, Ex &, Ex &)
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.61.1 Constructor & Destructor Documentation

9.61.1.1 drop_weight::drop_weight (const Kernel & *k,* Ex & *e,* Ex & *a*)

9.61.2 Member Function Documentation

9.61.2.1 Algorithm::result_t drop_weight::apply (iterator & *it*) [override], [virtual]

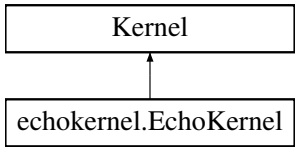
Implements cadabra::Algorithm.

The documentation for this class was generated from the following files:

- core/algorithms/drop_weight.hh
- core/algorithms/drop_weight.cc

9.62 echokernel.EchoKernel Class Reference

Inheritance diagram for echokernel.EchoKernel:



Public Member Functions

- def do_execute (self, code, silent, store_history=True, user_expressions=None, allow_stdin=False)

Static Public Attributes

- string implementation_version = '1.0'
- string language = 'no-op'
- string language_version = '0.1'
- dictionary language_info = {'mimetype': 'text/plain'}
- string banner = "Echo kernel - as useful as a parrot"
- string implementation = 'Echo'

9.62.1 Member Function Documentation

9.62.1.1 def echokernel.EchoKernel.do_execute (*self, code, silent, store_history* = `True`, *user_expressions* = `None`, *allow_stdin* = `False`)

9.62.2 Member Data Documentation

9.62.2.1 string echokernel.EchoKernel.banner = "Echo kernel - as useful as a parrot" [static]

9.62.2.2 string echokernel.EchoKernel.implementation = 'Echo' [static]

9.62.2.3 string echokernel.EchoKernel.implementation_version = '1.0' [static]

9.62.2.4 string echokernel.EchoKernel.language = 'no-op' [static]

9.62.2.5 dictionary echokernel.EchoKernel.language_info = {'mimetype': 'text/plain'} [static]

9.62.2.6 string echokernel.EchoKernel.language_version = '0.1' [static]

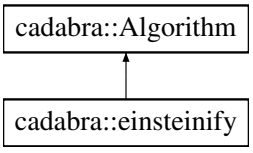
The documentation for this class was generated from the following file:

- core/echokernel.py

9.63 cadabra::einsteinify Class Reference

`#include <einsteinify.hh>`

Inheritance diagram for cadabra::einsteinify:



Public Member Functions

- einsteinify (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Attributes

- Ex metric

Additional Inherited Members

9.63.1 Constructor & Destructor Documentation

9.63.1.1 einsteinify::einsteinify (const Kernel & *k*, Ex & *e*, Ex & *m*)

9.63.2 Member Function Documentation

9.63.2.1 Algorithm::result_t einsteinify::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.63.2.2 bool einsteinify::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.63.3 Member Data Documentation

9.63.3.1 Ex cadabra::einsteinify::metric [private]

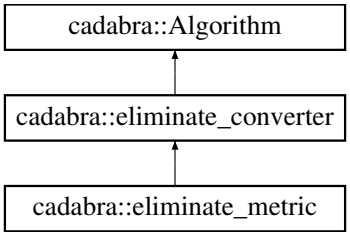
The documentation for this class was generated from the following files:

- core/algorithms/einsteinify.hh
- core/algorithms/einsteinify.cc

9.64 cadabra::eliminate_converter Class Reference

`#include <eliminate_metric.hh>`

Inheritance diagram for cadabra::eliminate_converter:



Public Member Functions

- eliminate_converter (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Protected Member Functions

- virtual bool is_conversion_object (iterator) const =0

Private Member Functions

- bool handle_one_index (iterator, iterator, iterator, sibling_iterator)

Private Attributes

- Ex preferred
- index_map_t ind_dummy
- index_map_t ind_free

Additional Inherited Members

9.64.1 Constructor & Destructor Documentation

9.64.1.1 eliminate_converter::eliminate_converter (const Kernel & *k,* Ex & *e,* Ex & *pref*)

9.64.2 Member Function Documentation

9.64.2.1 Algorithm::result_t eliminate_converter::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.64.2.2 bool eliminate_converter::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.64.2.3 bool eliminate_converter::handle_one_index (iterator *ind1,* iterator *ind2,* iterator *fit,* sibling_iterator *objs*) [private]

9.64.2.4 virtual bool cadabra::eliminate_converter::is_conversion_object (iterator) const [protected], [pure virtual]

Implemented in cadabra::eliminate_metric.

9.64.3 Member Data Documentation

9.64.3.1 index_map_t cadabra::eliminate_converter::ind_dummy [private]

9.64.3.2 index_map_t cadabra::eliminate_converter::ind_free [private]

9.64.3.3 Ex cadabra::eliminate_converter::preferred [private]

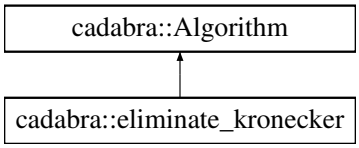
The documentation for this class was generated from the following files:

- core/algorithms/eliminate_metric.hh
- core/algorithms/eliminate_metric.cc

9.65 cadabra::eliminate_kronecker Class Reference

`#include <eliminate_kronecker.hh>`

Inheritance diagram for cadabra::eliminate_kronecker:



Public Member Functions

- eliminate_kronecker (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.65.1 Constructor & Destructor Documentation

9.65.1.1 eliminate_kronecker::eliminate_kronecker (const Kernel & *k*, Ex & *tr*)

9.65.2 Member Function Documentation

9.65.2.1 Algorithm::result_t eliminate_kronecker::apply (iterator & *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.65.2.2 bool eliminate_kronecker::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

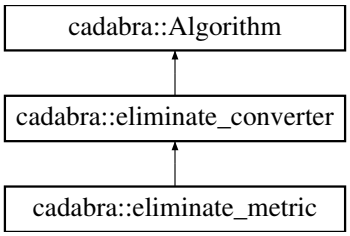
The documentation for this class was generated from the following files:

- core/algorithms/eliminate_kronecker.hh
- core/algorithms/eliminate_kronecker.cc

9.66 cadabra::eliminate_metric Class Reference

`#include <eliminate_metric.hh>`

Inheritance diagram for cadabra::eliminate_metric:



Public Member Functions

- eliminate_metric (const Kernel &, Ex &, Ex &)

Protected Member Functions

- virtual bool is_conversion_object (iterator) const override

Additional Inherited Members

9.66.1 Constructor & Destructor Documentation

9.66.1.1 eliminate_metric::eliminate_metric (const Kernel & *k*, Ex & *e*, Ex & *pref*)

9.66.2 Member Function Documentation

9.66.2.1 bool eliminate_metric::is_conversion_object (iterator *fit*) const [override], [protected], [virtual]

Implements cadabra::eliminate_converter.

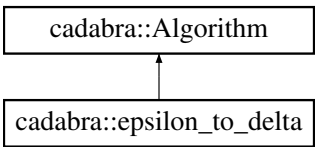
The documentation for this class was generated from the following files:

- core/algorithms/eliminate_metric.hh
- core/algorithms/eliminate_metric.cc

9.67 cadabra::epsilon_to_delta Class Reference

`#include <epsilon_to_delta.hh>`

Inheritance diagram for cadabra::epsilon_to_delta:



Public Member Functions

- epsilon_to_delta (const Kernel &, Ex &, bool reduce_)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Attributes

- bool reduce
- std::vector< Ex::iterator > epsilons
- int signature
- Ex repdelta

Additional Inherited Members

9.67.1 Constructor & Destructor Documentation

9.67.1.1 epsilon_to_delta::epsilon_to_delta (const Kernel & *k*, Ex & *tr*, bool *reduce_*)

9.67.2 Member Function Documentation

9.67.2.1 Algorithm::result_t epsilon_to_delta::apply (iterator & *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.67.2.2 bool epsilon_to_delta::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.67.3 Member Data Documentation

9.67.3.1 std::vector<Ex::iterator> cadabra::epsilon_to_delta::epsilons [private]

9.67.3.2 bool cadabra::epsilon_to_delta::reduce [private]

9.67.3.3 Ex cadabra::epsilon_to_delta::repdelta [private]

9.67.3.4 int cadabra::epsilon_to_delta::signature [private]

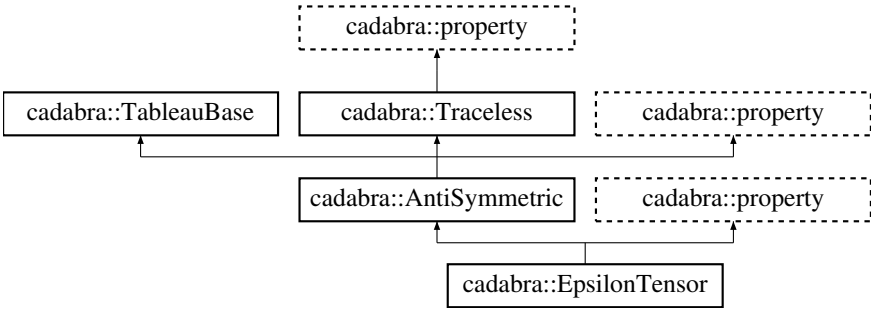
The documentation for this class was generated from the following files:

- core/algorithms/epsilon_to_delta.hh
- core/algorithms/epsilon_to_delta.cc

9.68 cadabra::EpsilonTensor Class Reference

`#include <EpsilonTensor.hh>`

Inheritance diagram for cadabra::EpsilonTensor:



Public Member Functions

- virtual ~EpsilonTensor ()
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual std::string name () const override

Public Attributes

- Ex metric
- Ex krdelta

Additional Inherited Members

9.68.1 Constructor & Destructor Documentation

9.68.1.1 virtual cadabra::EpsilonTensor::~EpsilonTensor ( ) [inline], [virtual]

9.68.2 Member Function Documentation

9.68.2.1 std::string EpsilonTensor::name ( ) const [override], [virtual]

Implements cadabra::property.

9.68.2.2 bool EpsilonTensor::parse (const Kernel & *k*, keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

9.68.3 Member Data Documentation

9.68.3.1 Ex cadabra::EpsilonTensor::krdelta

9.68.3.2 Ex cadabra::EpsilonTensor::metric

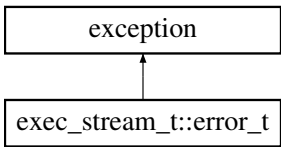
The documentation for this class was generated from the following files:

- core/properties/EpsilonTensor.hh
- core/properties/EpsilonTensor.cc

9.69 exec_stream_t::error_t Class Reference

`#include <exec-stream.h>`

Inheritance diagram for exec_stream_t::error_t:



Public Member Functions

- error_t (std::string const &msg)
- error_t (std::string const &msg, error_code_t code)
- ~error_t () throw ()
- virtual char const * what () const throw ()

Protected Member Functions

- error_t ()
- void compose (std::string const &msg, error_code_t code)

Protected Attributes

- std::string m_msg

9.69.1 Constructor & Destructor Documentation

9.69.1.1 exec_stream_t::error_t::error_t (std::string const & *msg*)

9.69.1.2 exec_stream_t::error_t::error_t (std::string const & *msg*, error_code_t *code*)

9.69.1.3 exec_stream_t::error_t::∼error_t ( ) throw)

9.69.1.4 exec_stream_t::error_t::error_t ( ) [protected]

9.69.2 Member Function Documentation

9.69.2.1 void exec_stream_t::error_t::compose (std::string const & *msg*, error_code_t *code*) [protected]

9.69.2.2 char const ∗ exec_stream_t::error_t::what ( ) const throw) [virtual]

9.69.3 Member Data Documentation

9.69.3.1 std::string exec_stream_t::error_t::m_msg [protected]

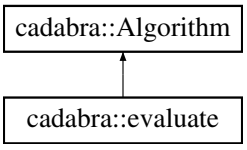
The documentation for this class was generated from the following files:

- frontend/common/exec-stream.h
- frontend/common/exec-stream.cc

9.70 cadabra::evaluate Class Reference

```
#include <evaluate.hh>
```

Inheritance diagram for cadabra::evaluate:



Public Member Functions

- evaluate (const Kernel &, Ex &, const Ex &component_values, bool rhs=false)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override
- void merge_components (iterator it1, iterator it2)

  Merge the information in two 'components' nodes at the given iterators, moving all out of the second one into the first one.

Private Member Functions

- bool is_component (iterator it) const
- iterator handle_components (iterator it)
- iterator handle_sum (iterator it)
- iterator handle_prod (iterator it)
- iterator handle_derivative (iterator it)
- iterator handle_epsilon (iterator it)
- iterator handle_factor (sibling_iterator sib, const index_map_t &full_ind_free)

  Replace a single factor with a 'components' ...
- iterator dense_factor (iterator sib, const index_map_t &ind_free, const index_map_t &ind_dummy)

  Expand a tensor factor into a components node with all components written out explicitly.
- void merge_component_children (iterator it)

  Merge entries in a single 'components' node when they are for the.
- void cleanup_components (iterator it1)

  Cleanup all components in a 'components' node; that is, call the cleanup_dispatch function on them.
- void simplify_components (iterator)

  Simplify all components of a 'components' node by running sympy's simplify on them.
- iterator wrap_scalar_in_components_node (iterator sib)

  Wrap a non-component scalar node in a 'components' node.
- std::set< Ex, tree_exact_less_obj > dependencies (iterator it)

  Determine all the Coordinate dependencies of the object at 'it'.

Private Attributes

- const Ex & components
- bool only_rhs

Additional Inherited Members

9.70.1 Constructor & Destructor Documentation

9.70.1.1 evaluate::evaluate (const Kernel & *k*, Ex & *tr*, const Ex & *component_values*, bool *rhs* = `false`)

9.70.2 Member Function Documentation

9.70.2.1 Algorithm::result_t evaluate::apply (iterator & *it*) `[override], [virtual]`

Implements cadabra::Algorithm.

9.70.2.2 bool evaluate::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.70.2.3 void evaluate::cleanup_components (iterator *it1*) [private]

Cleanup all components in a 'components' node; that is, call the cleanup_dispatch function on them.

9.70.2.4 Ex::iterator evaluate::dense_factor (iterator *sib*, const index_map_t & *ind_free*, const index_map_t & *ind_dummy*) [private]

Expand a tensor factor into a components node with all components written out explicitly.

Used when there is no sparse rule matching this factor.

9.70.2.5 std::set< Ex, tree_exact_less_obj > evaluate::dependencies (iterator *it*) [private]

Determine all the Coordinate dependencies of the object at 'it'.

For the time being this can only be a 'components' node.

9.70.2.6 Ex::iterator evaluate::handle_components (iterator *it*) [private]

9.70.2.7 Ex::iterator evaluate::handle_derivative (iterator *it*) [private]

9.70.2.8 Ex::iterator evaluate::handle_epsilon (iterator *it*) [private]

9.70.2.9 Ex::iterator evaluate::handle_factor (sibling_iterator *sib*, const index_map_t & *full_ind_free*) [private]

Replace a single factor with a 'components' ...

The full_ind_free argument can contain a list of indices in the order in which values should be stored in index value sets.

9.70.2.10 Ex::iterator evaluate::handle_prod (iterator *it*) [private]

9.70.2.11 Ex::iterator evaluate::handle_sum (iterator *it*) [private]

9.70.2.12 bool evaluate::is_component (iterator *it*) const [private]

9.70.2.13 void evaluate::merge_component_children (iterator *it*) [private]

Merge entries in a single 'components' node when they are for the.

same index value(s).

9.70.2.14 void evaluate::merge_components (iterator *it1*, iterator *it2*)

Merge the information in two 'components' nodes at the given iterators, moving all out of the second one into the first one.

9.70.2.15 void evaluate::simplify_components (iterator *it*) [private]

Simplify all components of a 'components' node by running sympy's simplify on them.

Returns a replacement iterator to the top.

9.70.2.16 Algorithm::iterator evaluate::wrap_scalar_in_components_node (iterator *sib*) [private]

Wrap a non-component scalar node in a 'components' node.

9.70.3 Member Data Documentation

9.70.3.1 const Ex& cadabra::evaluate::components [private]

9.70.3.2 bool cadabra::evaluate::only_rhs [private]

The documentation for this class was generated from the following files:

- core/algorithms/evaluate.hh
- core/algorithms/evaluate.cc

9.71 cadabra::Ex Class Reference

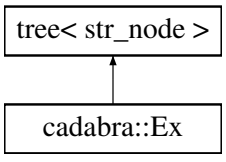
9.71.1 Description

Basic storage class for symbolic mathemematical expressions.

The full meaning of an expression typically requires knowledge about properties of patterns in it, which this class does *not* contain. All property dependent algorithms acting on Ex objects are in Algorithm.hh.

```
#include <Storage.hh>
```

Inheritance diagram for cadabra::Ex:



Public Types

- enum result_t { l_checkpointed, l_no_action, l_applied, l_error }

  Keeping track of what algorithms have done to this expression.

Public Member Functions

- Ex ()
- Ex (tree< str_node >::iterator)
- Ex (const str_node &)
- Ex (const Ex &)
- Ex (const std::string &)

  Initialise with given string as head node (does not parse this string).
- Ex (int)
- result_t state () const
- void update_state (result_t)
- void reset_state ()
- bool changed_state ()

  A status query method mainly to implement a simple method to apply algorithms until they converge.
- bool is_rational () const

  Test if the expression is a rational number.
- multiplier_t to_rational () const
- std::ostream & print_entire_tree (std::ostream &str) const

  Output helpers mainly for debugging purposes.
- iterator named_parent (iterator it, const std::string &) const

  Step up until matching node is found (if current node matches, do nothing)
- iterator erase_expression (iterator it)
- hashval_t calc_hash (iterator it) const

  Calculate the hash value for the subtree starting at 'it'.
- multiplier_t arg_to_num (sibling_iterator, unsigned int) const
- unsigned int number_of_steps (iterator it) const
- unsigned int number_of_equations () const
- unsigned int equation_number (iterator it) const
- nset_t::iterator equation_label (iterator it) const
- iterator equation_by_number (unsigned int i) const
- iterator equation_by_name (nset_t::iterator it) const
- iterator equation_by_name (nset_t::iterator it, unsigned int &) const
- iterator equation_by_number_or_name (iterator it, unsigned int last_used_equation) const
- iterator equation_by_number_or_name (iterator it, unsigned int last_used_equation, unsigned int &) const
- std::string equation_number_or_name (iterator it, unsigned int last_used_equation) const
- iterator procedure_by_name (nset_t::iterator it) const
- iterator replace_index (iterator position, const iterator &from, bool keep_parent_rel=false)

  Replace the index-like object (originally intended to replace indices only, but now used also for e.g.
- iterator move_index (iterator position, const iterator &from)

  As in replace_index, but moves the index rather than making a copy (so that iterators.
- void list_wrap_single_element (iterator &)

  Make sure that the node pointed to is a \comma object, i.e.
- void list_unwrap_single_element (iterator &)
- iterator flatten_and_erase (iterator position)

  Replace the node with the children of the node, useful for e.g.
- bool operator== (const Ex &other) const

  Compare two Ex objects for exact equality; no dummy equivalence or other things that require property information.

- void push_history (Ex selector)

  Push a copy of the current state of the expression onto the history stack.
- Ex pop_history ()

  Pop the most recent state of the expression off the history stack; returns the selector that got us there.
- int history_size () const

  Return the size of the history; 0 means no history, just the current expression.

Static Public Member Functions

- static std::ostream & print_python (std::ostream &str, Ex::iterator it)

  Display expression in Python/Cadabra input form.
- static std::ostream & print_recursive_treeform (std::ostream &str, Ex::iterator it)
- static std::ostream & print_recursive_treeform (std::ostream &str, Ex::iterator it, unsigned int &number)
- static sibling_iterator arg (iterator, unsigned int)

  Quick access to arguments or argument lists for A(B)(C,D) type nodes.
- static unsigned int arg_size (sibling_iterator)

Private Attributes

- result_t state_
- std::vector< tree< str_node > > history
- std::vector< tree< str_node > > selectors

  Patterns which describe how to get from one history step to the next.

9.71.2 Member Enumeration Documentation

9.71.2.1 enum cadabra::Ex::result_t

Keeping track of what algorithms have done to this expression.

After a reset_state (or at initialisation), the expression sits in the 'checkpointed' state. When an algorithm acts, it can then move to 'no_action' (unchanged), 'applied' (changed) or 'error'. Once it is in 'error', it will stay there until the next 'reset'. FIXME: the following should implement a stack of states, so that it can be used with nested functions.

Enumerator

l_checkpointed

l_no_action

l_applied

l_error

9.71.3 Constructor & Destructor Documentation

9.71.3.1 cadabra::Ex::Ex ( )

9.71.3.2 cadabra::Ex::Ex (tree< str_node >::iterator *it*)

9.71.3.3 cadabra::Ex::Ex (const str_node & *x*)

9.71.3.4 cadabra::Ex::Ex (const Ex & *other*)

9.71.3.5 cadabra::Ex::Ex (const std::string & *str*)

Initialise with given string as head node (does not parse this string).

9.71.3.6 cadabra::Ex::Ex (int *val*)

9.71.4 Member Function Documentation

9.71.4.1 Ex::sibling_iterator cadabra::Ex::arg (iterator *it,* unsigned int *num*) [static]

Quick access to arguments or argument lists for A(B)(C,D) type nodes.

9.71.4.2 unsigned int cadabra::Ex::arg_size (sibling_iterator *sib*) [static]

9.71.4.3 multiplier_t cadabra::Ex::arg_to_num (sibling_iterator *sib,* unsigned int *num*) const

9.71.4.4 hashval_t cadabra::Ex::calc_hash (iterator *it*) const

Calculate the hash value for the subtree starting at 'it'.

9.71.4.5 bool cadabra::Ex::changed_state ( )

A status query method mainly to implement a simple method to apply algorithms until they converge.

Returns true when the expression is in 'checkpointed' or 'applied' state. Will set the state to 'no_action'.

9.71.4.6 Ex::iterator cadabra::Ex::equation_by_name (nset_t::iterator *it*) const

9.71.4.7 Ex::iterator cadabra::Ex::equation_by_name (nset_t::iterator *it,* unsigned int & *tmp*) const

9.71.4.8 Ex::iterator cadabra::Ex::equation_by_number (unsigned int *i*) const

9.71.4.9 Ex::iterator cadabra::Ex::equation_by_number_or_name (iterator *it,* unsigned int *last_used_equation*) const

9.71.4.10 Ex::iterator cadabra::Ex::equation_by_number_or_name (iterator *it,* unsigned int *last_used_equation,* unsigned int & *real_eqno*) const

9.71.4.11 nset_t::iterator cadabra::Ex::equation_label (iterator *it*) const

9.71.4.12 unsigned int cadabra::Ex::equation_number (iterator *it*) const

9.71.4.13 std::string cadabra::Ex::equation_number_or_name (iterator *it,* unsigned int *last_used_equation*) const

9.71.4.14 Ex::iterator cadabra::Ex::erase_expression (iterator *it*)

9.71.4.15 Ex::iterator cadabra::Ex::flatten_and_erase (iterator *position*)

Replace the node with the children of the node, useful for e.g.

\prod{A} -> A. This algorithm takes care of the multiplier of the top node, i.e. it does 2\prod{A} -> 2 A. Returns an iterator to the new location of the first child of the original node.

9.71.4.16 int cadabra::Ex::history_size ( ) const

Return the size of the history; 0 means no history, just the current expression.

9.71.4.17 bool cadabra::Ex::is_rational ( ) const

Test if the expression is a rational number.

FIXME: add tests for integers as well.

9.71.4.18 void cadabra::Ex::list_unwrap_single_element (iterator & *it*)

9.71.4.19 void cadabra::Ex::list_wrap_single_element (iterator & *it*)

Make sure that the node pointed to is a \comma object, i.e.

wrap the node if not already inside such a \comma. DEPRECATED: in favour of 'do_list' in Functional.hh.

9.71.4.20 Ex::iterator cadabra::Ex::move_index (iterator *position,* const iterator & *from*)

As in replace_index, but moves the index rather than making a copy (so that iterators.

pointing to the original remain valid).

9.71.4.21 Ex::iterator cadabra::Ex::named_parent (iterator *it,* const std::string &) const

Step up until matching node is found (if current node matches, do nothing)

9.71.4.22 unsigned int cadabra::Ex::number_of_equations ( ) const

9.71.4.23 unsigned int cadabra::Ex::number_of_steps (iterator *it*) const

9.71.4.24 bool cadabra::Ex::operator== (const Ex & *other*) const

Compare two Ex objects for exact equality; no dummy equivalence or other things that require property information.

9.71.4.25 Ex cadabra::Ex::pop_history ( )

Pop the most recent state of the expression off the history stack; returns the selector that got us there.

9.71.4.26 std::ostream & cadabra::Ex::print_entire_tree (std::ostream & *str*) const

Output helpers mainly for debugging purposes.

9.71.4.27 std::ostream & cadabra::Ex::print_python (std::ostream & *str,* Ex::iterator *it*) [static]

Display expression in Python/Cadabra input form.

This is fairly straightforward so not handled with a separate DisplayBase derived class.

9.71.4.28 std::ostream & cadabra::Ex::print_recursive_treeform (std::ostream & *str,* Ex::iterator *it*) [static]

9.71.4.29 std::ostream & cadabra::Ex::print_recursive_treeform (std::ostream & *str,* Ex::iterator *it,* unsigned int & *number*) [static]

9.71.4.30 Ex::iterator cadabra::Ex::procedure_by_name (nset_t::iterator *it*) const

9.71.4.31 void cadabra::Ex::push_history (Ex *selector*)

Push a copy of the current state of the expression onto the history stack.

9.71.4.32 Ex::iterator cadabra::Ex::replace_index (iterator *position,* const iterator & *from,* bool *keep_parent_rel* = false)

Replace the index-like object (originally intended to replace indices only, but now used also for e.g.

normal function arguments, as in

$$\partial_z A(z)$$

with a replacement of z).

Note: this originally kept the bracket and parent_rel, but that is not a good idea, because it prevents us from changing those. If we want to use a _{z} pattern replacing a A(z) index, it is better to make a rule that matches (z) and at the time we find and match _{z}. So this should be handled by the replacement_map logic in Compare.cc.

9.71.4.33 void cadabra::Ex::reset_state ( )

9.71.4.34 Ex::result_t cadabra::Ex::state ( ) const

9.71.4.35 multiplier_t cadabra::Ex::to_rational ( ) const

9.71.4.36 void cadabra::Ex::update_state (Ex::result_t *newstate*)

9.71.5 Member Data Documentation

9.71.5.1 std::vector<tree<str_node> > cadabra::Ex::history [private]

9.71.5.2 std::vector<tree<str_node> > cadabra::Ex::selectors [private]

Patterns which describe how to get from one history step to the next.

9.71.5.3 result_t cadabra::Ex::state_ [private]

The documentation for this class was generated from the following files:

- core/Storage.hh
- core/Storage.cc

9.72 cadabra::Ex_comparator Class Reference

9.72.1 Description

A generic tree comparison class which will take into account index contractions and will also keep track of a replacement list for all types of cadabra wildcards.

The entry point is typically 'equal_subtree' or 'match_subproduct'.

`#include <Compare.hh>`

Public Types

- enum match_t { match_t::node_match =0, match_t::subtree_match =1, match_t::match_index_less =2, match_t::match_index_greater =3, match_t::no_match_indexpos_less =4, match_t::no_match_indexpos_greater =5, match_t::no_match_less =6, match_t::no_match_greater =7 }
- enum useprops_t { useprops_t::always =0, useprops_t::not_at_top, useprops_t::never =2 }
- typedef std::map< Ex, Ex, tree_exact_less_no_wildcards_obj > replacement_map_t

  Map for the replacement of nodes (indices, patterns).
- typedef std::map< nset_t::iterator, Ex::iterator, nset_it_less > subtree_replacement_map_t

  Map for the replacement of entire subtrees (object patterns).

Public Member Functions

- Ex_comparator (const Properties &)
- void clear ()

  Reset the object for a new match.
- match_t equal_subtree (Ex::iterator i1, Ex::iterator i2, useprops_t use_props=useprops_t::always, bool ignore_parent_rel=false)

  Match two subtrees taking into account symbol properties.
- match_t match_subproduct (const Ex &, Ex::sibling_iterator lhs, Ex::sibling_iterator tofind, Ex::sibling_iterator st, Ex::iterator conditions)

  Find a sub-product in a product.
- match_t match_subsum (const Ex &, Ex::sibling_iterator lhs, Ex::sibling_iterator tofind, Ex::sibling_iterator st, Ex::iterator conditions)

  Find a sub-sum in a sum.
- bool satisfies_conditions (Ex::iterator conditions, std::string &error)

  Check whether the a match found by calling equal_subtree or match_subproduct satisfies the conditions as stated.
- bool should_swap (Ex::iterator obj, match_t subtree_comparison)

  Determine whether two objects should be swapped according to the available SortOrder properties.
- int can_swap (Ex::iterator one, Ex::iterator two, match_t subtree_comparison, bool ignore_implicit_indices=false)

  Determine whether obj and obj+1 be exchanged? If yes, return the sign, if no return zero.
- int can_move_adjacent (Ex::iterator prod, Ex::sibling_iterator one, Ex::sibling_iterator two, bool fix_one=false)

  Determine whether object 'one' and 'two' can be moved next to each other by moving either one or the other: if fix_one==true the first node is kept fixed, otherwise the second node is kept fixed.

Public Attributes

- replacement_map_t replacement_map
- subtree_replacement_map_t subtree_replacement_map
- replacement_map_t index_value_map

  Map for matching of index names to index values.
- std::vector< Ex::sibling_iterator > factor_locations

  Information to keep track of where individual factors/terms in a sub-product/sub-sum were found, and (for sub-products) whether moving them into the searched-for order leads to sign flips.
- std::vector< int > factor_moving_signs
- multiplier_t term_ratio
- bool lhs_contains_dummies

  Flag to indicate whether additional care must be taken to handle dummies in the lhs of the pattern.

Protected Member Functions

- match_t compare (const Ex::iterator &, const Ex::iterator &, bool nobrackets=false, useprops_t use_↩ props=useprops_t::always, bool ignore_parent_rel=false)

  Internal entry point.
- int can_swap_prod_obj (Ex::iterator prod, Ex::iterator obj, bool)
- int can_swap_prod_prod (Ex::iterator prod1, Ex::iterator prod2, bool)
- int can_swap_sum_obj (Ex::iterator sum, Ex::iterator obj, bool)
- int can_swap_prod_sum (Ex::iterator prod, Ex::iterator sum, bool)
- int can_swap_sum_sum (Ex::iterator sum1, Ex::iterator sum2, bool)
- int can_swap_ilist_ilist (Ex::iterator obj1, Ex::iterator obj2)
- std::string tab () const
- match_t report (match_t r) const

Protected Attributes

- const Properties & properties

Static Protected Attributes

- static int offset =0

9.72.2 Member Typedef Documentation

9.72.2.1 typedef std::map<Ex, Ex, tree_exact_less_no_wildcards_obj> cadabra::Ex_comparator↩ ::replacement_map_t

Map for the replacement of nodes (indices, patterns).

9.72.2.2 typedef std::map<nset_t::iterator, Ex::iterator, nset_it_less> cadabra::Ex_comparator::subtree_↩ replacement_map_t

Map for the replacement of entire subtrees (object patterns).

9.72.3 Member Enumeration Documentation

9.72.3.1 enum cadabra::Ex_comparator::match_t [strong]

Enumerator

node_match

subtree_match

match_index_less

match_index_greater

no_match_indexpos_less

no_match_indexpos_greater

no_match_less

no_match_greater

9.72.3.2 enum cadabra::Ex_comparator::useprops_t [strong]

Enumerator

always

not_at_top

never

9.72.4 Constructor & Destructor Documentation

9.72.4.1 cadabra::Ex_comparator::Ex_comparator (const Properties & *k*)

9.72.5 Member Function Documentation

9.72.5.1 int cadabra::Ex_comparator::can_move_adjacent (Ex::iterator *prod,* Ex::sibling_iterator *one,* Ex::sibling_iterator *two,* bool *fix_one* = false)

Determine whether object 'one' and 'two' can be moved next to each other by moving either one or the other: if fix_one==true the first node is kept fixed, otherwise the second node is kept fixed.

9.72.5.2 int cadabra::Ex_comparator::can_swap (Ex::iterator *one,* Ex::iterator *two,* match_t *subtree_comparison,* bool *ignore_implicit_indices* = false)

Determine whether obj and obj+1 be exchanged? If yes, return the sign, if no return zero.

This is the general entry point for two arbitrary nodes (which may be a product or sum).

The last flag ('ignore_implicit_indices') is used to disable all checks dealing with implicit indices (this is useful for algorithms which re-order objects with implicit indices, which would otherwise always receive a 0 from this function).

9.72.5.3 int cadabra::Ex_comparator::can_swap_ilist_ilist (Ex::iterator *obj1,* Ex::iterator *obj2*) [protected]

9.72.5.4 int cadabra::Ex_comparator::can_swap_prod_obj (Ex::iterator *prod,* Ex::iterator *obj,* bool *ignore_implicit_indices*) [protected]

9.72.5.5 int cadabra::Ex_comparator::can_swap_prod_prod (Ex::iterator *prod1,* Ex::iterator *prod2,* bool *ignore_implicit_indices*) [protected]

9.72.5.6 int cadabra::Ex_comparator::can_swap_prod_sum (Ex::iterator *prod,* Ex::iterator *sum,* bool *ignore_implicit_indices*) [protected]

9.72.5.7 int cadabra::Ex_comparator::can_swap_sum_obj (Ex::iterator *sum,* Ex::iterator *obj,* bool *ignore_implicit_indices*) [protected]

9.72.5.8 int cadabra::Ex_comparator::can_swap_sum_sum (Ex::iterator *sum1,* Ex::iterator *sum2,* bool *ignore_implicit_indices*) [protected]

9.72.5.9 void cadabra::Ex_comparator::clear ( )

Reset the object for a new match.

9.72.5.10 **Ex_comparator::match_t** cadabra::Ex_comparator::compare (const Ex::iterator & *one,* const Ex::iterator & *two,* bool *nobrackets* = `false`, **useprops_t** *use_props* = **useprops_t::always**, bool *ignore_parent_rel* = `false`) [protected]

Internal entry point.

This comparison function tries to match a single node in the tree, except when the node is an index. Indices are considered to be leaf-nodes, and for these a full subtree match will be attempted (using subtree_compare).

9.72.5.11 **Ex_comparator::match_t** cadabra::Ex_comparator::equal_subtree (Ex::iterator *i1,* Ex::iterator *i2,* **useprops_t** *use_props* = **useprops_t::always**, bool *ignore_parent_rel* = `false`)

Match two subtrees taking into account symbol properties.

'i1' can be a pattern. Returns subtree_match or one of the no_match results. You need to fill lhs_contains_↩ dummies before calling! If use_props is false, it will not try to fetch any property information at the TOP level of the comparison. Properties will always be used at levels.

9.72.5.12 **Ex_comparator::match_t** cadabra::Ex_comparator::match_subproduct (const Ex & *tr,* Ex::sibling_iterator *lhs,* Ex::sibling_iterator *tofind,* Ex::sibling_iterator *st,* Ex::iterator *conditions*)

Find a sub-product in a product.

The 'lhs' iterator points to the product which we want to find, the 'tofind' iterator to the current factor which we are looking for. The product in which to search is pointed to by 'st'. Once 'tofind' is found, this routine calls itself to find the next factor in 'lhs'. If the next factor cannot be found, we backtrack and try to find the previous factor again (it may have appeared multiple times).

9.72.5.13 Ex_comparator::match_t cadabra::Ex_comparator::match_subsum (const Ex & *tr,* Ex::sibling_iterator *lhs,* Ex::sibling_iterator *tofind,* Ex::sibling_iterator *st,* Ex::iterator *conditions*)

Find a sub-sum in a sum.

The 'lhs' iterator points to the sum which we want to find, the 'tofind' iterator to the current term which we are looking for. The sum in which to search is pointed to by 'st'. Once 'tofind' is found, this routine calls itself to find the next term in 'lhs'. Since Cadabra assumes all terms in a sum commute, we do not need the backtracking logic of subproduct.

9.72.5.14 Ex_comparator::match_t cadabra::Ex_comparator::report (Ex_comparator::match_t *r*) const [protected]

9.72.5.15 bool cadabra::Ex_comparator::satisfies_conditions (Ex::iterator *conditions,* std::string & *error*)

Check whether the a match found by calling equal_subtree or match_subproduct satisfies the conditions as stated.

FIXME: document possible conditions.

9.72.5.16 bool cadabra::Ex_comparator::should_swap (Ex::iterator *obj,* match_t *subtree_comparison*)

Determine whether two objects should be swapped according to the available SortOrder properties.

9.72.5.17 std::string cadabra::Ex_comparator::tab ( ) const [protected]

9.72.6 Member Data Documentation

9.72.6.1 std::vector<Ex::sibling_iterator> cadabra::Ex_comparator::factor_locations

Information to keep track of where individual factors/terms in a sub-product/sub-sum were found, and (for sub-products) whether moving them into the searched-for order leads to sign flips.

9.72.6.2 std::vector<int> cadabra::Ex_comparator::factor_moving_signs

9.72.6.3 replacement_map_t cadabra::Ex_comparator::index_value_map

Map for matching of index names to index values.

Note: this is in the opposite order from replacement_map!

9.72.6.4 bool cadabra::Ex_comparator::lhs_contains_dummies

Flag to indicate whether additional care must be taken to handle dummies in the lhs of the pattern.

FIXME: would be better if this were automatic.

9.72.6.5 int cadabra::Ex_comparator::offset =0 [static], [protected]

9.72.6.6 const Properties& cadabra::Ex_comparator::properties [protected]

9.72.6.7 replacement_map_t cadabra::Ex_comparator::replacement_map

9.72.6.8 subtree_replacement_map_t cadabra::Ex_comparator::subtree_replacement_map

9.72.6.9 multiplier_t cadabra::Ex_comparator::term_ratio

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.73 cadabra::Ex_is_equivalent Class Reference

9.73.1 Description

Basic comparison operator for tree iterators, so we can use them as keys in maps.

`#include <Compare.hh>`

Public Member Functions

- Ex_is_equivalent (const Properties &)
- bool operator() (const Ex &, const Ex &)

Private Attributes

- const Properties & properties

9.73.2 Constructor & Destructor Documentation

9.73.2.1 cadabra::Ex_is_equivalent::Ex_is_equivalent (const Properties & *k*)

9.73.3 Member Function Documentation

9.73.3.1 bool cadabra::Ex_is_equivalent::operator() (const Ex & *one*, const Ex & *two*)

9.73.4 Member Data Documentation

9.73.4.1 const Properties& cadabra::Ex_is_equivalent::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.74 cadabra::Ex_is_less Class Reference

`#include <Compare.hh>`

Public Member Functions

- Ex_is_less (const Properties &, int mod_prel)
- bool operator() (const Ex &, const Ex &)

Private Attributes

- const Properties & properties
- int mod_prel

9.74.1 Constructor & Destructor Documentation

9.74.1.1 cadabra::Ex_is_less::Ex_is_less (const Properties & *k,* int *mod_prel*)

9.74.2 Member Function Documentation

9.74.2.1 bool cadabra::Ex_is_less::operator() (const Ex & *one,* const Ex & *two*)

9.74.3 Member Data Documentation

9.74.3.1 int cadabra::Ex_is_less::mod_prel [private]

9.74.3.2 const Properties& cadabra::Ex_is_less::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.75 cadabra::exchange Class Reference

`#include <Exchange.hh>`

Classes

- struct identical_tensors_t
- struct tensor_type_t

Static Public Member Functions

- static int collect_identical_tensors (const Properties &pr, Ex &tr, Ex::iterator it, std::vector< identical_↩ tensors_t > &idts)
- static unsigned int possible_singlets (Ex &, Ex::iterator)
- static bool get_node_gs (const Properties &, Ex &, Ex::iterator, std::vector< std::vector< int > > &)

9.75.1 Member Function Documentation

9.75.1.1 int exchange::collect_identical_tensors (const Properties & *pr,* Ex & *tr,* Ex::iterator *it,* std::vector< identical_tensors_t > & *idts*) [static]

9.75.1.2 bool exchange::get_node_gs (const Properties & *properties,* Ex & *tr,* Ex::iterator *it,* std::vector< std::vector< int > > & *gs*) [static]

9.75.1.3 static unsigned int cadabra::exchange::possible_singlets (Ex & , Ex::iterator) [static]

The documentation for this class was generated from the following files:

- core/Exchange.hh
- core/Exchange.cc

9.76 exec_stream_t Class Reference

```
#include <exec-stream.h>
```

Classes

- class error_t
- class next_arg_impl_t
- class next_arg_t

Public Types

- enum stream_kind_t {
  s_in =1, s_out =2, s_err =4, s_all =s_in|s_out|s_err,
  s_child =8 }
- typedef unsigned long timeout_t
- typedef unsigned long error_code_t

Public Member Functions

- exec_stream_t ()
- exec_stream_t (std::string const &program, std::string const &arguments)
- template<class iterator >
  exec_stream_t (std::string const &program, iterator args_begin, iterator args_end)
- ~exec_stream_t ()
- void set_buffer_limit (int stream_kind, std::size_t size)
- void set_wait_timeout (int stream_kind, timeout_t milliseconds)
- void set_binary_mode (int stream_kind)
- void set_text_mode (int stream_kind)
- void start (std::string const &program, std::string const &arguments)
- template<class iterator >
  void start (std::string const &program, iterator args_begin, iterator args_end)
- void start (std::string const &program, char const *arg1, char const *arg2)
- void start (std::string const &program, char *arg1, char *arg2)
- bool close_in ()
- bool close ()
- void kill ()
- int exit_code ()
- std::ostream & in ()
- std::istream & out ()
- std::istream & err ()

Private Member Functions

- exec_stream_t (exec_stream_t const &)
- exec_stream_t & operator= (exec_stream_t const &)
- void exceptions (bool enable)
- void new_impl ()
- void start (std::string const &program, next_arg_t &next_arg)

Private Attributes

- impl_t * m_impl

Friends

- struct impl_t

9.76.1 Member Typedef Documentation

9.76.1.1 typedef unsigned long exec_stream_t::error_code_t

9.76.1.2 typedef unsigned long exec_stream_t::timeout_t

9.76.2 Member Enumeration Documentation

9.76.2.1 enum exec_stream_t::stream_kind_t

Enumerator

s_in

s_out

s_err

s_all

s_child

9.76.3 Constructor & Destructor Documentation

9.76.3.1 exec_stream_t::exec_stream_t ( )

9.76.3.2 exec_stream_t::exec_stream_t (std::string const & *program,* std::string const & *arguments*)

9.76.3.3 template<class iterator > exec_stream_t::exec_stream_t (std::string const & *program,* iterator *args_begin,* iterator *args_end*) [inline]

9.76.3.4 exec_stream_t::~exec_stream_t ( )

9.76.3.5 exec_stream_t::exec_stream_t (**exec_stream_t** const &) [private]

9.76.4 Member Function Documentation

9.76.4.1 bool exec_stream_t::close ( )

9.76.4.2 bool exec_stream_t::close_in ( )

9.76.4.3 std::istream & exec_stream_t::err ( )

9.76.4.4 void exec_stream_t::exceptions (bool *enable*) [private]

9.76.4.5 int exec_stream_t::exit_code ( )

9.76.4.6 std::ostream & exec_stream_t::in ( )

9.76.4.7 void exec_stream_t::kill ( )

9.76.4.8 void exec_stream_t::new_impl ( ) [private]

9.76.4.9 **exec_stream_t**& exec_stream_t::operator= (**exec_stream_t** const &) [private]

9.76.4.10 std::istream & exec_stream_t::out ( )

9.76.4.11 void exec_stream_t::set_binary_mode (int *stream_kind*)

9.76.4.12 void exec_stream_t::set_buffer_limit (int *stream_kind,* std::size_t *size*)

9.76.4.13 void exec_stream_t::set_text_mode (int *stream_kind*)

9.76.4.14 void exec_stream_t::set_wait_timeout (int *stream_kind,* **timeout_t** *milliseconds*)

9.76.4.15 void exec_stream_t::start (std::string const & *program,* std::string const & *arguments*)

9.76.4.16 template<class iterator > void exec_stream_t::start (std::string const & *program,* iterator *args_begin,* iterator *args_end*) [inline]

9.76.4.17 void exec_stream_t::start (std::string const & *program,* char const * *arg1,* char const * *arg2*) [inline]

9.76.4.18 void exec_stream_t::start (std::string const & *program,* char * *arg1,* char * *arg2*) [inline]

9.76.4.19 void exec_stream_t::start (std::string const & *program,* next_arg_t & *next_arg*) [private]

9.76.5 Friends And Related Function Documentation

9.76.5.1 friend struct impl_t [friend]

9.76.6 Member Data Documentation

9.76.6.1 impl_t* exec_stream_t::m_impl [private]

The documentation for this class was generated from the following files:

- frontend/common/exec-stream.h
- frontend/common/exec-stream.cc

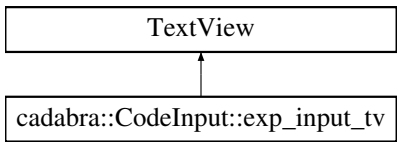
9.77 cadabra::CodeInput::exp_input_tv Class Reference

9.77.1 Description

The actual text widget used by CodeInput.

`#include <CodeInput.hh>`

Inheritance diagram for cadabra::CodeInput::exp_input_tv:



Public Member Functions

- exp_input_tv (DTree::iterator, Glib::RefPtr< Gtk::TextBuffer >, double scale)
- virtual bool on_key_press_event (GdkEventKey *) override
- virtual bool on_draw (const Cairo::RefPtr< Cairo::Context > &) override
- virtual bool on_focus_in_event (GdkEventFocus *) override
- virtual void on_show () override
- void shift_enter_pressed ()

Public Attributes

- sigc::signal1< bool, DTree::iterator > content_execute
- sigc::signal2< bool, std::string, DTree::iterator > content_changed
- sigc::signal3< bool, std::string, int, DTree::iterator > content_insert
- sigc::signal3< bool, int, int, DTree::iterator > content_erase
- sigc::signal1< bool, DTree::iterator > cell_got_focus
- friend CodeInput

Private Attributes

- double scale_
- DTree::iterator datacell

9.77.2 Constructor & Destructor Documentation

9.77.2.1 CodeInput::exp_input_tv::exp_input_tv (DTree::iterator *it,* Glib::RefPtr< Gtk::TextBuffer > *tb,* double *scale*)

9.77.3 Member Function Documentation

9.77.3.1 bool CodeInput::exp_input_tv::on_draw (const Cairo::RefPtr< Cairo::Context > & *cr*) [override], [virtual]

9.77.3.2 bool CodeInput::exp_input_tv::on_focus_in_event (GdkEventFocus * *event*) [override], [virtual]

9.77.3.3 bool CodeInput::exp_input_tv::on_key_press_event (GdkEventKey * *event*) [override], [virtual]

9.77.3.4 void CodeInput::exp_input_tv::on_show ( ) [override], [virtual]

9.77.3.5 void CodeInput::exp_input_tv::shift_enter_pressed ( )

9.77.4 Member Data Documentation

9.77.4.1 sigc::signal1<bool, DTree::iterator> cadabra::CodeInput::exp_input_tv::cell_got_focus

9.77.4.2 friend cadabra::CodeInput::exp_input_tv::CodeInput

9.77.4.3 sigc::signal2<bool, std::string, DTree::iterator> cadabra::CodeInput::exp_input_tv::content_changed

9.77.4.4 sigc::signal3<bool, int, int, DTree::iterator> cadabra::CodeInput::exp_input_tv::content_erase

9.77.4.5 sigc::signal1<bool, DTree::iterator> cadabra::CodeInput::exp_input_tv::content_execute

9.77.4.6 sigc::signal3<bool, std::string, int, DTree::iterator> cadabra::CodeInput::exp_input_tv::content_insert

9.77.4.7 DTree::iterator cadabra::CodeInput::exp_input_tv::datacell [private]

9.77.4.8 double cadabra::CodeInput::exp_input_tv::scale_ [private]

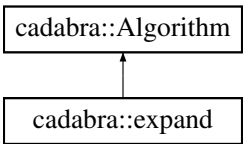
The documentation for this class was generated from the following files:

- frontend/gtkmm/CodeInput.hh
- frontend/gtkmm/CodeInput.cc

9.78 cadabra::expand Class Reference

`#include <expand.hh>`

Inheritance diagram for cadabra::expand:



Public Member Functions

- expand (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Attributes

- iterator mx_first
- iterator mx_last
- iterator ii_first
- iterator ii_last
- bool one_index

Additional Inherited Members

9.78.1 Constructor & Destructor Documentation

9.78.1.1 expand::expand (const Kernel & *k*, Ex & *e*)

9.78.2 Member Function Documentation

9.78.2.1 Algorithm::result_t expand::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.78.2.2 bool expand::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.78.3 Member Data Documentation

9.78.3.1 iterator cadabra::expand::ii_first [private]

9.78.3.2 iterator cadabra::expand::ii_last [private]

9.78.3.3 iterator cadabra::expand::mx_first [private]

9.78.3.4 iterator cadabra::expand::mx_last [private]

9.78.3.5 bool cadabra::expand::one_index [private]

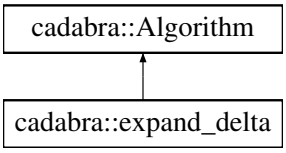
The documentation for this class was generated from the following files:

- core/algorithms/expand.hh
- core/algorithms/expand.cc

9.79 cadabra::expand_delta Class Reference

`#include <expand_delta.hh>`

Inheritance diagram for cadabra::expand_delta:



Public Member Functions

- expand_delta (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.79.1 Constructor & Destructor Documentation

9.79.1.1 expand_delta::expand_delta (const Kernel & *k*, Ex & *tr*)

9.79.2 Member Function Documentation

9.79.2.1 Algorithm::result_t expand_delta::apply (iterator & *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.79.2.2 bool expand_delta::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

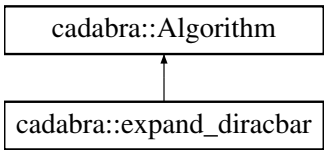
The documentation for this class was generated from the following files:

- core/algorithms/expand_delta.hh
- core/algorithms/expand_delta.cc

9.80 cadabra::expand_diracbar Class Reference

`#include <expand_diracbar.hh>`

Inheritance diagram for cadabra::expand_diracbar:



Public Member Functions

- expand_diracbar (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.80.1 Constructor & Destructor Documentation

9.80.1.1 expand_diracbar::expand_diracbar (const Kernel & *k*, Ex & *e*)

9.80.2 Member Function Documentation

9.80.2.1 Algorithm::result_t expand_diracbar::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.80.2.2 bool expand_diracbar::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

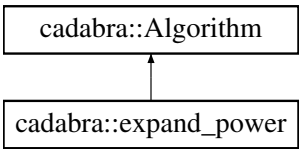
The documentation for this class was generated from the following files:

- core/algorithms/expand_diracbar.hh
- core/algorithms/expand_diracbar.cc

9.81 cadabra::expand_power Class Reference

`#include <expand_power.hh>`

Inheritance diagram for cadabra::expand_power:



Public Member Functions

- expand_power (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.81.1 Constructor & Destructor Documentation

9.81.1.1 expand_power::expand_power (const Kernel & *k*, Ex & *e*)

9.81.2 Member Function Documentation

9.81.2.1 Algorithm::result_t expand_power::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.81.2.2 bool expand_power::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

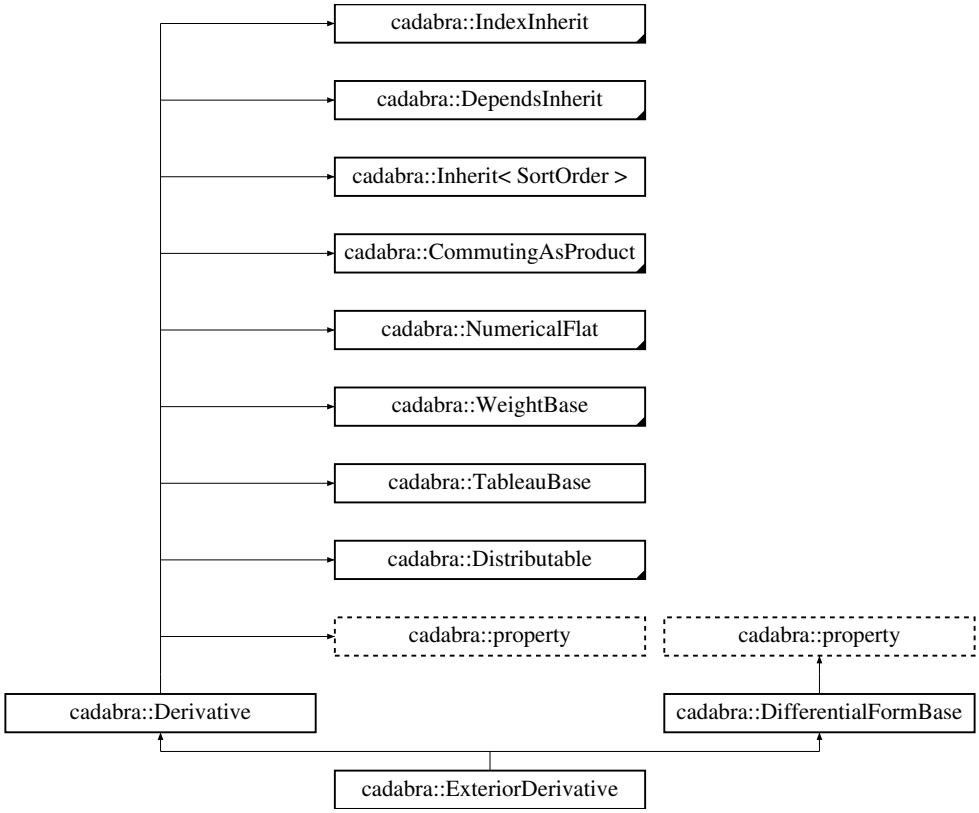
The documentation for this class was generated from the following files:

- core/algorithms/expand_power.hh
- core/algorithms/expand_power.cc

9.82 cadabra::ExteriorDerivative Class Reference

`#include <ExteriorDerivative.hh>`

Inheritance diagram for cadabra::ExteriorDerivative:



Public Member Functions

- virtual std::string name () const override
- virtual Ex degree (const Properties &, Ex::iterator) const override

Additional Inherited Members

9.82.1 Member Function Documentation

9.82.1.1 Ex ExteriorDerivative::degree (const Properties & *props*, Ex::iterator *it*) const [override], [virtual]

Implements cadabra::DifferentialFormBase.

9.82.1.2 std::string ExteriorDerivative::name ( ) const [override], [virtual]

Reimplemented from cadabra::Derivative.

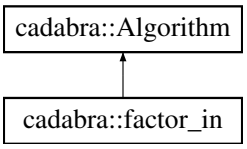
The documentation for this class was generated from the following files:

- core/properties/ExteriorDerivative.hh
- core/properties/ExteriorDerivative.cc

9.83 cadabra::factor_in Class Reference

`#include <factor_in.hh>`

Inheritance diagram for cadabra::factor_in:



Public Member Functions

- factor_in (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Protected Types

- typedef std::multimap< hashval_t, Ex::sibling_iterator > term_hash_t
- typedef term_hash_t::iterator term_hash_iterator_t

Protected Member Functions

- bool compare_restricted (iterator one, iterator two) const
- bool compare_prod_nonprod (iterator prod, iterator nonprod) const
- hashval_t calc_restricted_hash (iterator it) const
- void fill_hash_map (iterator)

Protected Attributes

- Ex & factors
- std::set< Ex, tree_exact_less_for_indexmap_obj > factnodes
- term_hash_t term_hash

Additional Inherited Members

9.83.1 Member Typedef Documentation

9.83.1.1 typedef term_hash_t::iterator cadabra::factor_in::term_hash_iterator_t [protected]

9.83.1.2 typedef std::multimap<hashval_t, Ex::sibling_iterator> cadabra::factor_in::term_hash_t [protected]

9.83.2 Constructor & Destructor Documentation

9.83.2.1 factor_in::factor_in (const Kernel & *k*, Ex & *tr*, Ex & *factors_*)

9.83.3 Member Function Documentation

9.83.3.1 Algorithm::result_t factor_in::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.83.3.2 hashval_t factor_in::calc_restricted_hash (iterator *it*) const [protected]

9.83.3.3 bool factor_in::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.83.3.4 bool factor_in::compare_prod_nonprod (iterator *prod*, iterator *nonprod*) const [protected]

9.83.3.5 bool factor_in::compare_restricted (iterator *one*, iterator *two*) const [protected]

9.83.3.6 void factor_in::fill_hash_map (iterator *it*) [protected]

9.83.4 Member Data Documentation

9.83.4.1 std::set<Ex, tree_exact_less_for_indexmap_obj> cadabra::factor_in::factnodes [protected]

9.83.4.2 Ex& cadabra::factor_in::factors [protected]

9.83.4.3 term_hash_t cadabra::factor_in::term_hash [protected]

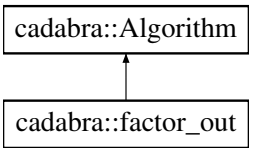
The documentation for this class was generated from the following files:

- core/algorithms/factor_in.hh
- core/algorithms/factor_in.cc

9.84 cadabra::factor_out Class Reference

`#include <factor_out.hh>`

Inheritance diagram for cadabra::factor_out:



Public Member Functions

- factor_out (const Kernel &, Ex &, Ex &, bool right)
- virtual bool can_apply (iterator) override

  Check if the expression is a sum with more than one term.
- virtual result_t apply (iterator &) override

Private Types

- typedef std::vector< Ex > to_factor_out_t

Private Attributes

- bool to_right
- to_factor_out_t to_factor_out

Additional Inherited Members

9.84.1 Member Typedef Documentation

9.84.1.1 typedef std::vector<Ex> cadabra::factor_out::to_factor_out_t [private]

9.84.2 Constructor & Destructor Documentation

9.84.2.1 factor_out::factor_out (const Kernel & *k,* Ex & *e,* Ex & *args,* bool *right*)

9.84.3 Member Function Documentation

9.84.3.1 Algorithm::result_t factor_out::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.84.3.2 bool factor_out::can_apply (iterator *st*) [override], [virtual]

Check if the expression is a sum with more than one term.

Implements cadabra::Algorithm.

9.84.4 Member Data Documentation

9.84.4.1 to_factor_out_t cadabra::factor_out::to_factor_out [private]

9.84.4.2 bool cadabra::factor_out::to_right [private]

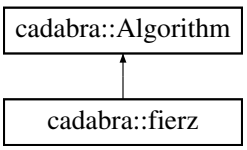
The documentation for this class was generated from the following files:

- core/algorithms/factor_out.hh
- core/algorithms/factor_out.cc

9.85 cadabra::fierz Class Reference

`#include <fierz.hh>`

Inheritance diagram for cadabra::fierz:



Public Member Functions

- fierz (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Attributes

- Ex spinor_list
- iterator spin1
- iterator spin2
- iterator spin3
- iterator spin4
- const Spinor * prop1
- const Spinor * prop2
- const Spinor * prop3
- const Spinor * prop4
- iterator gam1
- iterator gam2
- int dim
- int spinordim
- const Indices * indprop

Additional Inherited Members

9.85.1 Constructor & Destructor Documentation

9.85.1.1 fierz::fierz (const Kernel & *k,* Ex & *e,* Ex & *args*)

9.85.2 Member Function Documentation

9.85.2.1 Algorithm::result_t fierz::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.85.2.2 bool fierz::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.85.3 Member Data Documentation

9.85.3.1 int cadabra::fierz::dim [private]

9.85.3.2 iterator cadabra::fierz::gam1 [private]

9.85.3.3 iterator cadabra::fierz::gam2 [private]

9.85.3.4 const Indices* cadabra::fierz::indprop [private]

9.85.3.5 const Spinor* cadabra::fierz::prop1 [private]

9.85.3.6 const Spinor * cadabra::fierz::prop2 [private]

9.85.3.7 const Spinor * cadabra::fierz::prop3 [private]

9.85.3.8 const Spinor * cadabra::fierz::prop4 [private]

9.85.3.9 iterator cadabra::fierz::spin1 [private]

9.85.3.10 iterator cadabra::fierz::spin2 [private]

9.85.3.11 iterator cadabra::fierz::spin3 [private]

9.85.3.12 iterator cadabra::fierz::spin4 [private]

9.85.3.13 Ex cadabra::fierz::spinor_list [private]

9.85.3.14 int cadabra::fierz::spinordim [private]

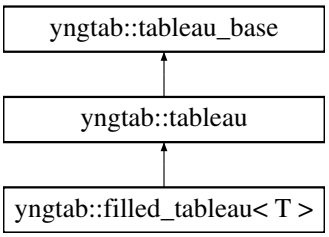
The documentation for this class was generated from the following files:

- core/algorithms/fierz.hh
- core/algorithms/fierz.cc

9.86 yngtab::filled_tableau< T > Class Template Reference

`#include <YoungTab.hh>`

Inheritance diagram for yngtab::filled_tableau< T >:



Classes

- class in_column_iterator

  An iterator which stays inside a given column of a tableau.
- class iterator

  An iterator over all boxes of a tableau, left to right, top to bottom.
- class iterator_base

Public Types

- typedef T value_type

Public Member Functions

- virtual ~filled_tableau ()
- virtual unsigned int number_of_rows () const
- virtual unsigned int row_size (unsigned int row) const
- virtual void add_box (unsigned int row)
- virtual void remove_box (unsigned int row)
- std::pair< int, int > find (const T &) const
- virtual void clear ()
- void copy_shape (const tableau &)
- T & operator() (unsigned int row, unsigned int col)
- const T & operator() (unsigned int row, unsigned int col) const
- const T & operator[ ] (unsigned int boxnum) const
- void add_box (unsigned int rownum, T val)
- void swap_columns (unsigned int c1, unsigned int c2)
- bool compare_without_multiplicity (const filled_tableau< T > &other) const
- bool has_nullifying_trace () const
- void sort_within_columns ()
- void sort_columns ()
- void canonicalise ()

  Sort equal-length columns and sort within columns.
- std::pair< int, int > nonstandard_loc () const
- template<class StrictWeakOrdering >
  void sort_within_columns (StrictWeakOrdering comp)

- template<class StrictWeakOrdering >
  void sort_columns (StrictWeakOrdering comp)
- template<class StrictWeakOrdering >
  void canonicalise (StrictWeakOrdering comp, bool only_col_ex=false)
- void projector (combin::symmetriser< T > &, bool modulo_monoterm=false) const
- void projector (combin::symmetriser< T > &, combin::range_vector_t &) const
- yngrat_t projector_normalisation () const
- filled_tableau< T > & operator= (const filled_tableau< T > &)
- in_column_iterator begin_column (unsigned int column_number)
- in_column_iterator end_column (unsigned int column_number)
- iterator begin () const
- iterator end () const
- template<class OutputIterator >
  OutputIterator Garnir_set (OutputIterator, unsigned int, unsigned int) const

Private Types

- typedef std::vector< T > box_row
- typedef std::vector< box_row > row_stack

Private Attributes

- row_stack rows

Additional Inherited Members

9.86.1 Member Typedef Documentation

9.86.1.1 template<class T> typedef std::vector<T> yngtab::filled_tableau< T >::box_row [private]

9.86.1.2 template<class T> typedef std::vector<box_row> yngtab::filled_tableau< T >::row_stack [private]

9.86.1.3 template<class T> typedef T yngtab::filled_tableau< T >::value_type

9.86.2 Constructor & Destructor Documentation

9.86.2.1 template<class T > yngtab::filled_tableau< T >::~filled_tableau ( ) [virtual]

9.86.3 Member Function Documentation

9.86.3.1 template<class T > void yngtab::filled_tableau< T >::add_box (unsigned int *row*) [virtual]

Reimplemented from yngtab::tableau.

9.86.3.2 template<class T> void yngtab::filled_tableau< T >::add_box (unsigned int *rownum,* T *val*)

9.86.3.3 template<class T > filled_tableau< T >::iterator yngtab::filled_tableau< T >::begin ( ) const

9.86.3.4 template<class T > filled_tableau< T >::in_column_iterator yngtab::filled_tableau< T >::begin_column (unsigned int *column_number*)

9.86.3.5 template<class T > void yngtab::filled_tableau< T >::canonicalise ( )

Sort equal-length columns and sort within columns.

9.86.3.6 template<class T > template<class StrictWeakOrdering > void yngtab::filled_tableau< T >::canonicalise (StrictWeakOrdering *comp,* bool *only_col_ex* = `false`)

9.86.3.7 template<class T > void yngtab::filled_tableau< T >::clear ( ) `[virtual]`

Reimplemented from yngtab::tableau.

9.86.3.8 template<class T> bool yngtab::filled_tableau< T >::compare_without_multiplicity (const filled_tableau< T > & *other*) const

9.86.3.9 template<class T > void yngtab::filled_tableau< T >::copy_shape (const tableau & *other*)

9.86.3.10 template<class T > filled_tableau< T >::iterator yngtab::filled_tableau< T >::end ( ) const

9.86.3.11 template<class T > filled_tableau< T >::in_column_iterator yngtab::filled_tableau< T >::end_column (unsigned int *column_number*)

9.86.3.12 template<class T> std::pair< int, int > yngtab::filled_tableau< T >::find (const T & *obj*) const

9.86.3.13 template<class T > template<class OutputIterator > OutputIterator yngtab::filled_tableau< T >::Garnir_set (OutputIterator *it,* unsigned int *row,* unsigned int *col*) const

9.86.3.14 template<class T > bool yngtab::filled_tableau< T >::has_nullifying_trace ( ) const

9.86.3.15 template<class T > std::pair< int, int > yngtab::filled_tableau< T >::nonstandard_loc ( ) const

9.86.3.16 template<class T > unsigned int yngtab::filled_tableau< T >::number_of_rows ( ) const `[virtual]`

Reimplemented from yngtab::tableau.

9.86.3.17 template<class T > T & yngtab::filled_tableau< T >::operator() (unsigned int *row,* unsigned int *col*)

9.86.3.18 template<class T > const T & yngtab::filled_tableau< T >::operator() (unsigned int *row,* unsigned int *col*) const

9.86.3.19 template<class T> filled_tableau< T > & yngtab::filled_tableau< T >::operator= (const filled_tableau< T > & *other*)

9.86.3.20 template<class T > const T & yngtab::filled_tableau< T >::operator[ ] (unsigned int *boxnum*) const

9.86.3.21 template<class T> void yngtab::filled_tableau< T >::projector (combin::symmetriser< T > & *sym,* bool *modulo_monoterm* = `false`) const

9.86.3.22 template<class T> void yngtab::filled_tableau< T >::projector (combin::symmetriser< T > & *sym,* combin::range_vector_t & *sublengths_scattered*) const

anti-symmetrise over boxes in columns

9.86.3.23 template<class T > yngrat_t yngtab::filled_tableau< T >::projector_normalisation ( ) const

9.86.3.24 template<class T > void yngtab::filled_tableau< T >::remove_box (unsigned int *row*) `[virtual]`

Reimplemented from yngtab::tableau.

9.86.3.25 template<class T > unsigned int yngtab::filled_tableau< T >::row_size (unsigned int *row*) const `[virtual]`

Reimplemented from yngtab::tableau.

9.86.3.26 template<class T > void yngtab::filled_tableau< T >::sort_columns ( )

9.86.3.27 template<class T > template<class StrictWeakOrdering > void yngtab::filled_tableau< T >::sort_columns (StrictWeakOrdering *comp*)

9.86.3.28 template<class T > void yngtab::filled_tableau< T >::sort_within_columns ( )

9.86.3.29 template<class T > template<class StrictWeakOrdering > void yngtab::filled_tableau< T >::sort_within_columns (StrictWeakOrdering *comp*)

9.86.3.30 template<class T > void yngtab::filled_tableau< T >::swap_columns (unsigned int *c1,* unsigned int *c2*)

9.86.4 Member Data Documentation

9.86.4.1 template<class T> row_stack yngtab::filled_tableau< T >::rows `[private]`

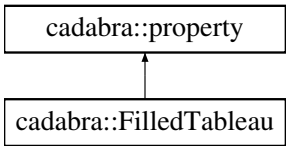
The documentation for this class was generated from the following file:

- core/YoungTab.hh

9.87 cadabra::FilledTableau Class Reference

`#include <FilledTableau.hh>`

Inheritance diagram for cadabra::FilledTableau:



Public Member Functions

- virtual ~FilledTableau ()
- virtual std::string name () const
- virtual bool parse (Ex &, Ex::iterator, Ex::iterator, keyval_t &)

Public Attributes

- int dimension

Additional Inherited Members

9.87.1 Constructor & Destructor Documentation

9.87.1.1 virtual cadabra::FilledTableau::~FilledTableau ( ) [inline], [virtual]

9.87.2 Member Function Documentation

9.87.2.1 std::string FilledTableau::name ( ) const [virtual]

Implements cadabra::property.

9.87.2.2 bool FilledTableau::parse (Ex & *tr,* Ex::iterator *pat,* Ex::iterator *prop,* keyval_t & *keyvals*) [virtual]

9.87.3 Member Data Documentation

9.87.3.1 int cadabra::FilledTableau::dimension

The documentation for this class was generated from the following files:

- core/properties/FilledTableau.hh
- core/properties/FilledTableau.cc

9.88 cadabra::str_node::flag_t Struct Reference

`#include <Storage.hh>`

Public Attributes

- bool keep_after_eval: 1
- bracket_t bracket: 3
- parent_rel_t parent_rel: 3
- bool line_per_node: 1

9.88.1 Member Data Documentation

9.88.1.1 bracket_t cadabra::str_node::flag_t::bracket

9.88.1.2 bool cadabra::str_node::flag_t::keep_after_eval

9.88.1.3 bool cadabra::str_node::flag_t::line_per_node

9.88.1.4 parent_rel_t cadabra::str_node::flag_t::parent_rel

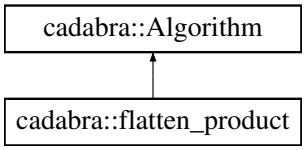
The documentation for this struct was generated from the following file:

- core/Storage.hh

9.89 cadabra::flatten_product Class Reference

`#include <flatten_product.hh>`

Inheritance diagram for cadabra::flatten_product:



Public Member Functions

- flatten_product (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Public Attributes

- bool make_consistent_only
- bool is_diff

Additional Inherited Members

9.89.1 Constructor & Destructor Documentation

9.89.1.1 flatten_product::flatten_product (const Kernel & *k*, Ex & *tr*)

9.89.2 Member Function Documentation

9.89.2.1 Algorithm::result_t flatten_product::apply (iterator & *it*) [virtual]

Implements cadabra::Algorithm.

9.89.2.2 bool flatten_product::can_apply (iterator *it*) [virtual]

Implements cadabra::Algorithm.

9.89.3 Member Data Documentation

9.89.3.1 bool cadabra::flatten_product::is_diff

9.89.3.2 bool cadabra::flatten_product::make_consistent_only

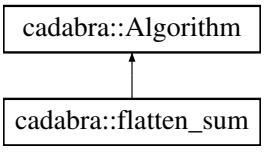
The documentation for this class was generated from the following files:

- core/algorithms/flatten_product.hh
- core/algorithms/flatten_product.cc

9.90 cadabra::flatten_sum Class Reference

```
#include <flatten_sum.hh>
```

Inheritance diagram for cadabra::flatten_sum:



Public Member Functions

- flatten_sum (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Public Attributes

- bool make_consistent_only

Additional Inherited Members

9.90.1 Constructor & Destructor Documentation

9.90.1.1 flatten_sum::flatten_sum (const Kernel & *k,* Ex & *tr*)

9.90.2 Member Function Documentation

9.90.2.1 Algorithm::result_t flatten_sum::apply (iterator & *it*) [virtual]

Implements cadabra::Algorithm.

9.90.2.2 bool flatten_sum::can_apply (iterator *it*) [virtual]

Implements cadabra::Algorithm.

9.90.3 Member Data Documentation

9.90.3.1 bool cadabra::flatten_sum::make_consistent_only

The documentation for this class was generated from the following files:

- core/algorithms/flatten_sum.hh
- core/algorithms/flatten_sum.cc

9.91 snoop::Flush Class Reference

```
#include <Snoop.hh>
```

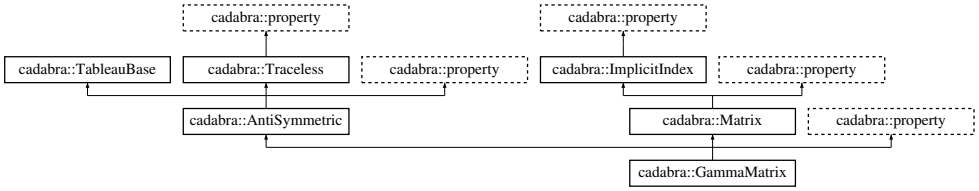
The documentation for this class was generated from the following file:

- client_server/Snoop.hh

9.92 cadabra::GammaMatrix Class Reference

`#include <GammaMatrix.hh>`

Inheritance diagram for cadabra::GammaMatrix:



Public Member Functions

- virtual std::string name () const override
- virtual void latex (std::ostream &) const override

  Display the property on the stream.
- virtual bool parse (const Kernel &, keyval_t &keyvals) override

Public Attributes

- Ex metric

Additional Inherited Members

9.92.1 Member Function Documentation

9.92.1.1 void GammaMatrix::latex (std::ostream & *str*) const [override], [virtual]

Display the property on the stream.

Generate a LaTeX representation of the property, assuming LaTeX is in text mode (so it needs dollar symbols to switch to maths).

Reimplemented from cadabra::property.

9.92.1.2 std::string GammaMatrix::name ( ) const [override], [virtual]

Implements cadabra::property.

9.92.1.3 bool GammaMatrix::parse (const Kernel & *kernel,* keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

9.92.2 Member Data Documentation

9.92.2.1 Ex cadabra::GammaMatrix::metric

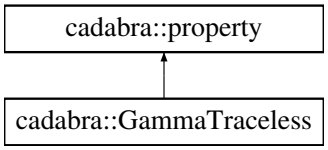
The documentation for this class was generated from the following files:

- core/properties/GammaMatrix.hh
- core/properties/GammaMatrix.cc

9.93 cadabra::GammaTraceless Class Reference

`#include <GammaTraceless.hh>`

Inheritance diagram for cadabra::GammaTraceless:



Public Member Functions

- virtual ~GammaTraceless ()
- virtual std::string name () const

Additional Inherited Members

9.93.1 Constructor & Destructor Documentation

9.93.1.1 virtual cadabra::GammaTraceless::~GammaTraceless ( ) [inline], [virtual]

9.93.2 Member Function Documentation

9.93.2.1 std::string GammaTraceless::name ( ) const [virtual]

Implements cadabra::property.

The documentation for this class was generated from the following files:

- core/properties/GammaTraceless.hh
- core/properties/GammaTraceless.cc

9.94 cadabra::GUIBase Class Reference

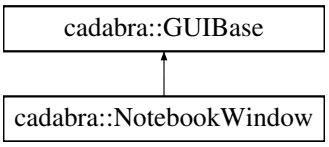
9.94.1 Description

Abstract base class with methods that need to be implemented by any GUI.

You need to derive from this class as well as from the DocumentThread class.

`#include <GUIBase.hh>`

Inheritance diagram for cadabra::GUIBase:



Public Member Functions

- virtual void update_cell (const DTree &, DTree::iterator)=0

  The basic manipulations that a GUI needs to implement are adding, removing and updating (refreshing the display of) a cell.
- virtual void remove_cell (const DTree &, DTree::iterator)=0

  Remove a single cell.
- virtual void remove_all_cells ()=0

  Remove all GUI cells from the display (used as a quick way to clear all before loading a new document).
- virtual void add_cell (const DTree &, DTree::iterator, bool visible)=0

  Add a GUI cell corresponding to the document cell at the iterator.
- virtual void position_cursor (const DTree &, DTree::iterator, int)=0

  Position the cursor in the current canvas in the widget corresponding to the indicated cell.
- virtual size_t get_cursor_position (const DTree &, DTree::iterator)=0

  Retrieve the position of the cursor in the current cell.
- virtual void process_data ()=0

  When the ComputeThread needs to modify the document, it stores an ActionBase object on the stack (see the DocumenThread class) and then wakes up the GUI thread signalling it to process this action.

- virtual void on_connect ()=0

  Network status is propagated from the ComputeThread to the.
- virtual void on_disconnect (const std::string &reason)=0
- virtual void on_network_error ()=0
- virtual void on_kernel_runstatus (bool)=0

9.94.2 Member Function Documentation

9.94.2.1 virtual void cadabra::GUIBase::add_cell (const DTree & , DTree::iterator , bool *visible*) [pure virtual]

Add a GUI cell corresponding to the document cell at the iterator.

The GUI needs to figure out from the location of this cell in the DTree where to insert the cell in the visual display. If the 'visible' flag is false, hide the cell from view independent of whether its hidden flag is set (this is only used when constructing a document on load time and we do not want to show cells until they have all been added to the document).

Implemented in cadabra::NotebookWindow.

9.94.2.2 virtual size_t cadabra::GUIBase::get_cursor_position (const DTree & , DTree::iterator) [pure virtual]

Retrieve the position of the cursor in the current cell.

Implemented in cadabra::NotebookWindow.

9.94.2.3 virtual void cadabra::GUIBase::on_connect ( ) [pure virtual]

Network status is propagated from the ComputeThread to the.

GUI using the following methods. These get called on the compute thread (as opposed to the functions above, which get called on the gui thread).

Implemented in cadabra::NotebookWindow.

9.94.2.4 virtual void cadabra::GUIBase::on_disconnect (const std::string & *reason*) [pure virtual]

Implemented in cadabra::NotebookWindow.

9.94.2.5 virtual void cadabra::GUIBase::on_kernel_runstatus (bool) [pure virtual]

Implemented in cadabra::NotebookWindow.

9.94.2.6 virtual void cadabra::GUIBase::on_network_error ( ) [pure virtual]

Implemented in cadabra::NotebookWindow.

9.94.2.7 virtual void cadabra::GUIBase::position_cursor (const DTree & , DTree::iterator , int) [pure virtual]

Position the cursor in the current canvas in the widget corresponding to the indicated cell.

Implemented in cadabra::NotebookWindow.

9.94.2.8 virtual void cadabra::GUIBase::process_data ( ) [pure virtual]

When the ComputeThread needs to modify the document, it stores an ActionBase object on the stack (see the DocumenThread class) and then wakes up the GUI thread signalling it to process this action.

The following member should wake up the GUI thread and make it enter the processing part.

Implemented in cadabra::NotebookWindow.

9.94.2.9 virtual void cadabra::GUIBase::remove_all_cells ( ) [pure virtual]

Remove all GUI cells from the display (used as a quick way to clear all before loading a new document).

Implemented in cadabra::NotebookWindow.

9.94.2.10 virtual void cadabra::GUIBase::remove_cell (const DTree & , DTree::iterator) [pure virtual]

Remove a single cell.

Implemented in cadabra::NotebookWindow.

9.94.2.11 virtual void cadabra::GUIBase::update_cell (const DTree & , DTree::iterator) [pure virtual]

The basic manipulations that a GUI needs to implement are adding, removing and updating (refreshing the display of) a cell.

The code in DocumentThread will call these to make the GUI update its display. Called on the document thread.

Implemented in cadabra::NotebookWindow.

The documentation for this class was generated from the following file:

- client_server/GUIBase.hh

9.95 cadabra::DataCell::id_t Class Reference

9.95.1 Description

Each cell is identified by a serial number 'id' which is used to keep track of it across network calls, and a bool indicating whether the client or the server has created this cell.

`#include <DataCell.hh>`

Public Member Functions

- id_t ()
- bool operator< (const id_t &other) const

Public Attributes

- uint64_t id
- bool created_by_client

9.95.2 Constructor & Destructor Documentation

9.95.2.1 DataCell::id_t::id_t ( )

9.95.3 Member Function Documentation

9.95.3.1 bool DataCell::id_t::operator< (const id_t & *other*) const

9.95.4 Member Data Documentation

9.95.4.1 bool cadabra::DataCell::id_t::created_by_client

9.95.4.2 uint64_t cadabra::DataCell::id_t::id

The documentation for this class was generated from the following files:

- client_server/DataCell.hh
- client_server/DataCell.cc

9.96 cadabra::exchange::identical_tensors_t Struct Reference

`#include <Exchange.hh>`

Public Attributes

- unsigned int number_of_indices
- std::vector< Ex::sibling_iterator > tensors
- std::vector< int > seq_numbers_of_first_indices
- const SelfCommutingBehaviour * comm
- const Spinor * spino
- const TableauBase * tab
- const Traceless * traceless
- const GammaTraceless * gammatraceless
- int extra_sign

9.96.1 Member Data Documentation

9.96.1.1 const SelfCommutingBehaviour* cadabra::exchange::identical_tensors_t::comm

9.96.1.2 int cadabra::exchange::identical_tensors_t::extra_sign

9.96.1.3 const GammaTraceless* cadabra::exchange::identical_tensors_t::gammatraceless

9.96.1.4 unsigned int cadabra::exchange::identical_tensors_t::number_of_indices

9.96.1.5 std::vector<int> cadabra::exchange::identical_tensors_t::seq_numbers_of_first_indices

9.96.1.6 const Spinor* cadabra::exchange::identical_tensors_t::spino

9.96.1.7 const TableauBase* cadabra::exchange::identical_tensors_t::tab

9.96.1.8 std::vector<Ex::sibling_iterator> cadabra::exchange::identical_tensors_t::tensors

9.96.1.9 const Traceless* cadabra::exchange::identical_tensors_t::traceless

The documentation for this struct was generated from the following file:

- core/Exchange.hh

9.97 Images Class Reference

`#include <Permutations.hh>`

Public Member Functions

- template<class iterator >
  void find (iterator start1, iterator end1, iterator start2, iterator end2)
  Find the permutation that takes [start1, end1> to [start2, end2>.

Public Attributes

- std::vector< int > images

9.97.1 Member Function Documentation

9.97.1.1 template<class iterator > void Images::find (iterator *start1,* iterator *end1,* iterator *start2,* iterator *end2*)

Find the permutation that takes [start1, end1> to [start2, end2>.

9.97.2 Member Data Documentation

9.97.2.1 std::vector<int> Images::images

The documentation for this class was generated from the following file:

- core/Permutations.hh

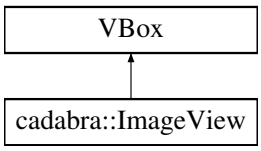
9.98 cadabra::ImageView Class Reference

9.98.1 Description

An image viewing widget.

`#include <ImageView.hh>`

Inheritance diagram for cadabra::ImageView:



Public Member Functions

- ImageView ()
- virtual ~ImageView ()
- void set_image_from_base64 (const std::string &b64)

Private Attributes

- Gtk::Image image

9.98.2 Constructor & Destructor Documentation

9.98.2.1 ImageView::ImageView ( )

9.98.2.2 ImageView::~ImageView ( ) [virtual]

9.98.3 Member Function Documentation

9.98.3.1 void ImageView::set_image_from_base64 (const std::string & *b64*)

9.98.4 Member Data Documentation

9.98.4.1 Gtk::Image cadabra::ImageView::image [private]

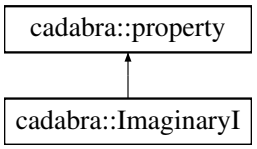
The documentation for this class was generated from the following files:

- frontend/gtkmm/ImageView.hh
- frontend/gtkmm/ImageView.cc

9.99 cadabra::ImaginaryI Class Reference

`#include <ImaginaryI.hh>`

Inheritance diagram for cadabra::ImaginaryI:



Public Member Functions

- virtual std::string name () const override

Additional Inherited Members

9.99.1 Member Function Documentation

9.99.1.1 std::string ImaginaryI::name ( ) const [override], [virtual]

Implements cadabra::property.

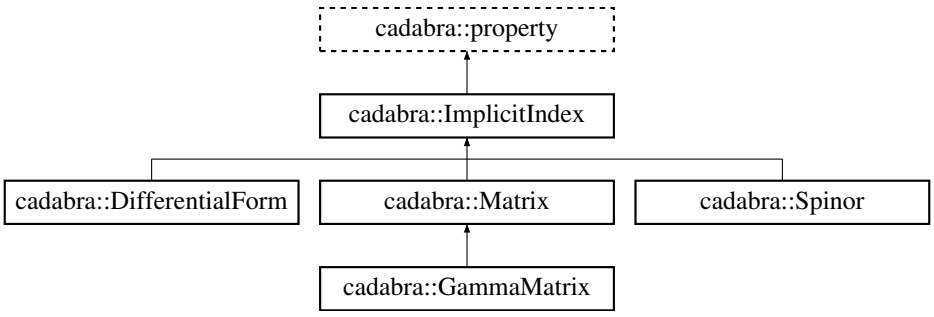
The documentation for this class was generated from the following files:

- core/properties/ImaginaryI.hh
- core/properties/ImaginaryI.cc

9.100 cadabra::ImplicitIndex Class Reference

`#include <ImplicitIndex.hh>`

Inheritance diagram for cadabra::ImplicitIndex:



Public Member Functions

- virtual bool parse (const Kernel &, keyval_t &) override
- virtual std::string name () const override
- virtual std::string unnamed_argument () const override
- virtual void latex (std::ostream &str) const override

  Display the property on the stream.

Public Attributes

- std::vector< std::string > set_names

Additional Inherited Members

9.100.1 Member Function Documentation

9.100.1.1 void ImplicitIndex::latex (std::ostream & *str*) const [override], [virtual]

Display the property on the stream.

Generate a LaTeX representation of the property, assuming LaTeX is in text mode (so it needs dollar symbols to switch to maths).

Reimplemented from cadabra::property.

Reimplemented in cadabra::GammaMatrix.

9.100.1.2 std::string ImplicitIndex::name ( ) const [override], [virtual]

Implements cadabra::property.

Reimplemented in cadabra::DifferentialForm, cadabra::Spinor, cadabra::GammaMatrix, and cadabra::Matrix.

9.100.1.3 bool ImplicitIndex::parse (const Kernel & , keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

Reimplemented in cadabra::DifferentialForm, cadabra::Spinor, and cadabra::GammaMatrix.

9.100.1.4 virtual std::string cadabra::ImplicitIndex::unnamed_argument ( ) const [inline], [override], [virtual]

Reimplemented from cadabra::property.

9.100.2 Member Data Documentation

9.100.2.1 std::vector<std::string> cadabra::ImplicitIndex::set_names

The documentation for this class was generated from the following files:

- core/properties/ImplicitIndex.hh
- core/properties/ImplicitIndex.cc

9.101 yngtab::filled_tableau< T >::in_column_iterator Class Reference

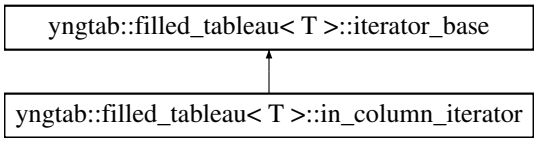
9.101.1 Description

template<class T>
class yngtab::filled_tableau< T >::in_column_iterator

An iterator which stays inside a given column of a tableau.

`#include <YoungTab.hh>`

Inheritance diagram for yngtab::filled_tableau< T >::in_column_iterator:



Public Member Functions

- in_column_iterator (unsigned int r, unsigned int c, filled_tableau< T > *)
- T & operator* () const
- T * operator-> () const
- in_column_iterator & operator++ ()
- in_column_iterator operator++ (int)
- in_column_iterator & operator-- ()
- in_column_iterator operator-- (int)
- in_column_iterator operator+ (unsigned int)
- in_column_iterator operator- (unsigned int)
- in_column_iterator & operator+= (unsigned int)
- in_column_iterator & operator-= (unsigned int)
- bool operator< (const in_column_iterator &other) const
- bool operator> (const in_column_iterator &other) const
- bool operator<= (const in_column_iterator &other) const
- bool operator>= (const in_column_iterator &other) const
- ptrdiff_t operator- (const in_column_iterator &) const
- bool operator== (const in_column_iterator &) const
- bool operator!= (const in_column_iterator &) const

Private Attributes

- filled_tableau< T > * tab
- unsigned int column_number
- unsigned int row_number

Friends

- class filled_tableau< T >

Additional Inherited Members

9.101.2 Constructor & Destructor Documentation

9.101.2.1 template<class T> **yngtab::filled_tableau< T >::in_column_iterator::in_column_iterator** (unsigned int *r,* unsigned int *c,* **filled_tableau< T > *** *t*)

9.101.3 Member Function Documentation

9.101.3.1 template<class T > bool **yngtab::filled_tableau< T >::in_column_iterator::operator!=** (const **in_column_iterator** & *other*) const

9.101.3.2 template<class T > T & **yngtab::filled_tableau< T >::in_column_iterator::operator*** ( ) const

9.101.3.3 template<class T > **filled_tableau< T >::in_column_iterator yngtab::filled_tableau< T >::in_column_iterator::operator+** (unsigned int *n*)

9.101.3.4 template<class T > **filled_tableau< T >::in_column_iterator & yngtab::filled_tableau< T >::in_column_iterator::operator++** ( )

9.101.3.5 template<class T > **filled_tableau< T >::in_column_iterator yngtab::filled_tableau< T >::in_column_iterator::operator++** (int)

9.101.3.6 template<class T > **filled_tableau< T >::in_column_iterator & yngtab::filled_tableau< T >::in_column_iterator::operator+=** (unsigned int *n*)

9.101.3.7 template<class T > **filled_tableau< T >::in_column_iterator yngtab::filled_tableau< T >::in_column_iterator::operator-** (unsigned int *n*)

9.101.3.8 template<class T > ptrdiff_t **yngtab::filled_tableau< T >::in_column_iterator::operator-** (const **in_column_iterator** & *other*) const

9.101.3.9 template<class T > **filled_tableau< T >::in_column_iterator & yngtab::filled_tableau< T >::in_column_iterator::operator--** ( )

9.101.3.10 template<class T > **filled_tableau< T >::in_column_iterator yngtab::filled_tableau< T >::in_column_iterator::operator--** (int)

9.101.3.11 template<class T > filled_tableau< T >::in_column_iterator & yngtab::filled_tableau< T >::in_column_iterator::operator-= (unsigned int *n*)

9.101.3.12 template<class T > T * yngtab::filled_tableau< T >::in_column_iterator::operator-> ( ) const

9.101.3.13 template<class T > bool yngtab::filled_tableau< T >::in_column_iterator::operator< (const in_column_iterator & *other*) const

9.101.3.14 template<class T > bool yngtab::filled_tableau< T >::in_column_iterator::operator<= (const in_column_iterator & *other*) const

9.101.3.15 template<class T > bool yngtab::filled_tableau< T >::in_column_iterator::operator== (const in_column_iterator & *other*) const

9.101.3.16 template<class T > bool yngtab::filled_tableau< T >::in_column_iterator::operator> (const in_column_iterator & *other*) const

9.101.3.17 template<class T > bool yngtab::filled_tableau< T >::in_column_iterator::operator>= (const in_column_iterator & *other*) const

9.101.4 Friends And Related Function Documentation

9.101.4.1 template<class T> friend class filled_tableau< T > [friend]

9.101.5 Member Data Documentation

9.101.5.1 template<class T> unsigned int yngtab::filled_tableau< T >::in_column_iterator::column_number [private]

9.101.5.2 template<class T> unsigned int yngtab::filled_tableau< T >::in_column_iterator::row_number [private]

9.101.5.3 template<class T> filled_tableau<T>* yngtab::filled_tableau< T >::in_column_iterator::tab [private]

The documentation for this class was generated from the following file:

- core/YoungTab.hh

9.102 cadabra::index_iterator Class Reference

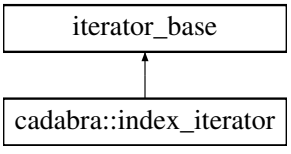
9.102.1 Description

An iterator which iterates over indices even if they are at lower levels, i.e.

taking into account the "Inherit" property of nodes. Needs access to Properties in the current scope in order to determine which objects are indices.

`#include <IndexIterator.hh>`

Inheritance diagram for cadabra::index_iterator:



Public Member Functions

- index_iterator (const Properties &)
- index_iterator (const index_iterator &)
- index_iterator & operator= (const index_iterator &)
- bool operator== (const index_iterator &) const
- bool operator!= (const index_iterator &) const
- index_iterator & operator++ ()
- index_iterator operator++ (int)
- index_iterator & operator+= (unsigned int)

Static Public Member Functions

- static index_iterator create (const Properties &, const iterator_base &)
- static index_iterator begin (const Properties &, const iterator_base &, int offset=0)
- static index_iterator end (const Properties &, const iterator_base &)

Public Attributes

- Ex::iterator halt
- Ex::iterator walk
- Ex::iterator roof

Private Member Functions

- bool is_index (Ex::iterator) const

Private Attributes

- const Properties * properties

9.102.2 Constructor & Destructor Documentation

9.102.2.1 index_iterator::index_iterator (const Properties & *k*)

9.102.2.2 index_iterator::index_iterator (const index_iterator & *other*)

9.102.3 Member Function Documentation

9.102.3.1 index_iterator index_iterator::begin (const Properties & *properties,* const iterator_base & *it,* int *offset* = `0`) `[static]`

9.102.3.2 index_iterator index_iterator::create (const Properties & *k,* const iterator_base & *other*) `[static]`

9.102.3.3 index_iterator index_iterator::end (const Properties & *properties,* const iterator_base & *it*) `[static]`

9.102.3.4 bool cadabra::index_iterator::is_index (Ex::iterator) const [private]

9.102.3.5 bool index_iterator::operator!= (const index_iterator & *other*) const

9.102.3.6 index_iterator & index_iterator::operator++ ( )

9.102.3.7 index_iterator cadabra::index_iterator::operator++ (int)

9.102.3.8 index_iterator & index_iterator::operator+= (unsigned int *num*)

9.102.3.9 index_iterator & index_iterator::operator= (const index_iterator & *other*)

9.102.3.10 bool index_iterator::operator== (const index_iterator & *other*) const

9.102.4 Member Data Documentation

9.102.4.1 Ex::iterator cadabra::index_iterator::halt

9.102.4.2 const Properties* cadabra::index_iterator::properties [private]

9.102.4.3 Ex::iterator cadabra::index_iterator::roof

9.102.4.4 Ex::iterator cadabra::index_iterator::walk

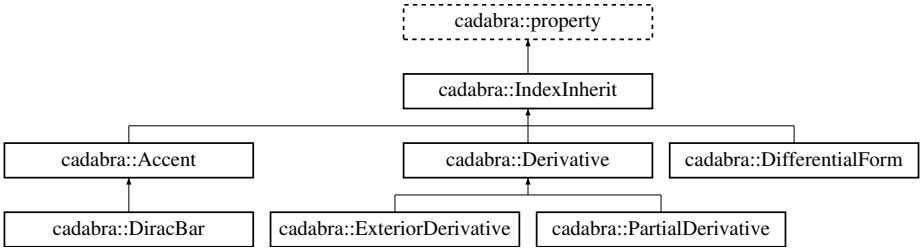
The documentation for this class was generated from the following files:

- core/IndexIterator.hh
- core/IndexIterator.cc

9.103 cadabra::IndexInherit Class Reference

`#include <IndexInherit.hh>`

Inheritance diagram for cadabra::IndexInherit:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.103.1 Member Function Documentation

9.103.1.1 virtual std::string cadabra::IndexInherit::name ( ) const [inline], [virtual]

Implements cadabra::property.

Reimplemented in cadabra::Accent, cadabra::Derivative, cadabra::DifferentialForm, cadabra::ExteriorDerivative, cadabra::DiracBar, and cadabra::PartialDerivative.

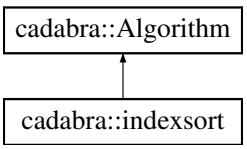
The documentation for this class was generated from the following file:

- core/properties/IndexInherit.hh

9.104 cadabra::indexsort Class Reference

`#include <indexsort.hh>`

Inheritance diagram for cadabra::indexsort:



Classes

- class less_indexed_treenode

Public Member Functions

- indexsort (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Private Attributes

- const TableauBase * tb

Additional Inherited Members

9.104.1 Constructor & Destructor Documentation

9.104.1.1 indexsort::indexsort (const Kernel & *k*, Ex & *tr*)

9.104.2 Member Function Documentation

9.104.2.1 Algorithm::result_t indexsort::apply (iterator & *st*) [virtual]

Implements cadabra::Algorithm.

9.104.2.2 bool indexsort::can_apply (iterator *st*) [virtual]

Implements cadabra::Algorithm.

9.104.3 Member Data Documentation

9.104.3.1 const TableauBase* cadabra::indexsort::tb [private]

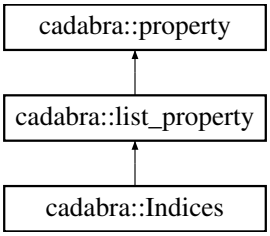
The documentation for this class was generated from the following files:

- core/algorithms/indexsort.hh
- core/algorithms/indexsort.cc

9.105 cadabra::Indices Class Reference

`#include <Indices.hh>`

Inheritance diagram for cadabra::Indices:



Public Types

- enum position_t { free, fixed, independent }

Public Member Functions

- Indices ()
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual std::string name () const override
- virtual std::string unnamed_argument () const override
- virtual match_t equals (const property *) const override
- virtual void latex (std::ostream &) const override

  Display the property on the stream.

Public Attributes

- std::string set_name
- std::string parent_name
- enum cadabra::Indices::position_t position_type
- std::vector< Ex > values

Private Member Functions

- void collect_index_values (Ex::iterator ind_values)

  Given the right-hand side of a 'values={...}' node, generate a list of all index values in index_values.

9.105.1 Member Enumeration Documentation

9.105.1.1 enum cadabra::Indices::position_t

Enumerator

free

fixed

independent

9.105.2 Constructor & Destructor Documentation

9.105.2.1 Indices::Indices ( )

9.105.3 Member Function Documentation

9.105.3.1 void Indices::collect_index_values (Ex::iterator *ind_values*) [private]

Given the right-hand side of a 'values={...}' node, generate a list of all index values in index_values.

9.105.3.2 property::match_t Indices::equals (const property * *other*) const [override], [virtual]

Reimplemented from cadabra::property.

9.105.3.3 void Indices::latex (std::ostream & *str*) const [override], [virtual]

Display the property on the stream.

Generate a LaTeX representation of the property, assuming LaTeX is in text mode (so it needs dollar symbols to switch to maths).

Reimplemented from cadabra::property.

9.105.3.4 std::string Indices::name ( ) const [override], [virtual]

Implements cadabra::property.

9.105.3.5 bool Indices::parse (const Kernel & , keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

9.105.3.6 virtual std::string cadabra::Indices::unnamed_argument ( ) const [inline], [override], [virtual]

Reimplemented from cadabra::property.

9.105.4 Member Data Documentation

9.105.4.1 std::string cadabra::Indices::parent_name

9.105.4.2 enum cadabra::Indices::position_t cadabra::Indices::position_type

9.105.4.3 std::string cadabra::Indices::set_name

9.105.4.4 std::vector<Ex> cadabra::Indices::values

The documentation for this class was generated from the following files:

- core/properties/Indices.hh
- core/properties/Indices.cc

9.106 cadabra::Inherit< T > Class Template Reference

9.106.1 Description

template<class T>
class cadabra::Inherit< T >

If a property X derives from Inherit<Y>, and get<Y> is called on an object which has an X property (but no Y property), then the get<Y> will look at the non-index child of the object to see if that has a Y property.

FIXME: need to decided what to do if there are more non-index children.

```
#include <Props.hh>
```

Public Member Functions

- virtual ~Inherit ()
- virtual std::string name () const

9.106.2 Constructor & Destructor Documentation

9.106.2.1 template<class T> virtual cadabra::Inherit< T >::~Inherit ( ) [inline], [virtual]

9.106.3 Member Function Documentation

9.106.3.1 template<class T> virtual std::string cadabra::Inherit< T >::name ( ) const [inline], [virtual]

Reimplemented in cadabra::Derivative, cadabra::ExteriorDerivative, and cadabra::PartialDerivative.

The documentation for this class was generated from the following file:

- core/Props.hh

9.107 cadabra::Integer Class Reference

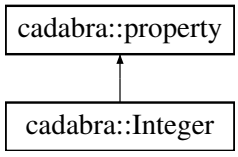
9.107.1 Description

Property indicating that a symbolic object always takes integer values.

Optionally takes a range over which it runs, which can be symbolic.

`#include <Integer.hh>`

Inheritance diagram for cadabra::Integer:



Public Member Functions

- virtual ~Integer ()
- virtual std::string name () const override
- virtual bool parse (const Kernel &, keyval_t &keyvals) override
- virtual void display (std::ostream &) const
- virtual std::string unnamed_argument () const override

Public Attributes

- Ex from
- Ex to
- Ex difference

Additional Inherited Members

9.107.2 Constructor & Destructor Documentation

9.107.2.1 virtual cadabra::Integer::~Integer ( ) [inline], [virtual]

9.107.3 Member Function Documentation

9.107.3.1 void Integer::display (std::ostream & *str*) const [virtual]

9.107.3.2 std::string Integer::name ( ) const [override], [virtual]

Implements cadabra::property.

9.107.3.3 bool Integer::parse (const Kernel & *kernel,* keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

9.107.3.4 virtual std::string cadabra::Integer::unnamed_argument ( ) const [inline], [override], [virtual]

Reimplemented from cadabra::property.

9.107.4 Member Data Documentation

9.107.4.1 Ex cadabra::Integer::difference

9.107.4.2 Ex cadabra::Integer::from

9.107.4.3 Ex cadabra::Integer::to

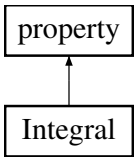
The documentation for this class was generated from the following files:

- core/properties/Integer.hh
- core/properties/Integer.cc

9.108 Integral Class Reference

`#include <Integral.hh>`

Inheritance diagram for Integral:



Public Member Functions

- virtual ~Integral ()
- virtual std::string name () const
- virtual bool parse (const Properties &, keyval_t &keyvals) override
- virtual void display (std::ostream &) const

9.108.1 Constructor & Destructor Documentation

9.108.1.1 virtual Integral::~Integral ( ) [inline], [virtual]

9.108.2 Member Function Documentation

9.108.2.1 virtual void Integral::display (std::ostream &) const [virtual]

9.108.2.2 virtual std::string Integral::name ( ) const [virtual]

9.108.2.3 virtual bool Integral::parse (const Properties & , keyval_t & *keyvals*) [override], [virtual]

The documentation for this class was generated from the following file:

- core/properties/Integral.hh

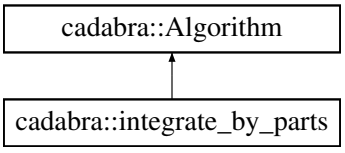
9.109 cadabra::integrate_by_parts Class Reference

9.109.1 Description

Integrate by parts away from the indicated derivative object.

`#include <integrate_by_parts.hh>`

Inheritance diagram for cadabra::integrate_by_parts:



Public Member Functions

- integrate_by_parts (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Member Functions

- result_t handle_term (iterator, iterator &)
- bool int_and_derivative_related (iterator int_it, iterator der_it) const
- Ex wrap (iterator, sibling_iterator, sibling_iterator) const
- bool derivative_acting_on_arg (iterator der_it) const

Private Attributes

- Ex away_from

Additional Inherited Members

9.109.2 Constructor & Destructor Documentation

9.109.2.1 integrate_by_parts::integrate_by_parts (const Kernel & *k,* Ex & *tr,* Ex & *af*)

9.109.3 Member Function Documentation

9.109.3.1 Algorithm::result_t integrate_by_parts::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.109.3.2 bool integrate_by_parts::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.109.3.3 bool integrate_by_parts::derivative_acting_on_arg (iterator *der_it*) const [private]

9.109.3.4 Algorithm::result_t integrate_by_parts::handle_term (iterator *int_it,* iterator & *it*) [private]

9.109.3.5 bool integrate_by_parts::int_and_derivative_related (iterator *int_it,* iterator *der_it*) const [private]

9.109.3.6 Ex integrate_by_parts::wrap (iterator *prod,* sibling_iterator *from,* sibling_iterator *to*) const [private]

9.109.4 Member Data Documentation

9.109.4.1 Ex cadabra::integrate_by_parts::away_from [private]

The documentation for this class was generated from the following files:

- core/algorithms/integrate_by_parts.hh
- core/algorithms/integrate_by_parts.cc

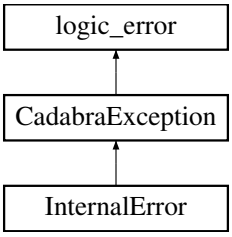
9.110 InternalError Class Reference

9.110.1 Description

Exception thrown when there the Cadabra code detects an internal inconsistency; this should never happen, but is better than bailing out with an assert.

`#include <Exceptions.hh>`

Inheritance diagram for InternalError:



Public Member Functions

- InternalError (std::string="")
- std::string py_what () const

9.110.2 Constructor & Destructor Documentation

9.110.2.1 InternalError::InternalError (std::string *s* = " ")

9.110.3 Member Function Documentation

9.110.3.1 std::string InternalError::py_what ( ) const

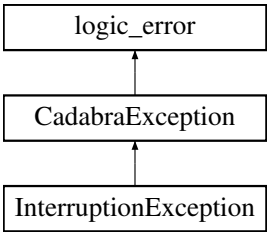
The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

9.111 InterruptionException Class Reference

`#include <Exceptions.hh>`

Inheritance diagram for InterruptionException:



Public Member Functions

- InterruptionException (std::string="")

9.111.1 Constructor & Destructor Documentation

9.111.1.1 InterruptionException::InterruptionException (std::string *s* = " ")

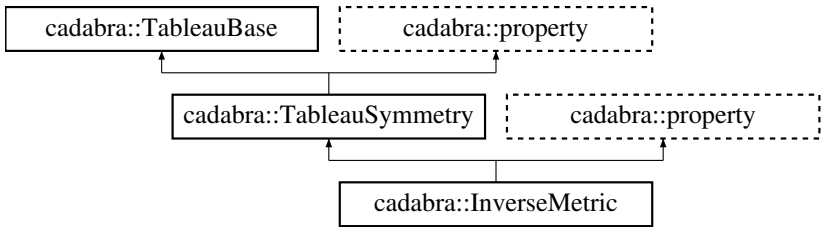
The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

9.112 cadabra::InverseMetric Class Reference

`#include <InverseMetric.hh>`

Inheritance diagram for cadabra::InverseMetric:



Public Member Functions

- InverseMetric ()
- virtual std::string name () const override
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual void validate (const Kernel &, const Ex &) const override

Public Attributes

- int signature

Additional Inherited Members

9.112.1 Constructor & Destructor Documentation

9.112.1.1 InverseMetric::InverseMetric ( )

9.112.2 Member Function Documentation

9.112.2.1 std::string InverseMetric::name ( ) const [override], [virtual]

Implements cadabra::property.

9.112.2.2 bool InverseMetric::parse (const Kernel & , keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

9.112.2.3 void InverseMetric::validate (const Kernel & , const Ex & *tr*) const [override], [virtual]

Reimplemented from cadabra::property.

9.112.3 Member Data Documentation

9.112.3.1 int cadabra::InverseMetric::signature

The documentation for this class was generated from the following files:

- core/properties/InverseMetric.hh
- core/properties/InverseMetric.cc

9.113 iterable_converter Class Reference

Public Member Functions

- template<typename Container >
  iterable_converter & from_python ()

Static Public Member Functions

- static void * convertible (PyObject *object)
- template<typename Container >
  static void construct (PyObject *object, boost::python::converter::rvalue_from_python_stage1_data *data)

9.113.1 Member Function Documentation

9.113.1.1 template<typename Container > static void iterable_converter::construct (PyObject * *object*, boost::python::converter::rvalue_from_python_stage1_data * *data*) [inline], [static]

9.113.1.2 static void* iterable_converter::convertible (PyObject * *object*) [inline], [static]

9.113.1.3 template<typename Container > iterable_converter& iterable_converter::from_python ( ) [inline]

The documentation for this class was generated from the following file:

- core/PythonCdb.cc

9.114 yngtab::filled_tableau< T >::iterator Class Reference

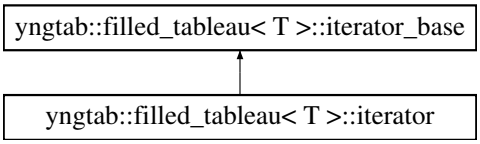
9.114.1 Description

template<class T>
class yngtab::filled_tableau< T >::iterator

An iterator over all boxes of a tableau, left to right, top to bottom.

`#include <YoungTab.hh>`

Inheritance diagram for yngtab::filled_tableau< T >::iterator:



Public Member Functions

- iterator (unsigned int r, unsigned int c, filled_tableau< T > *)
- T & operator* () const
- T * operator-> () const
- iterator & operator++ ()
- iterator operator++ (int)
- iterator & operator-- ()
- iterator operator-- (int)
- iterator operator+ (unsigned int)
- iterator operator- (unsigned int)
- iterator & operator+= (unsigned int)
- iterator & operator-= (unsigned int)
- bool operator< (const iterator &other) const
- bool operator> (const iterator &other) const
- ptrdiff_t operator- (const iterator &) const
- bool operator== (const iterator &) const
- bool operator!= (const iterator &) const

Private Attributes

- filled_tableau< T > * tab
- unsigned int column_number
- unsigned int row_number

Friends

- class filled_tableau< T >

Additional Inherited Members

9.114.2 Constructor & Destructor Documentation

9.114.2.1 template<class T> yngtab::filled_tableau< T >::iterator::iterator (unsigned int *r*, unsigned int *c*, filled_tableau< T > * *t*)

9.114.3 Member Function Documentation

9.114.3.1 template<class T > bool yngtab::filled_tableau< T >::iterator::operator!= (const iterator & *other*) const

9.114.3.2 template<class T > T & yngtab::filled_tableau< T >::iterator::operator* ( ) const

9.114.3.3 template<class T > filled_tableau< T >::iterator yngtab::filled_tableau< T >::iterator::operator+ (unsigned int *n*)

9.114.3.4 template<class T > filled_tableau< T >::iterator & yngtab::filled_tableau< T >::iterator::operator++ ( )

9.114.3.5 template<class T > filled_tableau< T >::iterator yngtab::filled_tableau< T >::iterator::operator++ (int)

9.114.3.6 template<class T > filled_tableau< T >::iterator & yngtab::filled_tableau< T >::iterator::operator+= (unsigned int *n*)

9.114.3.7 template<class T > filled_tableau< T >::iterator yngtab::filled_tableau< T >::iterator::operator- (unsigned int *n*)

9.114.3.8 template<class T > ptrdiff_t yngtab::filled_tableau< T >::iterator::operator- (const iterator & *other*) const

9.114.3.9 template<class T > filled_tableau< T >::iterator & yngtab::filled_tableau< T >::iterator::operator-- ( )

9.114.3.10 template<class T > filled_tableau< T >::iterator yngtab::filled_tableau< T >::iterator::operator-- (int)

9.114.3.11 template<class T > filled_tableau< T >::iterator & yngtab::filled_tableau< T >::iterator::operator-= (unsigned int *n*)

9.114.3.12 template<class T > T * yngtab::filled_tableau< T >::iterator::operator-> ( ) const

9.114.3.13 template<class T > bool yngtab::filled_tableau< T >::iterator::operator< (const iterator & *other*) const

9.114.3.14 template<class T > bool yngtab::filled_tableau< T >::iterator::operator== (const iterator & *other*) const

9.114.3.15 template<class T > bool yngtab::filled_tableau< T >::iterator::operator> (const iterator & *other*) const

9.114.4 Friends And Related Function Documentation

9.114.4.1 template<class T> friend class filled_tableau< T > [friend]

9.114.5 Member Data Documentation

9.114.5.1 template<class T> unsigned int yngtab::filled_tableau< T >::iterator::column_number [private]

9.114.5.2 template<class T> unsigned int yngtab::filled_tableau< T >::iterator::row_number [private]

9.114.5.3 template<class T> filled_tableau<T>* yngtab::filled_tableau< T >::iterator::tab [private]

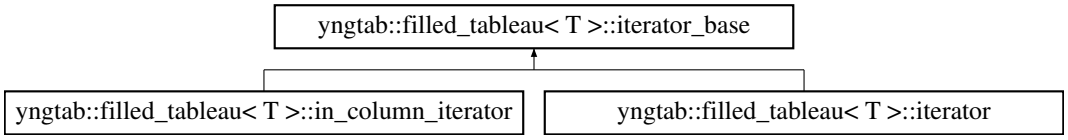
The documentation for this class was generated from the following file:

- core/YoungTab.hh

9.115 yngtab::filled_tableau< T >::iterator_base Class Reference

`#include <YoungTab.hh>`

Inheritance diagram for yngtab::filled_tableau< T >::iterator_base:



Public Types

- typedef T value_type
- typedef T * pointer
- typedef T & reference
- typedef size_t size_type
- typedef ptrdiff_t difference_type
- typedef std::random_access_iterator_tag iterator_category

9.115.1 Member Typedef Documentation

9.115.1.1 template<class T> typedef ptrdiff_t yngtab::filled_tableau< T >::iterator_base::difference_type

9.115.1.2 template<class T> typedef std::random_access_iterator_tag yngtab::filled_tableau< T >::iterator_base::iterator_category

9.115.1.3 template<class T> typedef T* yngtab::filled_tableau< T >::iterator_base::pointer

9.115.1.4 template<class T> typedef T& yngtab::filled_tableau< T >::iterator_base::reference

9.115.1.5 template<class T> typedef size_t yngtab::filled_tableau< T >::iterator_base::size_type

9.115.1.6 template<class T> typedef T yngtab::filled_tableau< T >::iterator_base::value_type

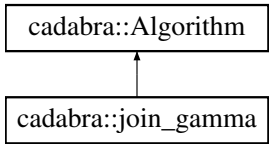
The documentation for this class was generated from the following file:

- core/YoungTab.hh

9.116 cadabra::join_gamma Class Reference

`#include <join_gamma.hh>`

Inheritance diagram for cadabra::join_gamma:



Public Member Functions

- join_gamma (const Kernel &, Ex &, bool expand, bool use_gendelta)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Public Attributes

- bool expand
- std::vector< int > only_expand
- const GammaMatrix * gm1
- const GammaMatrix * gm2

Private Member Functions

- void regroup_indices_ (sibling_iterator, sibling_iterator, unsigned int, std::vector< Ex > &, std::vector< Ex > &)
- void append_prod_ (const std::vector< Ex > &r1, const std::vector< Ex > &r2, unsigned int num1, unsigned int num2, unsigned int i, multiplier_t mult, Ex &rep, iterator loc)

Private Attributes

- bool use_generalised_delta_
- Ex::iterator gamma_name_
- str_node::bracket_t gamma_bracket_

Additional Inherited Members

9.116.1 Constructor & Destructor Documentation

9.116.1.1 join_gamma::join_gamma (const Kernel & *kernel,* Ex & *tr_,* bool *expand,* bool *use_gendelta*)

9.116.2 Member Function Documentation

9.116.2.1 void join_gamma::append_prod_ (const std::vector< Ex > & *r1,* const std::vector< Ex > & *r2,* unsigned int *num1,* unsigned int *num2,* unsigned int *i,* multiplier_t *mult,* Ex & *rep,* iterator *loc*) [private]

9.116.2.2 Algorithm::result_t join_gamma::apply (iterator & *st*) [virtual]

Implements cadabra::Algorithm.

9.116.2.3 bool join_gamma::can_apply (iterator *st*) [virtual]

Implements cadabra::Algorithm.

9.116.2.4 void join_gamma::regroup_indices_ (sibling_iterator *gam1*, sibling_iterator *gam2*, unsigned int *i*, std::vector< Ex > & *r1*, std::vector< Ex > & *r2*) [private]

9.116.3 Member Data Documentation

9.116.3.1 bool cadabra::join_gamma::expand

9.116.3.2 str_node::bracket_t cadabra::join_gamma::gamma_bracket_ [private]

9.116.3.3 Ex::iterator cadabra::join_gamma::gamma_name_ [private]

9.116.3.4 const GammaMatrix* cadabra::join_gamma::gm1

9.116.3.5 const GammaMatrix * cadabra::join_gamma::gm2

9.116.3.6 std::vector<int> cadabra::join_gamma::only_expand

9.116.3.7 bool cadabra::join_gamma::use_generalised_delta_ [private]

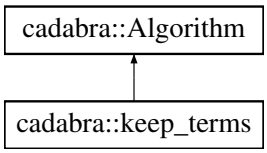
The documentation for this class was generated from the following files:

- core/algorithms/join_gamma.hh
- core/algorithms/join_gamma.cc

9.117 cadabra::keep_terms Class Reference

`#include <keep_terms.hh>`

Inheritance diagram for cadabra::keep_terms:



Public Member Functions

- keep_terms (const Kernel &, Ex &, std::vector< int > terms)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Private Attributes

- std::vector< int > terms_

Additional Inherited Members

9.117.1 Constructor & Destructor Documentation

9.117.1.1 keep_terms::keep_terms (const Kernel & *k,* Ex & *tr,* std::vector< int > *terms*)

9.117.2 Member Function Documentation

9.117.2.1 Algorithm::result_t keep_terms::apply (iterator & *it*) [virtual]

Implements cadabra::Algorithm.

9.117.2.2 bool keep_terms::can_apply (iterator *it*) [virtual]

Implements cadabra::Algorithm.

9.117.3 Member Data Documentation

9.117.3.1 std::vector<int> cadabra::keep_terms::terms_ [private]

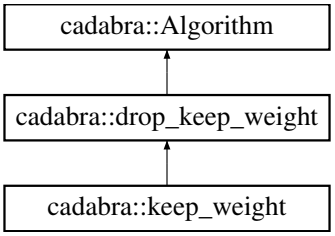
The documentation for this class was generated from the following files:

- core/algorithms/keep_terms.hh
- core/algorithms/keep_terms.cc

9.118 cadabra::keep_weight Class Reference

`#include <drop_weight.hh>`

Inheritance diagram for cadabra::keep_weight:



Public Member Functions

- keep_weight (const Kernel &, Ex &, Ex &)
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.118.1 Constructor & Destructor Documentation

9.118.1.1 keep_weight::keep_weight (const Kernel & *k*, Ex & *e*, Ex & *a*)

9.118.2 Member Function Documentation

9.118.2.1 Algorithm::result_t keep_weight::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

The documentation for this class was generated from the following files:

- core/algorithms/drop_weight.hh
- core/algorithms/drop_weight.cc

9.119 cadabra::Kernel Class Reference

`#include <Kernel.hh>`

Public Member Functions

- Kernel ()
- Kernel (const Kernel &other)=delete
- ~Kernel ()
- void inject_property (property *prop, std::shared_ptr< Ex > pattern, std::shared_ptr< Ex > property_↩ arguments)

  Inject a property into the system and attach it to the given pattern.
- std::shared_ptr< Ex > ex_from_string (const std::string &)

  Create an Ex expression object from a string, which will be parsed.

Public Attributes

- Properties properties

9.119.1 Constructor & Destructor Documentation

9.119.1.1 Kernel::Kernel ( )

9.119.1.2 cadabra::Kernel::Kernel (const Kernel & *other*) [delete]

9.119.1.3 Kernel::~Kernel ( )

9.119.2 Member Function Documentation

9.119.2.1 std::shared_ptr< cadabra::Ex > Kernel::ex_from_string (const std::string & *s*)

Create an Ex expression object from a string, which will be parsed.

9.119.2.2 void Kernel::inject_property (property * *prop*, std::shared_ptr< Ex > *pattern*, std::shared_ptr< Ex > *property_arguments*)

Inject a property into the system and attach it to the given pattern.

9.119.3 Member Data Documentation

9.119.3.1 Properties cadabra::Kernel::properties

The documentation for this class was generated from the following files:

- core/Kernel.hh
- core/Kernel.cc

9.120 cadabra::keyval_t Class Reference

9.120.1 Description

Arguments to properties get parsed into a keyval_t structure.

```
#include <Props.hh>
```

Public Types

- typedef std::pair< std::string, Ex::iterator > kvpair_t
- typedef std::list< kvpair_t > kvlist_t
- typedef kvlist_t::const_iterator const_iterator
- typedef kvlist_t::iterator iterator
- typedef kvpair_t value_type

Public Member Functions

- const_iterator find (const std::string &) const
- iterator find (const std::string &)
- const_iterator begin () const
- const_iterator end () const
- void push_back (const kvpair_t &)
- void erase (iterator)

Private Attributes

- kvlist_t keyvals

9.120.2 Member Typedef Documentation

9.120.2.1 typedef kvlist_t::const_iterator cadabra::keyval_t::const_iterator

9.120.2.2 typedef kvlist_t::iterator cadabra::keyval_t::iterator

9.120.2.3 typedef std::list<kvpair_t> cadabra::keyval_t::kvlist_t

9.120.2.4 typedef std::pair<std::string, Ex::iterator> cadabra::keyval_t::kvpair_t

9.120.2.5 typedef kvpair_t cadabra::keyval_t::value_type

9.120.3 Member Function Documentation

9.120.3.1 keyval_t::const_iterator keyval_t::begin ( ) const

9.120.3.2 keyval_t::const_iterator keyval_t::end ( ) const

9.120.3.3 void keyval_t::erase (iterator *it*)

9.120.3.4 keyval_t::const_iterator keyval_t::find (const std::string & *key*) const

9.120.3.5 keyval_t::iterator keyval_t::find (const std::string & *key*)

9.120.3.6 void keyval_t::push_back (const kvpair_t & *kv*)

9.120.4 Member Data Documentation

9.120.4.1 kvlist_t cadabra::keyval_t::keyvals [private]

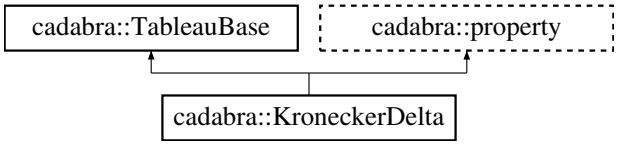
The documentation for this class was generated from the following files:

- core/Props.hh
- core/Props.cc

9.121 cadabra::KroneckerDelta Class Reference

`#include <KroneckerDelta.hh>`

Inheritance diagram for cadabra::KroneckerDelta:



Public Member Functions

- virtual ~KroneckerDelta ()
- virtual std::string name () const
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const

Additional Inherited Members

9.121.1 Constructor & Destructor Documentation

9.121.1.1 virtual cadabra::KroneckerDelta::~KroneckerDelta ( ) [inline], [virtual]

9.121.2 Member Function Documentation

9.121.2.1 TableauBase::tab_t KroneckerDelta::get_tab (const Properties & *properties,* Ex & *tr,* Ex::iterator *it,* unsigned int *num*) const [virtual]

Implements cadabra::TableauBase.

9.121.2.2 std::string KroneckerDelta::name ( ) const [virtual]

Implements cadabra::property.

9.121.2.3 unsigned int KroneckerDelta::size (const Properties & , Ex & , Ex::iterator) const [virtual]

Implements cadabra::TableauBase.

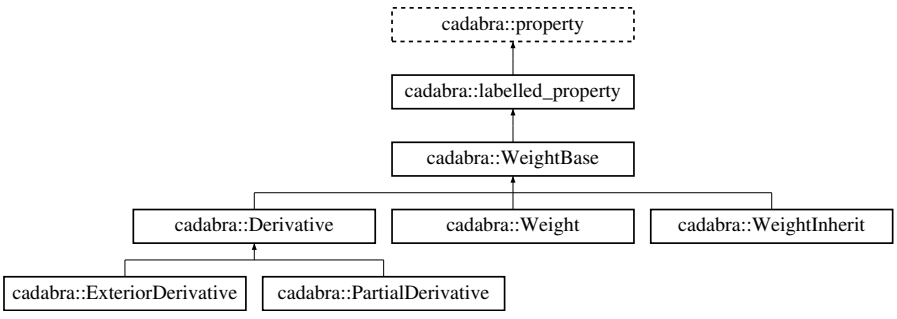
The documentation for this class was generated from the following files:

- core/properties/KroneckerDelta.hh
- core/properties/KroneckerDelta.cc

9.122 cadabra::labelled_property Class Reference

`#include <Props.hh>`

Inheritance diagram for cadabra::labelled_property:



Public Member Functions

- virtual bool parse (const Kernel &, keyval_t &) override

Public Attributes

- std::string label

Additional Inherited Members

9.122.1 Member Function Documentation

9.122.1.1 bool labelled_property::parse (const Kernel & , keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

Reimplemented in cadabra::WeightInherit, and cadabra::Weight.

9.122.2 Member Data Documentation

9.122.2.1 std::string cadabra::labelled_property::label

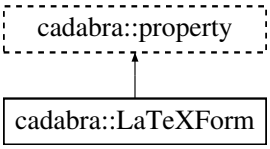
The documentation for this class was generated from the following files:

- core/Props.hh
- core/Props.cc

9.123 cadabra::LaTeXForm Class Reference

`#include <LaTeXForm.hh>`

Inheritance diagram for cadabra::LaTeXForm:



Public Member Functions

- virtual std::string name () const override
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual std::string unnamed_argument () const override
- std::string latex_form () const

Private Attributes

- std::string latex_

Additional Inherited Members

9.123.1 Member Function Documentation

9.123.1.1 std::string LaTeXForm::latex_form ( ) const

9.123.1.2 std::string LaTeXForm::name ( ) const [override], [virtual]

Implements cadabra::property.

9.123.1.3 bool LaTeXForm::parse (const Kernel & , keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

9.123.1.4 std::string LaTeXForm::unnamed_argument ( ) const [override], [virtual]

Reimplemented from cadabra::property.

9.123.2 Member Data Documentation

9.123.2.1 std::string cadabra::LaTeXForm::latex_ [private]

The documentation for this class was generated from the following files:

- core/properties/LaTeXForm.hh
- core/properties/LaTeXForm.cc

9.124 cadabra::indexsort::less_indexed_treenode Class Reference

`#include <indexsort.hh>`

Public Member Functions

- less_indexed_treenode (const Kernel &, Ex &, iterator it)
- bool operator() (unsigned int, unsigned int) const

Private Attributes

- const Kernel & kernel
- Ex & tr
- Ex::iterator it

9.124.1 Constructor & Destructor Documentation

9.124.1.1 indexsort::less_indexed_treenode::less_indexed_treenode (const Kernel & *k*, Ex & *t*, iterator *it*)

9.124.2 Member Function Documentation

9.124.2.1 bool indexsort::less_indexed_treenode::operator() (unsigned int *i1*, unsigned int *i2*) const

9.124.3 Member Data Documentation

9.124.3.1 Ex::iterator cadabra::indexsort::less_indexed_treenode::it [private]

9.124.3.2 const Kernel& cadabra::indexsort::less_indexed_treenode::kernel [private]

9.124.3.3 Ex& cadabra::indexsort::less_indexed_treenode::tr [private]

The documentation for this class was generated from the following files:

- core/algorithms/indexsort.hh
- core/algorithms/indexsort.cc

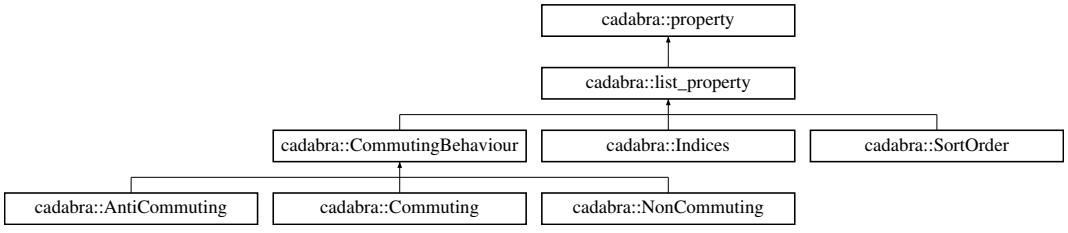
9.125 cadabra::list_property Class Reference

9.125.1 Description

Something cannot be both a list property and a normal property at the same time, so we can safely inherit without virtual.

```
#include <Props.hh>
```

Inheritance diagram for cadabra::list_property:



Additional Inherited Members

The documentation for this class was generated from the following file:

- core/Props.hh

9.126 snoop::Snoop::LogEntry Class Reference

9.126.1 Description

C++ representation of a log entry.

`#include <Snoop.hh>`

Public Member Functions

- LogEntry ()
- LogEntry (int log_id_, int client_log_id_, int id_, const std::string &, uint64_t, uint64_t, const std::string &, int, const std::string &, const std::string &, const std::string &, int status)
- std::string to_json (bool human_readable) const
- void from_json (const Json::Value &)

Public Attributes

- int log_id
- int client_log_id
- int id
- std::string uuid
- uint64_t create_millis
- uint64_t receive_millis
- std::string loc_file
- int loc_line
- std::string loc_method
- std::string type
- std::string message
- int server_status

9.126.2 Constructor & Destructor Documentation

9.126.2.1 Snoop::LogEntry::LogEntry ( )

9.126.2.2 Snoop::LogEntry::LogEntry (int *log_id_*, int *client_log_id_*, int *id_*, const std::string & *c1b*, uint64_t *c2*, uint64_t *c2b*, const std::string & *c3*, int *c4*, const std::string & *c5*, const std::string & *c6*, const std::string & *c7*, int *status*)

9.126.3 Member Function Documentation

9.126.3.1 void Snoop::LogEntry::from_json (const Json::Value & *entry*)

9.126.3.2 std::string Snoop::LogEntry::to_json (bool *human_readable*) const

9.126.4 Member Data Documentation

9.126.4.1 int snoop::Snoop::LogEntry::client_log_id

9.126.4.2 uint64_t snoop::Snoop::LogEntry::create_millis

9.126.4.3 int snoop::Snoop::LogEntry::id

9.126.4.4 std::string snoop::Snoop::LogEntry::loc_file

9.126.4.5 int snoop::Snoop::LogEntry::loc_line

9.126.4.6 std::string snoop::Snoop::LogEntry::loc_method

9.126.4.7 int snoop::Snoop::LogEntry::log_id

9.126.4.8 std::string snoop::Snoop::LogEntry::message

9.126.4.9 uint64_t snoop::Snoop::LogEntry::receive_millis

9.126.4.10 int snoop::Snoop::LogEntry::server_status

9.126.4.11 std::string snoop::Snoop::LogEntry::type

9.126.4.12 std::string snoop::Snoop::LogEntry::uuid

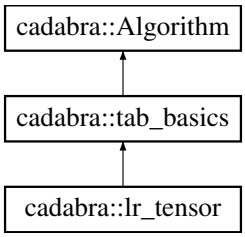
The documentation for this class was generated from the following files:

- client_server/Snoop.hh
- client_server/Snoop.cc

9.127 cadabra::lr_tensor Class Reference

`#include <lr_tensor.hh>`

Inheritance diagram for cadabra::lr_tensor:



Public Member Functions

- lr_tensor (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Public Attributes

- sibling_iterator tab1
- sibling_iterator tab2

Private Member Functions

- void do_tableau (iterator &)
- void do_filledtableau (iterator &)

Additional Inherited Members

9.127.1 Constructor & Destructor Documentation

9.127.1.1 lr_tensor::lr_tensor (const Kernel & *k,* Ex & *tr*)

9.127.2 Member Function Documentation

9.127.2.1 Algorithm::result_t lr_tensor::apply (iterator & *it*) `[override], [virtual]`

Implements cadabra::Algorithm.

9.127.2.2 bool lr_tensor::can_apply (iterator *it*) `[override], [virtual]`

Implements cadabra::Algorithm.

9.127.2.3 void lr_tensor::do_filledtableau (iterator & *it*) [private]

9.127.2.4 void lr_tensor::do_tableau (iterator & *it*) [private]

9.127.3 Member Data Documentation

9.127.3.1 sibling_iterator cadabra::lr_tensor::tab1

9.127.3.2 sibling_iterator cadabra::lr_tensor::tab2

The documentation for this class was generated from the following files:

- core/algorithms/lr_tensor.hh
- core/algorithms/lr_tensor.cc

9.128 cadabra::map_sympy Class Reference

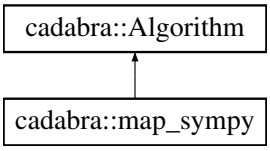
9.128.1 Description

Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place.

This is a higher-level thing than 'sympy::apply' in the SympyCdb.cc module.

```
#include <map_sympy.hh>
```

Inheritance diagram for cadabra::map_sympy:



Public Member Functions

- map_sympy (const Kernel &, Ex &, const std::string &head)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Private Attributes

- std::string head_
- std::vector< Ex::iterator > left
- std::set< Ex::iterator > index_factors

Additional Inherited Members

9.128.2 Constructor & Destructor Documentation

9.128.2.1 map_sympy::map_sympy (const Kernel & *k*, Ex & *tr*, const std::string & *head*)

9.128.3 Member Function Documentation

9.128.3.1 Algorithm::result_t map_sympy::apply (iterator & *it*) [virtual]

Implements cadabra::Algorithm.

9.128.3.2 bool map_sympy::can_apply (iterator *st*) [virtual]

Implements cadabra::Algorithm.

9.128.4 Member Data Documentation

9.128.4.1 std::string cadabra::map_sympy::head_ [private]

9.128.4.2 std::set<Ex::iterator> cadabra::map_sympy::index_factors [private]

9.128.4.3 std::vector<Ex::iterator> cadabra::map_sympy::left [private]

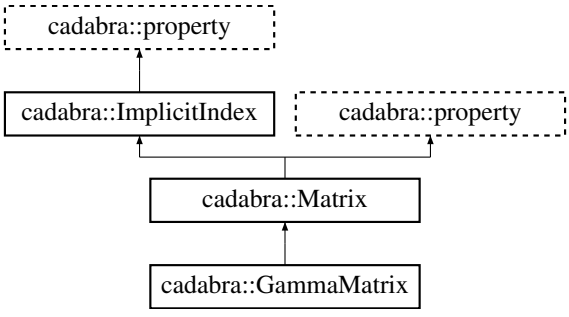
The documentation for this class was generated from the following files:

- core/algorithms/map_sympy.hh
- core/algorithms/map_sympy.cc

9.129 cadabra::Matrix Class Reference

`#include <Matrix.hh>`

Inheritance diagram for cadabra::Matrix:



Public Member Functions

- virtual ~Matrix ()
- virtual std::string name () const

Additional Inherited Members

9.129.1 Constructor & Destructor Documentation

9.129.1.1 virtual cadabra::Matrix::~Matrix ( ) [inline], [virtual]

9.129.2 Member Function Documentation

9.129.2.1 std::string Matrix::name ( ) const [virtual]

Implements cadabra::property.

Reimplemented in cadabra::GammaMatrix.

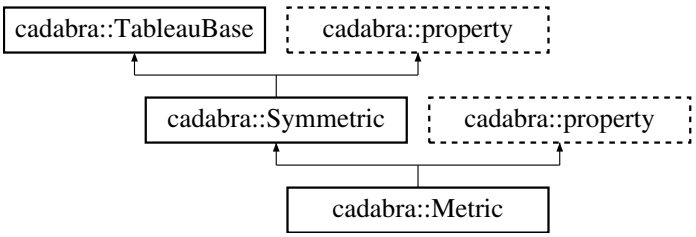
The documentation for this class was generated from the following files:

- core/properties/Matrix.hh
- core/properties/Matrix.cc

9.130 cadabra::Metric Class Reference

`#include <Metric.hh>`

Inheritance diagram for cadabra::Metric:



Public Member Functions

- Metric ()
- virtual std::string name () const override
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual void validate (const Kernel &, const Ex &) const override
- virtual void latex (std::ostream &) const override

  Display the property on the stream.

Public Attributes

- int signature

Additional Inherited Members

9.130.1 Constructor & Destructor Documentation

9.130.1.1 Metric::Metric ( )

9.130.2 Member Function Documentation

9.130.2.1 void Metric::latex (std::ostream & *str*) const [override], [virtual]

Display the property on the stream.

Generate a LaTeX representation of the property, assuming LaTeX is in text mode (so it needs dollar symbols to switch to maths).

Reimplemented from cadabra::property.

9.130.2.2 std::string Metric::name ( ) const [override], [virtual]

Implements cadabra::property.

9.130.2.3 bool Metric::parse (const Kernel & , keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

9.130.2.4 void Metric::validate (const Kernel & , const Ex & *tr*) const [override], [virtual]

Reimplemented from cadabra::property.

9.130.3 Member Data Documentation

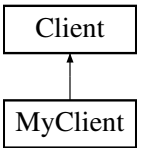
9.130.3.1 int cadabra::Metric::signature

The documentation for this class was generated from the following files:

- core/properties/Metric.hh
- core/properties/Metric.cc

9.131 MyClient Class Reference

Inheritance diagram for MyClient:



Public Member Functions

- MyClient ()
- void on_connect ()
- void on_disconnect ()
- void on_network_error ()
- void on_progress ()
- void before_tree_change (ActionBase &)
- void after_tree_change (ActionBase &)

9.131.1 Constructor & Destructor Documentation

9.131.1.1 MyClient::MyClient ( )

9.131.2 Member Function Documentation

9.131.2.1 void MyClient::after_tree_change (ActionBase & *ab*)

9.131.2.2 void MyClient::before_tree_change (ActionBase & *ab*)

9.131.2.3 void MyClient::on_connect ( )

9.131.2.4 void MyClient::on_disconnect ( )

9.131.2.5 void MyClient::on_network_error ( )

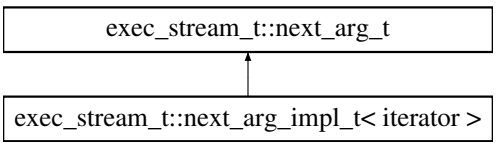
9.131.2.6 void MyClient::on_progress ( )

The documentation for this class was generated from the following file:

- client_server/test_client.cc

9.132 exec_stream_t::next_arg_impl_t< iterator > Class Template Reference

Inheritance diagram for exec_stream_t::next_arg_impl_t< iterator >:



Public Member Functions

- next_arg_impl_t (iterator args_begin, iterator args_end)
- virtual std::string const * next ()

Private Attributes

- iterator m_args_i
- iterator m_args_end
- std::string m_arg

9.132.1 Constructor & Destructor Documentation

9.132.1.1 template<class iterator> exec_stream_t::next_arg_impl_t< iterator >::next_arg_impl_t (iterator *args_begin,* iterator *args_end*) [inline]

9.132.2 Member Function Documentation

9.132.2.1 template<class iterator> virtual std::string const* exec_stream_t::next_arg_impl_t< iterator >::next ( ) [inline], [virtual]

Implements exec_stream_t::next_arg_t.

9.132.3 Member Data Documentation

9.132.3.1 template<class iterator> std::string exec_stream_t::next_arg_impl_t< iterator >::m_arg [private]

9.132.3.2 template<class iterator> iterator exec_stream_t::next_arg_impl_t< iterator >::m_args_end [private]

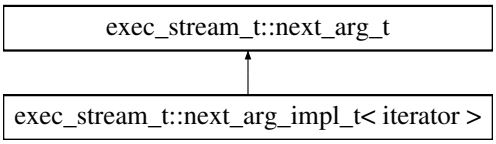
9.132.3.3 template<class iterator> iterator exec_stream_t::next_arg_impl_t< iterator >::m_args_i [private]

The documentation for this class was generated from the following file:

- frontend/common/exec-stream.h

9.133 exec_stream_t::next_arg_t Class Reference

Inheritance diagram for exec_stream_t::next_arg_t:



Public Member Functions

- virtual ~next_arg_t ()
- virtual std::string const * next ()=0

9.133.1 Constructor & Destructor Documentation

9.133.1.1 virtual exec_stream_t::next_arg_t::~next_arg_t ( ) [inline], [virtual]

9.133.2 Member Function Documentation

9.133.2.1 virtual std::string const* exec_stream_t::next_arg_t::next ( ) [pure virtual]

Implemented in exec_stream_t::next_arg_impl_t< iterator >.

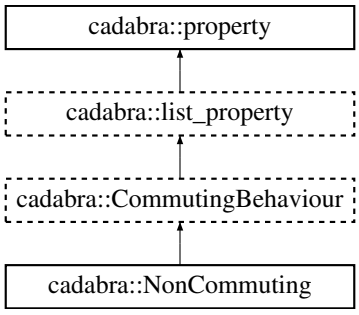
The documentation for this class was generated from the following file:

- frontend/common/exec-stream.h

9.134 cadabra::NonCommuting Class Reference

`#include <NonCommuting.hh>`

Inheritance diagram for cadabra::NonCommuting:



Public Member Functions

- virtual std::string name () const
- virtual int sign () const

Additional Inherited Members

9.134.1 Member Function Documentation

9.134.1.1 std::string NonCommuting::name ( ) const [virtual]

Implements cadabra::property.

9.134.1.2 virtual int cadabra::NonCommuting::sign ( ) const [inline], [virtual]

Implements cadabra::CommutingBehaviour.

The documentation for this class was generated from the following files:

- core/properties/NonCommuting.hh
- core/properties/NonCommuting.cc

9.135 NonScalarException Class Reference

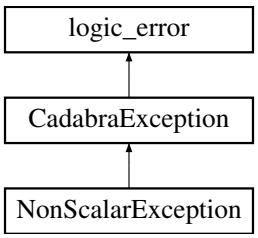
9.135.1 Description

Exception thrown when something requires that an expression is a pure scalar (i.e.

no free indices and no dummy indices), but isn't.

`#include <Exceptions.hh>`

Inheritance diagram for NonScalarException:



Public Member Functions

- NonScalarException (std::string="")
- std::string py_what () const

9.135.2 Constructor & Destructor Documentation

9.135.2.1 NonScalarException::NonScalarException (std::string *s* = " ")

9.135.3 Member Function Documentation

9.135.3.1 std::string NonScalarException::py_what ( ) const

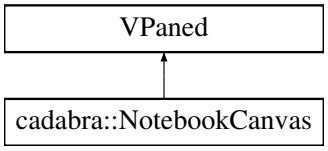
The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

9.136 cadabra::NotebookCanvas Class Reference

`#include <NotebookCanvas.hh>`

Inheritance diagram for cadabra::NotebookCanvas:



Public Member Functions

- NotebookCanvas ()
- ~NotebookCanvas ()
- void refresh_all ()

Public Attributes

- std::map< DataCell *, VisualCell > visualcells
- Gtk::EventBox ebox
- Gtk::ScrolledWindow scroll
- Gtk::HSeparator bottomline
- std::map< DataCell *, NSTextView * > visualcells

9.136.1 Constructor & Destructor Documentation

9.136.1.1 NotebookCanvas::NotebookCanvas ( )

9.136.1.2 NotebookCanvas::~NotebookCanvas ( )

9.136.2 Member Function Documentation

9.136.2.1 void NotebookCanvas::refresh_all ( )

9.136.3 Member Data Documentation

9.136.3.1 Gtk::HSeparator cadabra::NotebookCanvas::bottomline

9.136.3.2 Gtk::EventBox cadabra::NotebookCanvas::ebox

9.136.3.3 Gtk::ScrolledWindow cadabra::NotebookCanvas::scroll

9.136.3.4 std::map<DataCell *, NSTextView *> cadabra::NotebookCanvas::visualcells

9.136.3.5 std::map<DataCell *, VisualCell> cadabra::NotebookCanvas::visualcells

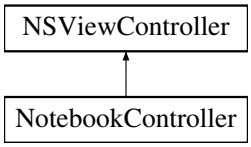
The documentation for this class was generated from the following files:

- frontend/gtkmm/NotebookCanvas.hh
- frontend/gtkmm/NotebookCanvas.cc

9.137 NotebookController Class Reference

```
#import <NotebookController.h>
```

Inheritance diagram for NotebookController:



Properties

- IBOutlet NSTextField * status_label
- IBOutlet NSScrollView * scrollbox

9.137.1 Property Documentation

9.137.1.1 - (IBOutlet NSScrollView∗) scrollbox [read], [write], [atomic], [weak]

9.137.1.2 - (IBOutlet NSTextField∗) status_label [read], [write], [atomic], [weak]

The documentation for this class was generated from the following file:

- frontend/osx/Cadabra/NotebookController.h

9.138 NotebookController() Category Reference

The documentation for this category was generated from the following file:

- frontend/osx/Cadabra/NotebookController.mm

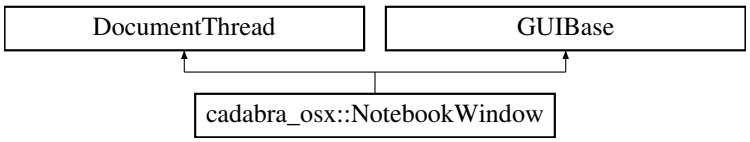
9.139 cadabra_osx::NotebookWindow Class Reference

9.139.1 Description

Objective-C++ class implementing DocumentThread and providing an OS-X notebook interface.

`#include <NotebookWindow.hh>`

Inheritance diagram for cadabra_osx::NotebookWindow:



Public Member Functions

- NotebookWindow (NotebookController ∗)
- virtual void add_cell (const DTree &, DTree::iterator, bool visible) override
- virtual void remove_cell (const DTree &, DTree::iterator) override
- virtual void update_cell (const DTree &, DTree::iterator) override
- virtual void position_cursor (const DTree &, DTree::iterator) override
- virtual void remove_all_cells () override
- virtual void on_connect () override
- virtual void on_disconnect () override
- virtual void on_network_error () override
- virtual void process_data () override

Private Attributes

- NotebookController * controller
- std::vector< NotebookCanvas * > canvasses
- int current_canvas

9.139.2 Constructor & Destructor Documentation

9.139.2.1 NotebookWindow::NotebookWindow (NotebookController * *nc*)

9.139.3 Member Function Documentation

9.139.3.1 void NotebookWindow::add_cell (const DTree & *tr,* DTree::iterator *it,* bool *visible*) [override], [virtual]

9.139.3.2 void NotebookWindow::on_connect ( ) [override], [virtual]

9.139.3.3 void NotebookWindow::on_disconnect ( ) [override], [virtual]

9.139.3.4 void NotebookWindow::on_network_error ( ) [override], [virtual]

9.139.3.5 void NotebookWindow::position_cursor (const DTree & , DTree::iterator) [override], [virtual]

9.139.3.6 void NotebookWindow::process_data ( ) [override], [virtual]

9.139.3.7 void NotebookWindow::remove_all_cells ( ) [override], [virtual]

9.139.3.8 void NotebookWindow::remove_cell (const DTree & , DTree::iterator) [override], [virtual]

9.139.3.9 void NotebookWindow::update_cell (const DTree & , DTree::iterator) [override], [virtual]

9.139.4 Member Data Documentation

9.139.4.1 std::vector<NotebookCanvas *> cadabra_osx::NotebookWindow::canvasses [private]

9.139.4.2 NotebookController* cadabra_osx::NotebookWindow::controller [private]

9.139.4.3 int cadabra_osx::NotebookWindow::current_canvas [private]

The documentation for this class was generated from the following files:

- frontend/osx/Cadabra/Cadabra/NotebookWindow.hh
- frontend/osx/Cadabra/Cadabra/NotebookWindow.mm

9.140 cadabra::NotebookWindow Class Reference

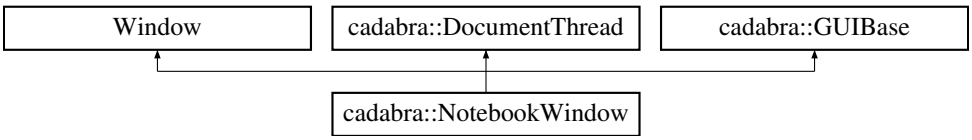
9.140.1 Description

Each notebook has one main window which controls it.

It has a menu bar, a status pane and one or more panels that represent a view on the document.

`#include <NotebookWindow.hh>`

Inheritance diagram for cadabra::NotebookWindow:



Classes

- class Prefs

Public Member Functions

- NotebookWindow (Cadabra *, bool read_only=false)
- ~NotebookWindow ()
- virtual void add_cell (const DTree &, DTree::iterator, bool) override

  Add a GUI cell corresponding to the document cell at the iterator.
- virtual void remove_cell (const DTree &, DTree::iterator) override

  Remove a single cell.
- virtual void remove_all_cells () override

  Remove all GUI cells from the display (used as a quick way to clear all before loading a new document).
- virtual void update_cell (const DTree &, DTree::iterator) override

  The basic manipulations that a GUI needs to implement are adding, removing and updating (refreshing the display of) a cell.
- virtual void position_cursor (const DTree &, DTree::iterator, int pos) override

  Position the cursor in the current canvas in the widget corresponding to the indicated cell.
- virtual size_t get_cursor_position (const DTree &, DTree::iterator) override

  Retrieve the position of the cursor in the current cell.
- virtual void on_connect () override

  Network status is propagated from the ComputeThread to the.
- virtual void on_disconnect (const std::string &) override
- virtual void on_network_error () override
- virtual void on_kernel_runstatus (bool) override
- virtual void process_data () override

  When the ComputeThread needs to modify the document, it stores an ActionBase object on the stack (see the DocumenThread class) and then wakes up the GUI thread signalling it to process this action.
- bool on_vscroll_changed (Gtk::ScrollType, double)
- bool on_scroll (GdkEventScroll *)
- void on_scroll_size_allocate (Gtk::Allocation &)
- void scroll_current_cell_into_view ()
- void set_name (const std::string &)
- void set_title_prefix (const std::string &)
- void load_file (const std::string &notebook_contents)

Public Attributes

- TeXEngine engine
- double scale

Protected Member Functions

- virtual bool on_key_press_event (GdkEventKey *) override
- virtual bool on_delete_event (GdkEventAny *) override
- virtual bool on_configure_event (GdkEventConfigure *cfg) override

Protected Attributes

- DTree::iterator current_cell

Private Member Functions

- void update_title ()
- void set_stop_sensitive (bool)
- void on_file_new ()
- void on_file_open ()
- void on_file_close ()
- void on_file_save ()
- void on_file_save_as ()
- void on_file_export_html ()
- void on_file_export_html_segment ()
- void on_file_export_latex ()
- void on_file_export_python ()
- void on_file_quit ()
- bool quit_safeguard (bool quit)
- void on_edit_undo ()
- void on_edit_insert_above ()
- void on_edit_insert_below ()
- void on_edit_delete ()
- void on_edit_split ()
- void on_edit_cell_is_latex ()
- void on_edit_cell_is_python ()
- void on_view_split ()
- void on_view_close ()
- void on_run_cell ()
- void on_run_runall ()
- void on_run_runtocursor ()
- void on_run_stop ()
- void on_prefs_font_size (int num)
- void on_help_about ()
- void on_help () const
- void on_kernel_restart ()
- std::string save (const std::string &fn) const
- void process_todo_queue ()
- void on_crash_window_closed (int)
- bool cell_got_focus (DTree::iterator, int)
- bool cell_toggle_visibility (DTree::iterator it, int)

- bool cell_content_insert (const std::string &, int, DTree::iterator, int)
- bool cell_content_erase (int, int, DTree::iterator, int)
- bool cell_content_execute (DTree::iterator, int, bool shift_enter_pressed)
- bool cell_content_changed (const std::string &content, DTree::iterator it, int canvas_number)
- void dim_output_cells (DTree::iterator it)
- bool on_tex_error (const std::string &, DTree::iterator)
- void setup_css_provider ()
- void on_text_scaling_factor_changed (const std::string &key)
- bool idle_handler ()

Private Attributes

- Cadabra * cdbapp
- Glib::Dispatcher dispatcher
- Glib::RefPtr< Gtk::ActionGroup > actiongroup
- Glib::RefPtr< Gtk::UIManager > uimanager
- Gtk::VBox topbox
- Gtk::HBox supermainbox
- Gtk::VBox mainbox
- Gtk::HBox statusbarbox
- std::vector< NotebookCanvas * > canvasses
- int current_canvas
- Gtk::ProgressBar progressbar
- Gtk::Spinner kernel_spinner
- bool kernel_spinner_status
- Gtk::Label status_label
- Gtk::Label kernel_label
- std::mutex status_mutex
- std::string status_string
- std::string kernel_string
- std::string name
- std::string title_prefix
- bool modified
- bool read_only
- bool crash_window_hidden
- Glib::RefPtr< Gtk::CssProvider > css_provider
- Glib::RefPtr< Gio::Settings > settings
- int last_configure_width
- DTree::iterator follow_cell
- Prefs prefs
- bool is_configured

Additional Inherited Members

9.140.2 Constructor & Destructor Documentation

9.140.2.1 NotebookWindow::NotebookWindow (Cadabra * *c*, bool *read_only* = `false`)

9.140.2.2 NotebookWindow::~NotebookWindow ( )

9.140.3 Member Function Documentation

9.140.3.1 void NotebookWindow::add_cell (const DTree & , DTree::iterator , bool *visible*) [override], [virtual]

Add a GUI cell corresponding to the document cell at the iterator.

The GUI needs to figure out from the location of this cell in the DTree where to insert the cell in the visual display. If the 'visible' flag is false, hide the cell from view independent of whether its hidden flag is set (this is only used when constructing a document on load time and we do not want to show cells until they have all been added to the document).

Implements cadabra::GUIBase.

9.140.3.2 bool NotebookWindow::cell_content_changed (const std::string & *content,* DTree::iterator *it,* int *canvas_number*) [private]

9.140.3.3 bool NotebookWindow::cell_content_erase (int *start,* int *end,* DTree::iterator *it,* int *canvas_number*) [private]

9.140.3.4 bool NotebookWindow::cell_content_execute (DTree::iterator *it,* int *canvas_number,* bool *shift_enter_pressed*) [private]

9.140.3.5 bool NotebookWindow::cell_content_insert (const std::string & *content,* int *pos,* DTree::iterator *it,* int *canvas_number*) [private]

9.140.3.6 bool NotebookWindow::cell_got_focus (DTree::iterator *it,* int *canvas_number*) [private]

9.140.3.7 bool NotebookWindow::cell_toggle_visibility (DTree::iterator *it,* int *canvas_number*) [private]

9.140.3.8 void NotebookWindow::dim_output_cells (DTree::iterator *it*) [private]

9.140.3.9 size_t NotebookWindow::get_cursor_position (const DTree & , DTree::iterator) [override], [virtual]

Retrieve the position of the cursor in the current cell.

Implements cadabra::GUIBase.

9.140.3.10 bool NotebookWindow::idle_handler ( ) [private]

9.140.3.11 void NotebookWindow::load_file (const std::string & *notebook_contents*)

9.140.3.12 bool NotebookWindow::on_configure_event (GdkEventConfigure * *cfg*) [override], [protected], [virtual]

9.140.3.13 void NotebookWindow::on_connect ( ) [override], [virtual]

Network status is propagated from the ComputeThread to the.

GUI using the following methods. These get called on the compute thread (as opposed to the functions above, which get called on the gui thread).

Implements cadabra::GUIBase.

9.140.3.14 void NotebookWindow::on_crash_window_closed (int) [private]

9.140.3.15 bool NotebookWindow::on_delete_event (GdkEventAny * *event*) [override], [protected], [virtual]

9.140.3.16 void NotebookWindow::on_disconnect (const std::string & *reason*) [override], [virtual]

Implements cadabra::GUIBase.

9.140.3.17 void NotebookWindow::on_edit_cell_is_latex ( ) [private]

9.140.3.18 void NotebookWindow::on_edit_cell_is_python ( ) [private]

9.140.3.19 void NotebookWindow::on_edit_delete ( ) [private]

9.140.3.20 void NotebookWindow::on_edit_insert_above ( ) [private]

9.140.3.21 void NotebookWindow::on_edit_insert_below ( ) [private]

9.140.3.22 void NotebookWindow::on_edit_split ( ) [private]

9.140.3.23 void NotebookWindow::on_edit_undo ( ) [private]

9.140.3.24 void NotebookWindow::on_file_close ( ) [private]

9.140.3.25 void NotebookWindow::on_file_export_html ( ) [private]

9.140.3.26 void NotebookWindow::on_file_export_html_segment ( ) [private]

9.140.3.27 void NotebookWindow::on_file_export_latex ( ) [private]

9.140.3.28 void NotebookWindow::on_file_export_python ( ) [private]

9.140.3.29 void NotebookWindow::on_file_new ( ) [private]

9.140.3.30 void NotebookWindow::on_file_open ( ) [private]

9.140.3.31 void NotebookWindow::on_file_quit ( ) [private]

9.140.3.32 void NotebookWindow::on_file_save ( ) [private]

9.140.3.33 void NotebookWindow::on_file_save_as ( ) [private]

9.140.3.34 void NotebookWindow::on_help ( ) const [private]

9.140.3.35 void NotebookWindow::on_help_about ( ) [private]

9.140.3.36 void NotebookWindow::on_kernel_restart ( ) [private]

9.140.3.37 void NotebookWindow::on_kernel_runstatus (bool *running*) [override], [virtual]

Implements cadabra::GUIBase.

9.140.3.38 bool NotebookWindow::on_key_press_event (GdkEventKey * *event*) [override], [protected], [virtual]

9.140.3.39 void NotebookWindow::on_network_error ( ) [override], [virtual]

Implements cadabra::GUIBase.

9.140.3.40 void NotebookWindow::on_prefs_font_size (int *num*) [private]

9.140.3.41 void NotebookWindow::on_run_cell ( ) [private]

9.140.3.42 void NotebookWindow::on_run_runall ( ) [private]

9.140.3.43 void NotebookWindow::on_run_runtocursor ( ) [private]

9.140.3.44 void NotebookWindow::on_run_stop ( ) [private]

9.140.3.45 bool NotebookWindow::on_scroll (GdkEventScroll * *s*)

9.140.3.46 void NotebookWindow::on_scroll_size_allocate (Gtk::Allocation & *scroll_alloc*)

9.140.3.47 bool NotebookWindow::on_tex_error (const std::string & *str,* DTree::iterator *it*) [private]

9.140.3.48 void NotebookWindow::on_text_scaling_factor_changed (const std::string & *key*) [private]

9.140.3.49 void NotebookWindow::on_view_close ( ) [private]

9.140.3.50 void NotebookWindow::on_view_split ( ) [private]

9.140.3.51 bool NotebookWindow::on_vscroll_changed (Gtk::ScrollType *st,* double *v*)

9.140.3.52 void NotebookWindow::position_cursor (const DTree & , DTree::iterator , int) [override], [virtual]

Position the cursor in the current canvas in the widget corresponding to the indicated cell.

Implements cadabra::GUIBase.

9.140.3.53 void NotebookWindow::process_data ( ) [override], [virtual]

When the ComputeThread needs to modify the document, it stores an ActionBase object on the stack (see the DocumenThread class) and then wakes up the GUI thread signalling it to process this action.

The following member should wake up the GUI thread and make it enter the processing part.

Implements cadabra::GUIBase.

9.140.3.54 void NotebookWindow::process_todo_queue ( ) [private]

9.140.3.55 bool NotebookWindow::quit_safeguard (bool *quit*) [private]

9.140.3.56 void NotebookWindow::remove_all_cells ( ) [override], [virtual]

Remove all GUI cells from the display (used as a quick way to clear all before loading a new document).

Implements cadabra::GUIBase.

9.140.3.57 void NotebookWindow::remove_cell (const DTree & , DTree::iterator) [override], [virtual]

Remove a single cell.

Implements cadabra::GUIBase.

9.140.3.58 std::string NotebookWindow::save (const std::string & *fn*) const [private]

9.140.3.59 void NotebookWindow::scroll_current_cell_into_view ( )

9.140.3.60 void NotebookWindow::set_name (const std::string & *n*)

9.140.3.61 void NotebookWindow::set_stop_sensitive (bool *s*) [private]

9.140.3.62 void NotebookWindow::set_title_prefix (const std::string & *pf*)

9.140.3.63 void cadabra::NotebookWindow::setup_css_provider ( ) [private]

9.140.3.64 void NotebookWindow::update_cell (const DTree & , DTree::iterator) [override], [virtual]

The basic manipulations that a GUI needs to implement are adding, removing and updating (refreshing the display of) a cell.

The code in DocumentThread will call these to make the GUI update its display. Called on the document thread.

Implements cadabra::GUIBase.

9.140.3.65 void NotebookWindow::update_title ( ) [private]

9.140.4 Member Data Documentation

9.140.4.1 Glib::RefPtr<Gtk::ActionGroup> cadabra::NotebookWindow::actiongroup [private]

9.140.4.2 std::vector<NotebookCanvas *> cadabra::NotebookWindow::canvasses [private]

9.140.4.3 Cadabra* cadabra::NotebookWindow::cdbapp [private]

9.140.4.4 bool cadabra::NotebookWindow::crash_window_hidden [private]

9.140.4.5 Glib::RefPtr<Gtk::CssProvider> cadabra::NotebookWindow::css_provider [private]

9.140.4.6 int cadabra::NotebookWindow::current_canvas [private]

9.140.4.7 DTree::iterator cadabra::NotebookWindow::current_cell [protected]

9.140.4.8 Glib::Dispatcher cadabra::NotebookWindow::dispatcher [private]

9.140.4.9 TeXEngine cadabra::NotebookWindow::engine

9.140.4.10 DTree::iterator cadabra::NotebookWindow::follow_cell [private]

9.140.4.11 bool cadabra::NotebookWindow::is_configured [private]

9.140.4.12 Gtk::Label cadabra::NotebookWindow::kernel_label [private]

9.140.4.13 Gtk::Spinner cadabra::NotebookWindow::kernel_spinner [private]

9.140.4.14 bool cadabra::NotebookWindow::kernel_spinner_status [private]

9.140.4.15 std::string cadabra::NotebookWindow::kernel_string [private]

9.140.4.16 int cadabra::NotebookWindow::last_configure_width [private]

9.140.4.17 Gtk::VBox cadabra::NotebookWindow::mainbox [private]

9.140.4.18 bool cadabra::NotebookWindow::modified [private]

9.140.4.19 std::string cadabra::NotebookWindow::name [private]

9.140.4.20 Prefs cadabra::NotebookWindow::prefs [private]

9.140.4.21 Gtk::ProgressBar cadabra::NotebookWindow::progressbar [private]

9.140.4.22 bool cadabra::NotebookWindow::read_only [private]

9.140.4.23 double cadabra::NotebookWindow::scale

9.140.4.24 Glib::RefPtr<Gio::Settings> cadabra::NotebookWindow::settings [private]

9.140.4.25 Gtk::Label cadabra::NotebookWindow::status_label [private]

9.140.4.26 std::mutex cadabra::NotebookWindow::status_mutex [private]

9.140.4.27 std::string cadabra::NotebookWindow::status_string [private]

9.140.4.28 Gtk::HBox cadabra::NotebookWindow::statusbarbox [private]

9.140.4.29 Gtk::HBox cadabra::NotebookWindow::supermainbox [private]

9.140.4.30 std::string cadabra::NotebookWindow::title_prefix [private]

9.140.4.31 Gtk::VBox cadabra::NotebookWindow::topbox [private]

9.140.4.32 Glib::RefPtr<Gtk::UIManager> cadabra::NotebookWindow::uimanager [private]

The documentation for this class was generated from the following files:

- frontend/gtkmm/NotebookWindow.hh
- frontend/gtkmm/NotebookWindow.cc

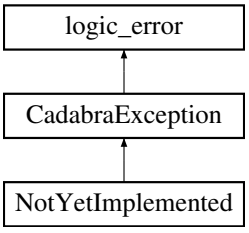
9.141 NotYetImplemented Class Reference

9.141.1 Description

Exception thrown when there the Cadabra code detects an internal inconsistency; this should never happen, but is better than bailing out with an assert.

`#include <Exceptions.hh>`

Inheritance diagram for NotYetImplemented:



Public Member Functions

- NotYetImplemented (std::string="")
- std::string py_what () const

9.141.2 Constructor & Destructor Documentation

9.141.2.1 NotYetImplemented::NotYetImplemented (std::string *s* = " ")

9.141.3 Member Function Documentation

9.141.3.1 std::string NotYetImplemented::py_what ( ) const

The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

9.142 cadabra::nset_it_less Class Reference

9.142.1 Description

Compare two nset iterators by comparing the strings to which they point.

```
#include <Storage.hh>
```

Public Member Functions

- bool operator() (nset_t::iterator first, nset_t::iterator second) const

9.142.2 Member Function Documentation

9.142.2.1 bool cadabra::nset_it_less::operator() (nset_t::iterator *first*, nset_t::iterator *second*) const

The documentation for this class was generated from the following files:

- core/Storage.hh
- core/Storage.cc

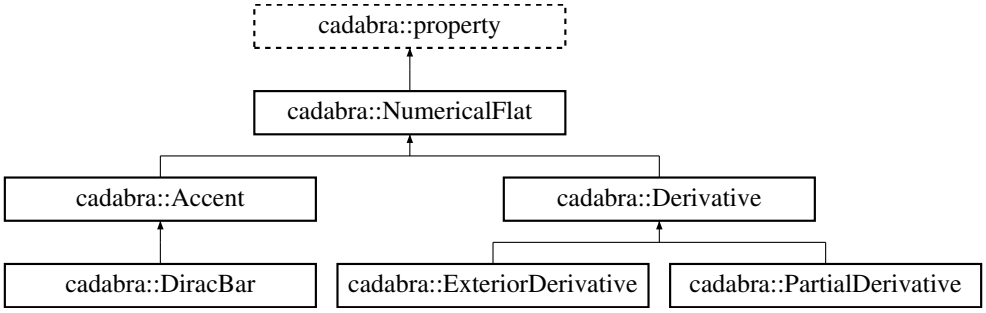
9.143 cadabra::NumericalFlat Class Reference

9.143.1 Description

Property indicating that an operator is numerically flat, so that numerical factors in the argument can be taken outside.

`#include <NumericalFlat.hh>`

Inheritance diagram for cadabra::NumericalFlat:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.143.2 Member Function Documentation

9.143.2.1 std::string NumericalFlat::name ( ) const [virtual]

Implements cadabra::property.

Reimplemented in cadabra::Accent, cadabra::Derivative, cadabra::ExteriorDerivative, cadabra::DiracBar, and cadabra::PartialDerivative.

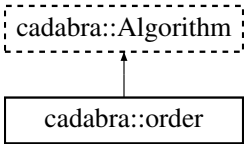
The documentation for this class was generated from the following files:

- core/properties/NumericalFlat.hh
- core/properties/NumericalFlat.cc

9.144 cadabra::order Class Reference

`#include <order.hh>`

Inheritance diagram for cadabra::order:



Public Member Functions

- order (const Kernel &, Ex &, Ex &objs, bool ac)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Protected Attributes

- Ex objects
- bool anticomm

Additional Inherited Members

9.144.1 Constructor & Destructor Documentation

9.144.1.1 order::order (const Kernel & *k,* Ex & *tr,* Ex & *objs,* bool *ac*)

9.144.2 Member Function Documentation

9.144.2.1 Algorithm::result_t order::apply (iterator & *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.144.2.2 bool order::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.144.3 Member Data Documentation

9.144.3.1 bool cadabra::order::anticomm [protected]

9.144.3.2 Ex cadabra::order::objects [protected]

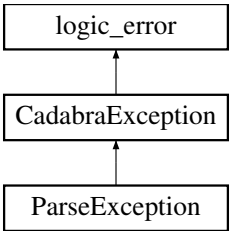
The documentation for this class was generated from the following files:

- core/algorithms/order.hh
- core/algorithms/order.cc

9.145 ParseException Class Reference

`#include <Exceptions.hh>`

Inheritance diagram for ParseException:



Public Member Functions

- ParseException (std::string)

9.145.1 Constructor & Destructor Documentation

9.145.1.1 ParseException::ParseException (std::string *s*)

The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

9.146 cadabra::Parser Class Reference

`#include <Parser.hh>`

Public Member Functions

- Parser ()
- Parser (std::shared_ptr< Ex >)
- Parser (std::shared_ptr< Ex >, const std::string &)
- void erase ()
- void remove_empty_nodes ()
- void finalise ()
- bool string2tree (const std::string &inp)

Public Attributes

- std::shared_ptr< Ex > tree

Private Types

- enum mode_t {
  m_skipwhite, m_name, m_findchildren, m_singlecharname,
  m_backslashname, m_childgroup, m_initialgroup, m_verbatim,
  m_property }

Private Member Functions

- void advance (unsigned int &i)
- unsigned char get_token (unsigned int i)
- bool is_number (const std::string &str) const
- str_node::bracket_t is_closing_bracket (const unsigned char &br) const
- str_node::bracket_t is_opening_bracket (const unsigned char &br) const
- str_node::parent_rel_t is_link (const unsigned char &ln) const

Private Attributes

- Ex::iterator parts
- std::string str
- std::vector< mode_t > current_mode
- std::vector< str_node::bracket_t > current_bracket
- std::vector< str_node::parent_rel_t > current_parent_rel

9.146.1 Member Enumeration Documentation

9.146.1.1 enum cadabra::Parser::mode_t [private]

Enumerator

m_skipwhite

m_name

m_findchildren

m_singlecharname

m_backslashname

m_childgroup

m_initialgroup

m_verbatim

m_property

9.146.2 Constructor & Destructor Documentation

9.146.2.1 Parser::Parser ( )

9.146.2.2 Parser::Parser (std::shared_ptr< Ex > *t*)

9.146.2.3 Parser::Parser (std::shared_ptr< Ex > *t*, const std::string & *str*)

9.146.3 Member Function Documentation

9.146.3.1 void Parser::advance (unsigned int & *i*) [private]

9.146.3.2 void Parser::erase ( )

9.146.3.3 void Parser::finalise ( )

9.146.3.4 unsigned char Parser::get_token (unsigned int *i*) [private]

9.146.3.5 str_node::bracket_t Parser::is_closing_bracket (const unsigned char & *br*) const [private]

9.146.3.6 str_node::parent_rel_t Parser::is_link (const unsigned char & *ln*) const [private]

9.146.3.7 bool Parser::is_number (const std::string & *str*) const [private]

9.146.3.8 str_node::bracket_t Parser::is_opening_bracket (const unsigned char & *br*) const [private]

9.146.3.9 void Parser::remove_empty_nodes ( )

9.146.3.10 bool Parser::string2tree (const std::string & *inp*)

9.146.4 Member Data Documentation

9.146.4.1 std::vector<str_node::bracket_t> cadabra::Parser::current_bracket [private]

9.146.4.2 std::vector<mode_t> cadabra::Parser::current_mode [private]

9.146.4.3 std::vector<str_node::parent_rel_t> cadabra::Parser::current_parent_rel [private]

9.146.4.4 Ex::iterator cadabra::Parser::parts [private]

9.146.4.5 std::string cadabra::Parser::str [private]

9.146.4.6 std::shared_ptr<Ex> cadabra::Parser::tree

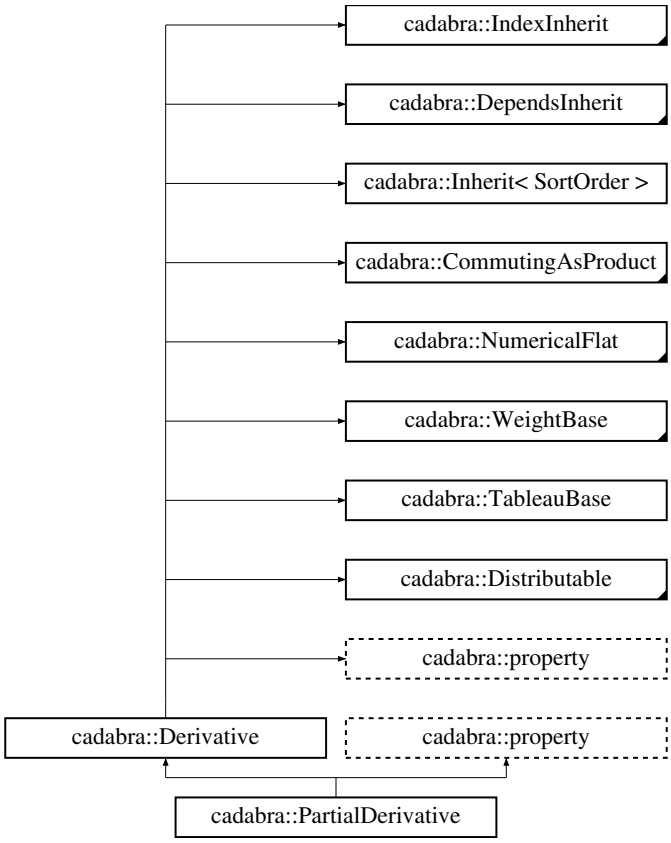
The documentation for this class was generated from the following files:

- core/Parser.hh
- core/Parser.cc

9.147 cadabra::PartialDerivative Class Reference

`#include <PartialDerivative.hh>`

Inheritance diagram for cadabra::PartialDerivative:



Public Member Functions

- virtual ~PartialDerivative ()
- virtual std::string name () const
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const

Additional Inherited Members

9.147.1 Constructor & Destructor Documentation

9.147.1.1 virtual cadabra::PartialDerivative::~PartialDerivative ( ) [inline], [virtual]

9.147.2 Member Function Documentation

9.147.2.1 TableauBase::tab_t PartialDerivative::get_tab (const Properties & *properties,* Ex & *tr,* Ex::iterator *it,* unsigned int *num*) const [virtual]

Reimplemented from cadabra::Derivative.

9.147.2.2 std::string PartialDerivative::name ( ) const [virtual]

Implements cadabra::property.

9.147.2.3 unsigned int PartialDerivative::size (const Properties & *properties*, Ex & *tr*, Ex::iterator *it*) const [virtual]

Reimplemented from cadabra::Derivative.

The documentation for this class was generated from the following files:

- core/properties/PartialDerivative.hh
- core/properties/PartialDerivative.cc

9.148 cadabra::pattern Class Reference

`#include <Props.hh>`

Public Member Functions

- pattern ()
- pattern (const Ex &)
- bool match (const Properties &, const Ex::iterator &, bool ignore_parent_rel=false) const
- bool children_wildcard () const

Public Attributes

- Ex obj

9.148.1 Constructor & Destructor Documentation

9.148.1.1 pattern::pattern ( )

9.148.1.2 pattern::pattern (const Ex & *o*)

9.148.2 Member Function Documentation

9.148.2.1 bool pattern::children_wildcard ( ) const

9.148.2.2 bool pattern::match (const Properties & *properties*, const Ex::iterator & *it*, bool *ignore_parent_rel* = false) const

9.148.3 Member Data Documentation

9.148.3.1 Ex cadabra::pattern::obj

The documentation for this class was generated from the following files:

- core/Props.hh
- core/Props.cc

9.149 Perm Class Reference

`#include <Permutations.hh>`

Public Member Functions

- template<class iterator >
  void find (iterator start1, iterator end1, iterator start2, iterator end2)

  Find the permutation that takes [start1, end1> to [start2, end2>.
- template<class iterator >
  void apply (iterator start1, iterator end1)

  Apply the permutation 'perm' on the given range.

Public Attributes

- std::vector< int > perm

9.149.1 Member Function Documentation

9.149.1.1 template<class iterator > void Perm::apply (iterator *start1,* iterator *end1*)

Apply the permutation 'perm' on the given range.

9.149.1.2 template<class iterator > void Perm::find (iterator *start1,* iterator *end1,* iterator *start2,* iterator *end2*)

Find the permutation that takes [start1, end1> to [start2, end2>.

This will be available in 'perm' afterwards.

9.149.2 Member Data Documentation

9.149.2.1 std::vector<int> Perm::perm

The documentation for this class was generated from the following file:

- core/Permutations.hh

9.150 PermutationException Class Reference

9.150.1 Description

Generic permutation group material.

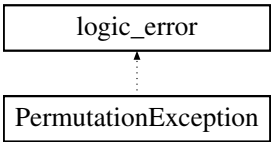
Largely follows the notation of xperm to avoid confusion.

Example: 1->4, 4->3, 3->1, 5->6, 6->5

Perm(3, 2, 4, 1, 6, 5) Images(4, 2, 1, 3, 6, 5)

`#include <Permutations.hh>`

Inheritance diagram for PermutationException:



Public Member Functions

- PermutationException (const std::string &ex)

9.150.2 Constructor & Destructor Documentation

9.150.2.1 PermutationException::PermutationException (const std::string & *ex*) [inline]

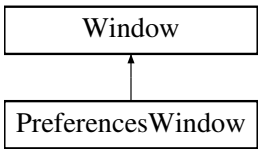
The documentation for this class was generated from the following file:

- core/Permutations.hh

9.151 PreferencesWindow Class Reference

`#include <Preferences.hh>`

Inheritance diagram for PreferencesWindow:



Private Attributes

- Gtk::VBox mainbox

9.151.1 Member Data Documentation

9.151.1.1 Gtk::VBox PreferencesWindow::mainbox [private]

The documentation for this class was generated from the following file:

- frontend/gtkmm/Preferences.hh

9.152 cadabra::NotebookWindow::Prefs Class Reference

Public Member Functions

- Prefs ()

Public Attributes

- int font_step

9.152.1 Constructor & Destructor Documentation

9.152.1.1 NotebookWindow::Prefs::Prefs ( )

9.152.2 Member Data Documentation

9.152.2.1 int cadabra::NotebookWindow::Prefs::font_step

The documentation for this class was generated from the following files:

- frontend/gtkmm/NotebookWindow.hh
- frontend/gtkmm/NotebookWindow.cc

9.153 preprocessor Class Reference

9.153.1 Description

Preprocessing class which takes infix mathematical notation with all sorts of maths shortcuts and transforms it into a string which is properly formatted in prefix notation.

Use the Parser class to turn this string into an Ex expression tree.

```
#include <PreProcessor.hh>
```

Classes

- class accu_t

Public Types

- enum order_labels {
  order_factorial =0, order_pow, order_frac, order_prod,
  order_wedge, order_minus, order_plus, order_dot,
  order_equals, order_unequals, order_less_than, order_greater_than,
  order_conditions, order_arrow, order_set_option, order_colon,
  order_comma, order_tilde }

Public Member Functions

- preprocessor ()
- void erase ()
- void strip_outer_brackets () const

Static Public Attributes

- static const unsigned char orders [ ]
- static const char ∗const order_names [ ]

Private Member Functions

- void parse_ (const std::string &)
- void parse_internal_ ()
- bool unwind_ (unsigned int tolevel, unsigned int bracketgoal=0, bool usebracket=true) const
- unsigned char get_token_ (unsigned char prev_token)
- void show_and_throw_ (const std::string &str) const
- void bracket_strings_ (unsigned int cb, std::string &obrack, std::string &cbrack) const
- bool is_infix_operator_ (unsigned char c) const
- bool is_link_ (unsigned char c) const
- unsigned int is_opening_bracket_ (unsigned char c) const
- unsigned int is_closing_bracket_ (unsigned char c) const
- unsigned int is_bracket_ (unsigned char c) const
- bool is_already_bracketed_ (const std::string &str) const
- bool is_digits_ (const std::string &str) const
- unsigned int current_bracket_ (bool deep=false) const
- void print_stack () const
- bool default_is_product_ () const

Private Attributes

- bool verbatim_
- bool next_is_product_
- bool eat_initial_whitespace_
- unsigned int cur_pos
- std::string cur_str
- accu_t cur
- std::vector< accu_t > accus

Static Private Attributes

- static const unsigned char open_brackets [ ] ={ '{', '(', '[' }
- static const unsigned char close_brackets [ ] ={ '}', ')', ']' }

Friends

- std::istream & operator>> (std::istream &, preprocessor &)
- std::ostream & operator<< (std::ostream &, const preprocessor &)

9.153.2 Member Enumeration Documentation

9.153.2.1 enum preprocessor::order_labels

Enumerator

order_factorial

order_pow

order_frac

order_prod

order_wedge

order_minus

order_plus

order_dot

order_equals

order_unequals

order_less_than

order_greater_than

order_conditions

order_arrow

order_set_option

order_colon

order_comma

order_tilde

9.153.3 Constructor & Destructor Documentation

9.153.3.1 preprocessor::preprocessor ( )

9.153.4 Member Function Documentation

9.153.4.1 void preprocessor::bracket_strings_ (unsigned int *cb,* std::string & *obrack,* std::string & *cbrack*) const [private]

9.153.4.2 unsigned int preprocessor::current_bracket_ (bool *deep* = false) const [private]

9.153.4.3 bool preprocessor::default_is_product_ ( ) const [private]

9.153.4.4 void preprocessor::erase ( )

9.153.4.5 unsigned char preprocessor::get_token_ (unsigned char *prev_token*) [private]

9.153.4.6 bool preprocessor::is_already_bracketed_ (const std::string & *str*) const [private]

9.153.4.7 unsigned int preprocessor::is_bracket_ (unsigned char *c*) const [private]

9.153.4.8 unsigned int preprocessor::is_closing_bracket_ (unsigned char *c*) const [private]

9.153.4.9 bool preprocessor::is_digits_ (const std::string & *str*) const [private]

9.153.4.10 bool preprocessor::is_infix_operator_ (unsigned char *c*) const [private]

9.153.4.11 bool preprocessor::is_link_ (unsigned char *c*) const [private]

9.153.4.12 unsigned int preprocessor::is_opening_bracket_ (unsigned char *c*) const [private]

9.153.4.13 void preprocessor::parse_ (const std::string & *str*) [private]

9.153.4.14 void preprocessor::parse_internal_ ( ) [private]

9.153.4.15 void preprocessor::print_stack ( ) const [private]

9.153.4.16 void preprocessor::show_and_throw_ (const std::string & *str*) const [private]

9.153.4.17 void preprocessor::strip_outer_brackets ( ) const

9.153.4.18 bool preprocessor::unwind_ (unsigned int *tolevel,* unsigned int *bracketgoal* = 0, bool *usebracket* = true) const [private]

9.153.5 Friends And Related Function Documentation

9.153.5.1 std::ostream& operator<< (std::ostream & , const **preprocessor** &) [friend]

9.153.5.2 std::istream& operator>> (std::istream & , **preprocessor** &) [friend]

9.153.6 Member Data Documentation

9.153.6.1 std::vector<**accu_t**> preprocessor::accus [mutable], [private]

9.153.6.2 const unsigned char preprocessor::close_brackets ={ '}', ')', ']' } [static], [private]

9.153.6.3 accu_t preprocessor::cur [mutable], [private]

9.153.6.4 unsigned int preprocessor::cur_pos [private]

9.153.6.5 std::string preprocessor::cur_str [private]

9.153.6.6 bool preprocessor::eat_initial_whitespace_ [private]

9.153.6.7 bool preprocessor::next_is_product_ [private]

9.153.6.8 const unsigned char preprocessor::open_brackets ={ '{', '(', '[' } [static], [private]

9.153.6.9 const char *const preprocessor::order_names [static]

Initial value:

```
={ "\\factorial",

                                                    "\\pow", "\\frac", "\\prod", "\\wedge",
                                                    "\\sub", "\\sum", "\\sequence",
                                                    "\\equals", "\\unequals", "\\less",
                                                    "\\greater", "\\conditional",
                                                    "\\arrow", "\\setoption", "\\declare",
                                                    "\\comma", "\\tie", 0 }
```

9.153.6.10 const unsigned char preprocessor::orders [static]

Initial value:

```
={ '!', tok_pow, '/', '*', tok_wedge,

      tok_sequence, '=', tok_unequals,

      tok_arrow, tok_set_option,

                                                    '-', '+',

                                                    '<', '>', '|',

                                                    tok_declare, ',', '~', 0 }
```

9.153.6.11 bool preprocessor::verbatim_ [private]

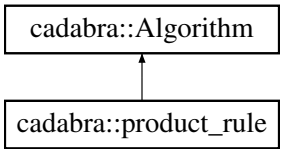
The documentation for this class was generated from the following files:

- core/PreProcessor.hh
- core/PreProcessor.cc

9.154 cadabra::product_rule Class Reference

`#include <product_rule.hh>`

Inheritance diagram for cadabra::product_rule:



Public Member Functions

- product_rule (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Public Attributes

- sibling_iterator prodnode
- unsigned int number_of_indices

Additional Inherited Members

9.154.1 Constructor & Destructor Documentation

9.154.1.1 product_rule::product_rule (const Kernel & *k,* Ex & *tr*)

9.154.2 Member Function Documentation

9.154.2.1 Algorithm::result_t product_rule::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.154.2.2 bool product_rule::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.154.3 Member Data Documentation

9.154.3.1 unsigned int cadabra::product_rule::number_of_indices

9.154.3.2 sibling_iterator cadabra::product_rule::prodnode

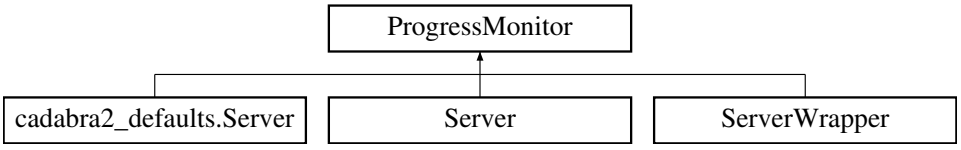
The documentation for this class was generated from the following files:

- core/algorithms/product_rule.hh
- core/algorithms/product_rule.cc

9.155 ProgressMonitor Class Reference

`#include <ProgressMonitor.hh>`

Inheritance diagram for ProgressMonitor:



Classes

- class Block
- class Total

Public Member Functions

- ProgressMonitor ()
- virtual ~ProgressMonitor ()
- virtual void group (std::string name="")
- virtual void progress (int n, int total)
- void print () const
- std::vector< Total > totals () const

Private Attributes

- std::stack< Block > call_stack
- std::map< std::string, Total > call_totals

9.155.1 Constructor & Destructor Documentation

9.155.1.1 ProgressMonitor::ProgressMonitor ( )

9.155.1.2 ProgressMonitor::~ProgressMonitor ( ) [virtual]

9.155.2 Member Function Documentation

9.155.2.1 void ProgressMonitor::group (std::string *name* = "") [virtual]

Reimplemented in ServerWrapper.

9.155.2.2 void ProgressMonitor::print ( ) const

9.155.2.3 void ProgressMonitor::progress (int *n*, int *total*) [virtual]

Reimplemented in ServerWrapper.

9.155.2.4 std::vector< ProgressMonitor::Total > ProgressMonitor::totals ( ) const

9.155.3 Member Data Documentation

9.155.3.1 std::stack<Block> ProgressMonitor::call_stack [private]

9.155.3.2 std::map<std::string, Total> ProgressMonitor::call_totals [private]

The documentation for this class was generated from the following files:

- core/ProgressMonitor.hh
- core/ProgressMonitor.cc

9.156 cadabra::Properties Class Reference

9.156.1 Description

Class holding a collection of properties attached to expressions.

Symbols and expressions do not have a default meaning in Cadabra. They get their meaning by attaching properties to them. When the core manipulator calls an algorithm object, it passes an instance of the Properties class along with the expression tree on which to act, so that the algorithm can figure out what the symbols in the expression tree mean.

`#include <Props.hh>`

Classes

- class registered_property_map_t

Public Types

- typedef std::pair< pattern *, const property * > pat_prop_pair_t
- typedef std::multimap< nset_t::iterator, pat_prop_pair_t, nset_it_less > property_map_t

  We keep two multi-maps: one from the pattern to the property (roughly) and one from the property to the pattern.
- typedef std::multimap< const property *, pattern * > pattern_map_t

Public Member Functions

- void register_property (property *(*)(), const std::string &name)

  Registering properties.
- std::string master_insert (Ex proptree, property *thepropbase)

  Register a property for the indicated Ex.
- void clear ()
- template<class T >
  const T * get (Ex::iterator, bool ignore_parent_rel=false) const
- template<class T >
  const T * get () const
- template<class T >
  const T * get_composite (Ex::iterator, bool ignore_parent_rel=false) const
- template<class T >
  const T * get_composite (Ex::iterator, int &serialnum, bool doserial=true, bool ignore_parent_rel=false) const
- template<class T >
  const T * get_composite (Ex::iterator, const std::string &label) const
- template<class T >
  const T * get_composite (Ex::iterator, int &serialnum, const std::string &label, bool doserial=true) const
- template<class T >
  const T * get_composite (Ex::iterator, Ex::iterator, bool ignore_parent_rel=false) const
- template<class T >
  const T * get_composite (Ex::iterator, Ex::iterator, int &, int &, bool ignore_parent_rel=false) const
- template<class T >
  Ex::iterator head (Ex::iterator, bool ignore_parent_rel=false) const
- bool has (const property *, Ex::iterator)
- int serial_number (const property *, const pattern *) const

Public Attributes

- registered_property_map_t registered_properties
- property_map_t props

  The following two maps own the pointers to the properties and patterns stored in them; use clear() to clean up.
- pattern_map_t pats

Private Member Functions

- void insert_prop (const Ex &, const property *)
- void insert_list_prop (const std::vector< Ex > &, const list_property *)

9.156.2 Member Typedef Documentation

9.156.2.1 typedef std::pair<pattern *, const property *> cadabra::Properties::pat_prop_pair_t

9.156.2.2 typedef std::multimap<const property *, pattern *> cadabra::Properties::pattern_map_t

9.156.2.3 typedef std::multimap<nset_t::iterator, pat_prop_pair_t, nset_it_less> cadabra::Properties::property_map_t

We keep two multi-maps: one from the pattern to the property (roughly) and one from the property to the pattern.

These are both multi-maps because one pattern can have multiple properties assigned to it, and one property can be assigned to multiple properties.

When we delete properties, we check the pats map to see if the reference count for that property has dropped to zero.

9.156.3 Member Function Documentation

9.156.3.1 void Properties::clear ( )

9.156.3.2 template<class T > const T * cadabra::Properties::get (Ex::iterator *it,* bool *ignore_parent_rel* = `false`) const

9.156.3.3 template<class T > const T * cadabra::Properties::get ( ) const

9.156.3.4 template<class T > const T * cadabra::Properties::get_composite (Ex::iterator *it,* bool *ignore_parent_rel* = `false`) const

9.156.3.5 template<class T > const T * cadabra::Properties::get_composite (Ex::iterator *it,* int & *serialnum,* bool *doserial* = `true`, bool *ignore_parent_rel* = `false`) const

9.156.3.6 template<class T > const T * cadabra::Properties::get_composite (Ex::iterator *it,* const std::string & *label*) const

9.156.3.7 template<class T > const T * cadabra::Properties::get_composite (Ex::iterator *it,* int & *serialnum,* const std::string & *label,* bool *doserial* = `true`) const

9.156.3.8 template<class T > const T * cadabra::Properties::get_composite (Ex::iterator *it1,* Ex::iterator *it2,* bool *ignore_parent_rel* = `false`) const

9.156.3.9 template<class T > const T * cadabra::Properties::get_composite (Ex::iterator *it1,* Ex::iterator *it2,* int & *serialnum1,* int & *serialnum2,* bool *ignore_parent_rel* = `false`) const

9.156.3.10 bool Properties::has (const property * *pb,* Ex::iterator *it*)

9.156.3.11 template<class T > Ex::iterator cadabra::Properties::head (Ex::iterator *it,* bool *ignore_parent_rel* = `false`) const

9.156.3.12 void Properties::insert_list_prop (const std::vector< Ex > & *its,* const list_property * *pr*) `[private]`

9.156.3.13 void Properties::insert_prop (const Ex & *et,* const property * *pr*) `[private]`

9.156.3.14 std::string Properties::master_insert (Ex *proptree,* property * *thepropbase*)

Register a property for the indicated Ex.

Takes both normal and list properties and works out which insert calls to make. The property ownership is transferred to us on using this call.

9.156.3.15 void Properties::register_property (property * *)(,* const std::string & *name*)

Registering properties.

When inserting a property or list_property, ownership of the property gets transferred to this class.

9.156.3.16 int Properties::serial_number (const property * *listprop,* const pattern * *pat*) const

9.156.4 Member Data Documentation

9.156.4.1 pattern_map_t cadabra::Properties::pats

9.156.4.2 property_map_t cadabra::Properties::props

The following two maps own the pointers to the properties and patterns stored in them; use clear() to clean up.

Note that pointers can sit in in more than one entry in this map (when they are pointing to list_property objects, which are shared between patterns).

9.156.4.3 registered_property_map_t cadabra::Properties::registered_properties

The documentation for this class was generated from the following files:

- core/Props.hh
- core/Props.cc

9.157 Property< T > Class Template Reference

9.157.1 Description

template<class T>
class Property< T >

Property is a templated wrapper around a C++ property object.

It provides it with _latex, **str** and **repr** methods. In order to have a quick way to figure out in Python whether an object is a property, we derive it from BaseProperty, which is an empty placeholder (in Python BaseProperty is called Property).

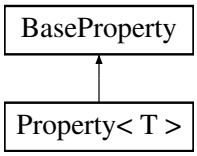
Properties can have arguments. These are not parsed by Python, but rather by the C++ side. There is a number of reasons for doing things this way. The most important of them is that this makes it a lot easier for Cadabra to provide useful feedback on parameters which are not valid. So all properties get initialised with two Cadabra Ex objects: the 1st is the pattern to which the property should be attached, the 2nd is the argument of the property, interpreted as a Cadabra expression.

Cadabra properties cannot be proper Python properties, because we need to give the latter names in order to prevent them from going out of scope. So Cadabra keeps a list of 'anonymous property objects in the current scope'.

The question is now what we do when Python keeps a pointer to these objects, and let that pointer escape local scope (e.g. by returning the Python property object). How do we keep it in scope?

`#include <PythonCdb.hh>`

Inheritance diagram for Property< T >:



Public Member Functions

- Property (std::shared_ptr< cadabra::Ex > obj, std::shared_ptr< cadabra::Ex > params=0)
- std::string str_ () const

  Human-readable form in text, i.e. no special formatting.
- std::string latex_ () const

  Human-readable form using LaTeX markup.
- std::string repr_ () const

  Python-parseable form. FIXME: not correct right now.
- template<>
  std::string latex_ () const

Private Attributes

- T * prop
- std::shared_ptr< cadabra::Ex > for_obj

9.157.2 Constructor & Destructor Documentation

9.157.2.1 template<class Prop > Property< Prop >::Property (std::shared_ptr< cadabra::Ex > *obj*, std::shared_ptr< cadabra::Ex > *params* = 0)

9.157.3 Member Function Documentation

9.157.3.1 template<class Prop > std::string Property< Prop >::latex_ ( ) const

Human-readable form using LaTeX markup.

9.157.3.2 template<> std::string Property< LaTeXForm >::latex_ ( ) const

9.157.3.3 template<class Prop > std::string Property< Prop >::repr_ ( ) const

Python-parseable form. FIXME: not correct right now.

9.157.3.4 template<class Prop > std::string Property< Prop >::str_ ( ) const

Human-readable form in text, i.e. no special formatting.

9.157.4 Member Data Documentation

9.157.4.1 template<class T > std::shared_ptr<cadabra::Ex> Property< T >::for_obj [private]

9.157.4.2 template<class T > T* Property< T >::prop [private]

The documentation for this class was generated from the following files:

- core/PythonCdb.hh
- core/PythonCdb.cc

9.158 cadabra::property Class Reference

9.158.1 Description

Base class for all properties, handling argument parsing and defining the interface.

Properties can have arguments. Parsing of these is done in the properties object itself, with the use of some helper functions. Parsing is done by implementing the virtual function property::parse(const Kernel&, keyval_t&). The argument is a container class which represents the arguments passed to the property as key/value pairs keyval_t type.

Properties will be asked to check that they can be associated to a given pattern through the virtual property↩ ::validate(const Properties&, const Ex&) function. The FIXME: the above two need to be merged, because parse may need access to the actual pattern tree, and once we are there, we may as well do checking. HOWEVER: in TableauSymmetry.cc: FIXME: we get the wrong pattern in case of a list! We should have been fed each individual item in the list, not the list itself.

This suggests that we should be calling once for every pattern, but that is wasteful in case we are just parsing arguments. Can we really avoid calling parse for every pattern?

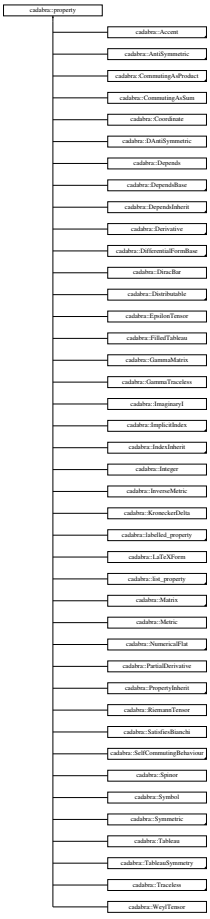
{A_{m n}, A_{m n p}, A_{m n p q}}::TableauSymmetry(shape={2,1}, indices={0,1,2}).

leads to a problem, because the property needs to setup its internal structures but also verify that these can match all objects in the same way.

Make all identical properties point to the same property object, so that normal and list properties become pretty much identical.

`#include <Props.hh>`

Inheritance diagram for cadabra::property:



Public Types

- enum match_t { no_match, id_match, exact_match }

Public Member Functions

- property (bool hidden=false)
- virtual ∼property ()
- bool parse_to_keyvals (const Ex &, keyval_t &)
- virtual bool parse (const Kernel &, keyval_t &keyvals)
- virtual void validate (const Kernel &, const Ex &) const
- virtual void latex (std::ostream &) const

  Display the property on the stream.
- virtual std::string name () const =0
- virtual std::string unnamed_argument () const
- virtual match_t equals (const property ∗) const
- void hidden (bool h)

  Properties can be hidden because they only make sense to the system; they will not be printed when the user asks for a list of properties.
- bool hidden (void) const

Private Member Functions

- bool parse_one_argument (Ex::iterator arg, keyval_t &keyvals)

Private Attributes

- bool hidden_

9.158.2 Member Enumeration Documentation

9.158.2.1 enum cadabra::property::match_t

Enumerator

no_match

id_match

exact_match

9.158.3 Constructor & Destructor Documentation

9.158.3.1 property::property (bool *hidden* = `false`)

9.158.3.2 virtual cadabra::property::∼property ( ) `[inline]`, `[virtual]`

9.158.4 Member Function Documentation

9.158.4.1 property::match_t property::equals (const property ∗) const `[virtual]`

Reimplemented in cadabra::Indices, cadabra::CommutingBehaviour, and cadabra::SortOrder.

9.158.4.2 void property::hidden (bool *h*)

Properties can be hidden because they only make sense to the system; they will not be printed when the user asks for a list of properties.

9.158.4.3 bool property::hidden (void) const

9.158.4.4 void property::latex (std::ostream & *str*) const [virtual]

Display the property on the stream.

Generate a LaTeX representation of the property, assuming LaTeX is in text mode (so it needs dollar symbols to switch to maths).

Reimplemented in cadabra::Indices, cadabra::Metric, cadabra::TableauSymmetry, cadabra::WeylTensor, cadabra::ImplicitIndex, and cadabra::GammaMatrix.

9.158.4.5 virtual std::string cadabra::property::name ( ) const [pure virtual]

Implemented in cadabra::PropertyInherit, cadabra::Accent, cadabra::Derivative, cadabra::WeightInherit, cadabra::Integer, cadabra::EpsilonTensor, cadabra::TableauSymmetry, cadabra::AntiSymmetric, cadabra::DifferentialForm, cadabra::NumericalFlat, cadabra::Weight, cadabra::DependsInherit, cadabra::ExteriorDerivative, cadabra::Indices, cadabra::InverseMetric, cadabra::Metric, cadabra::Spinor, cadabra::WeylTensor, cadabra::DAntiSymmetric, cadabra::Diagonal, cadabra::DiracBar, cadabra::Distributable, cadabra::GammaMatrix, cadabra::GammaTraceless, cadabra::ImplicitIndex, cadabra::KroneckerDelta, cadabra::Matrix, cadabra::PartialDerivative, cadabra::SatisfiesBianchi, cadabra::SelfCommuting, cadabra::Tableau, cadabra::Traceless, cadabra::AntiCommuting, cadabra::Commuting, cadabra::CommutingAsProduct, cadabra::CommutingAsSum, cadabra::Coordinate, cadabra::FilledTableau, cadabra::ImaginaryI, cadabra::IndexInherit, cadabra::LaTeXForm, cadabra::NonCommuting, cadabra::SelfAntiCommuting, cadabra::SelfNonCommuting, cadabra::SortOrder, cadabra::Symbol, cadabra::Symmetric, cadabra::RiemannTensor, and cadabra::Depends.

9.158.4.6 bool property::parse (const Kernel & , keyval_t & *keyvals*) [virtual]

Reimplemented in cadabra::labelled_property, cadabra::WeightInherit, cadabra::Integer, cadabra::DifferentialForm, cadabra::Spinor, cadabra::EpsilonTensor, cadabra::GammaMatrix, cadabra::InverseMetric, cadabra::Metric, cadabra::TableauSymmetry, cadabra::Indices, cadabra::LaTeXForm, cadabra::Weight, cadabra::ImplicitIndex, and cadabra::Depends.

9.158.4.7 bool property::parse_one_argument (Ex::iterator *arg,* keyval_t & *keyvals*) [private]

9.158.4.8 bool property::parse_to_keyvals (const Ex & *tr,* keyval_t & *keyvals*)

9.158.4.9 std::string property::unnamed_argument ( ) const [virtual]

Reimplemented in cadabra::Integer, cadabra::WeightInherit, cadabra::Indices, cadabra::ImplicitIndex, cadabra::LaTeXForm, cadabra::Weight, and cadabra::Depends.

9.158.4.10 void property::validate (const Kernel & , const Ex &) const [virtual]

Reimplemented in cadabra::InverseMetric, cadabra::Metric, cadabra::WeylTensor, and cadabra::RiemannTensor.

9.158.5 Member Data Documentation

9.158.5.1 bool cadabra::property::hidden_ [private]

The documentation for this class was generated from the following files:

- core/Props.hh
- core/Props.cc

9.159 cadabra::PropertyInherit Class Reference

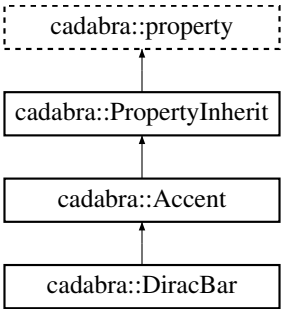
9.159.1 Description

PropertyInherit is like Inherit<T> for all properties.

This is very generic and almost never really useful.

```
#include <Props.hh>
```

Inheritance diagram for cadabra::PropertyInherit:



Public Member Functions

- virtual std::string name () const

Additional Inherited Members

9.159.2 Member Function Documentation

9.159.2.1 virtual std::string cadabra::PropertyInherit::name ( ) const [inline], [virtual]

Implements cadabra::property.

Reimplemented in cadabra::Accent, and cadabra::DiracBar.

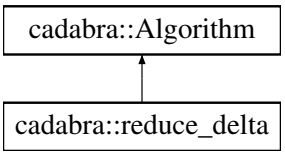
The documentation for this class was generated from the following file:

- core/Props.hh

9.160 cadabra::reduce_delta Class Reference

`#include <reduce_delta.hh>`

Inheritance diagram for cadabra::reduce_delta:



Public Member Functions

- reduce_delta (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Member Functions

- bool one_step_ (sibling_iterator)

Additional Inherited Members

9.160.1 Constructor & Destructor Documentation

9.160.1.1 reduce_delta::reduce_delta (const Kernel & *k*, Ex & *e*)

9.160.2 Member Function Documentation

9.160.2.1 Algorithm::result_t reduce_delta::apply (iterator & *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.160.2.2 bool reduce_delta::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.160.2.3 bool reduce_delta::one_step_ (sibling_iterator *dl*) [private]

The documentation for this class was generated from the following files:

- core/algorithms/reduce_delta.hh
- core/algorithms/reduce_delta.cc

9.161 cadabra::Properties::registered_property_map_t Class Reference

`#include <Props.hh>`

Public Types

- typedef std::map< std::string, property *(*)()> internal_property_map_t
- typedef internal_property_map_t::iterator iterator

Public Member Functions

- ~registered_property_map_t ()

Public Attributes

- internal_property_map_t store

9.161.1 Member Typedef Documentation

9.161.1.1 typedef std::map<std::string, property* (*)()> cadabra::Properties::registered_property_map_t::internal_property_map_t

9.161.1.2 typedef internal_property_map_t::iterator cadabra::Properties::registered_property_map_t::iterator

9.161.2 Constructor & Destructor Documentation

9.161.2.1 Properties::registered_property_map_t::~registered_property_map_t ( )

9.161.3 Member Data Documentation

9.161.3.1 internal_property_map_t cadabra::Properties::registered_property_map_t::store

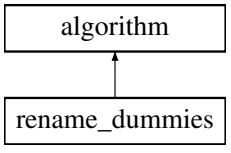
The documentation for this class was generated from the following files:

- core/Props.hh
- core/Props.cc

9.162 rename_dummies Class Reference

`#include <Dummies.hh>`

Inheritance diagram for rename_dummies:



Public Member Functions

- rename_dummies (Ex &, iterator)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

9.162.1 Constructor & Destructor Documentation

9.162.1.1 rename_dummies::rename_dummies (Ex & , iterator)

9.162.2 Member Function Documentation

9.162.2.1 virtual result_t rename_dummies::apply (iterator &) [virtual]

9.162.2.2 virtual bool rename_dummies::can_apply (iterator) [virtual]

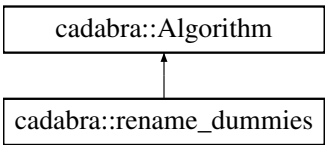
The documentation for this class was generated from the following file:

- core/Dummies.hh

9.163 cadabra::rename_dummies Class Reference

```
#include <rename_dummies.hh>
```

Inheritance diagram for cadabra::rename_dummies:



Public Member Functions

- rename_dummies (const Kernel &, Ex &, std::string, std::string)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Private Attributes

- std::string dset1
- std::string dset2

Additional Inherited Members

9.163.1 Constructor & Destructor Documentation

9.163.1.1 rename_dummies::rename_dummies (const Kernel & *k*, Ex & *tr*, std::string *d1*, std::string *d2*)

9.163.2 Member Function Documentation

9.163.2.1 Algorithm::result_t rename_dummies::apply (iterator & *st*) [virtual]

Implements cadabra::Algorithm.

9.163.2.2 bool rename_dummies::can_apply (iterator *st*) [virtual]

Implements cadabra::Algorithm.

9.163.3 Member Data Documentation

9.163.3.1 std::string cadabra::rename_dummies::dset1 [private]

9.163.3.2 std::string cadabra::rename_dummies::dset2 [private]

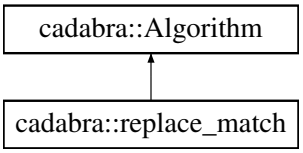
The documentation for this class was generated from the following files:

- core/algorithms/rename_dummies.hh
- core/algorithms/rename_dummies.cc

9.164 cadabra::replace_match Class Reference

`#include <replace_match.hh>`

Inheritance diagram for cadabra::replace_match:



Public Member Functions

- replace_match (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.164.1 Constructor & Destructor Documentation

9.164.1.1 replace_match::replace_match (const Kernel & *k,* Ex & *e*)

9.164.2 Member Function Documentation

9.164.2.1 Algorithm::result_t replace_match::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.164.2.2 bool replace_match::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

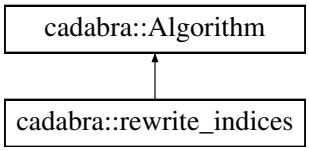
The documentation for this class was generated from the following files:

- core/algorithms/replace_match.hh
- core/algorithms/replace_match.cc

9.165 cadabra::rewrite_indices Class Reference

`#include <rewrite_indices.hh>`

Inheritance diagram for cadabra::rewrite_indices:



Public Member Functions

- rewrite_indices (const Kernel &, Ex &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Attributes

- Ex preferred
- Ex converters
- bool is_derivative_argument

Additional Inherited Members

9.165.1 Constructor & Destructor Documentation

9.165.1.1 rewrite_indices::rewrite_indices (const Kernel & *k*, Ex & *e*, Ex & *pref*, Ex & *conv*)

9.165.2 Member Function Documentation

9.165.2.1 Algorithm::result_t rewrite_indices::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.165.2.2 bool rewrite_indices::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.165.3 Member Data Documentation

9.165.3.1 Ex cadabra::rewrite_indices::converters [private]

9.165.3.2 bool cadabra::rewrite_indices::is_derivative_argument [private]

9.165.3.3 Ex cadabra::rewrite_indices::preferred [private]

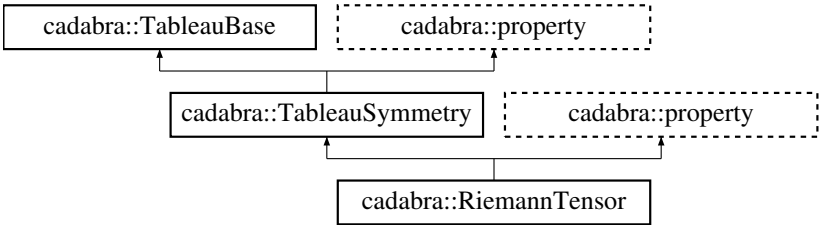
The documentation for this class was generated from the following files:

- core/algorithms/rewrite_indices.hh
- core/algorithms/rewrite_indices.cc

9.166 cadabra::RiemannTensor Class Reference

```
#include <RiemannTensor.hh>
```

Inheritance diagram for cadabra::RiemannTensor:



Public Member Functions

- RiemannTensor ()
- virtual std::string name () const override
- virtual void validate (const Kernel &, const Ex &) const override

Additional Inherited Members

9.166.1 Constructor & Destructor Documentation

9.166.1.1 RiemannTensor::RiemannTensor ( )

9.166.2 Member Function Documentation

9.166.2.1 std::string RiemannTensor::name ( ) const [override], [virtual]

Implements cadabra::property.

9.166.2.2 void RiemannTensor::validate (const Kernel & *kernel*, const Ex & *pat*) const [override], [virtual]

Reimplemented from cadabra::property.

The documentation for this class was generated from the following files:

- core/properties/RiemannTensor.hh
- core/properties/RiemannTensor.cc

9.167 RuntimeException Class Reference

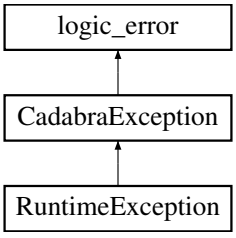
9.167.1 Description

Exception thrown when an algorithm cannot complete, e.g.

because it does not yet know how to process a particular expression. This gets thrown if returning the original expression would be a mathematical error.

```
#include <Exceptions.hh>
```

Inheritance diagram for RuntimeException:



Public Member Functions

- RuntimeException (std::string="")

9.167.2 Constructor & Destructor Documentation

9.167.2.1 RuntimeException::RuntimeException (std::string *s* = " ")

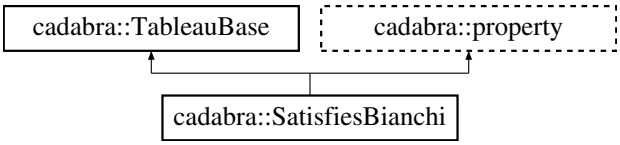
The documentation for this class was generated from the following files:

- core/Exceptions.hh
- core/Exceptions.cc

9.168 cadabra::SatisfiesBianchi Class Reference

`#include <SatisfiesBianchi.hh>`

Inheritance diagram for cadabra::SatisfiesBianchi:



Public Member Functions

- virtual std::string name () const override
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const override
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const override

Additional Inherited Members

9.168.1 Member Function Documentation

9.168.1.1 TableauBase::tab_t SatisfiesBianchi::get_tab (const Properties & *properties*, Ex & *tr*, Ex::iterator *it*, unsigned int *num*) const [override], [virtual]

Implements cadabra::TableauBase.

9.168.1.2 std::string SatisfiesBianchi::name ( ) const [override], [virtual]

Implements cadabra::property.

9.168.1.3 unsigned int SatisfiesBianchi::size (const Properties & *properties,* Ex & *tr,* Ex::iterator *it*) const [override], [virtual]

Implements cadabra::TableauBase.

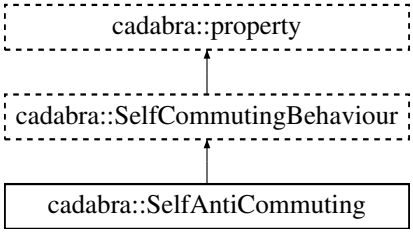
The documentation for this class was generated from the following files:

- core/properties/SatisfiesBianchi.hh
- core/properties/SatisfiesBianchi.cc

9.169 cadabra::SelfAntiCommuting Class Reference

`#include <SelfAntiCommuting.hh>`

Inheritance diagram for cadabra::SelfAntiCommuting:



Public Member Functions

- virtual std::string name () const override
- virtual int sign () const override

Additional Inherited Members

9.169.1 Member Function Documentation

9.169.1.1 std::string SelfAntiCommuting::name ( ) const [override], [virtual]

Implements cadabra::property.

9.169.1.2 int SelfAntiCommuting::sign ( ) const [override], [virtual]

Implements cadabra::SelfCommutingBehaviour.

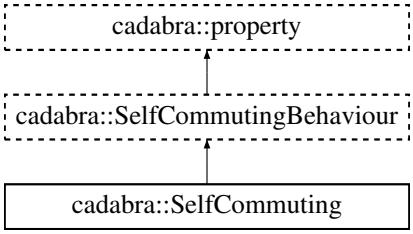
The documentation for this class was generated from the following files:

- core/properties/SelfAntiCommuting.hh
- core/properties/SelfAntiCommuting.cc

9.170 cadabra::SelfCommuting Class Reference

`#include <SelfCommuting.hh>`

Inheritance diagram for cadabra::SelfCommuting:



Public Member Functions

- virtual std::string name () const override
- virtual int sign () const override

Additional Inherited Members

9.170.1 Member Function Documentation

9.170.1.1 std::string SelfCommuting::name ( ) const [override], [virtual]

Implements cadabra::property.

9.170.1.2 int SelfCommuting::sign ( ) const [override], [virtual]

Implements cadabra::SelfCommutingBehaviour.

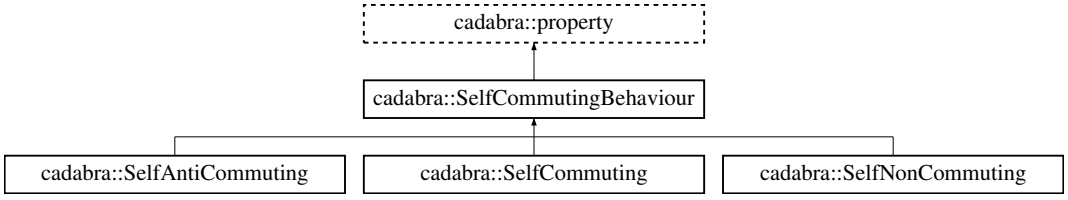
The documentation for this class was generated from the following files:

- core/properties/SelfCommuting.hh
- core/properties/SelfCommuting.cc

9.171 cadabra::SelfCommutingBehaviour Class Reference

`#include <SelfCommutingBehaviour.hh>`

Inheritance diagram for cadabra::SelfCommutingBehaviour:



Public Member Functions

- virtual int sign () const =0

Additional Inherited Members

9.171.1 Member Function Documentation

9.171.1.1 virtual int cadabra::SelfCommutingBehaviour::sign ( ) const [pure virtual]

Implemented in cadabra::SelfCommuting, cadabra::SelfAntiCommuting, and cadabra::SelfNonCommuting.

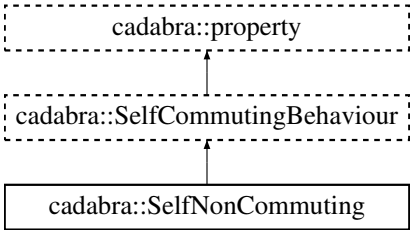
The documentation for this class was generated from the following file:

- core/properties/SelfCommutingBehaviour.hh

9.172 cadabra::SelfNonCommuting Class Reference

`#include <SelfNonCommuting.hh>`

Inheritance diagram for cadabra::SelfNonCommuting:



Public Member Functions

- virtual std::string name () const override
- virtual int sign () const override

Additional Inherited Members

9.172.1 Member Function Documentation

9.172.1.1 std::string SelfNonCommuting::name ( ) const [override], [virtual]

Implements cadabra::property.

9.172.1.2 int SelfNonCommuting::sign ( ) const [override], [virtual]

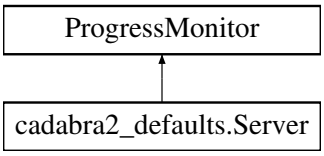
Implements cadabra::SelfCommutingBehaviour.

The documentation for this class was generated from the following files:

- core/properties/SelfNonCommuting.hh
- core/properties/SelfNonCommuting.cc

9.173 cadabra2_defaults.Server Class Reference

Inheritance diagram for cadabra2_defaults.Server:



Public Member Functions

- def send (self, data, typestr)
- def architecture (self)
- def test (self)
- def handles (self, otype)

9.173.1 Member Function Documentation

9.173.1.1 def cadabra2_defaults.Server.architecture (*self*)

9.173.1.2 def cadabra2_defaults.Server.handles (*self, otype*)

9.173.1.3 def cadabra2_defaults.Server.send (*self, data, typestr*)

9.173.1.4 def cadabra2_defaults.Server.test (*self*)

The documentation for this class was generated from the following file:

- core/cadabra2_defaults.py

9.174 Server Class Reference

9.174.1 Description

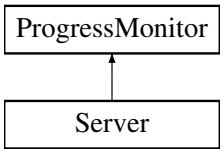
Object representing a Cadabra server, capable of receiving messages on a websocket, running Python code, and sending output back to the client.

Contains the logic to intercept raw Python output but also provides functionality to the Python side which enables clients to send various objects in JSON encoded form. See Server::on_block_finished for the format of these messages.

Every block is run on the same Python scope. They run sequentially, one at a time, on a thread separate from the websocket++ main loop. When the Python code finishes (or when it is interrupted), this thread locks the socket_↩ mutex and calls on_block_finished().

`#include <Server.hh>`

Inheritance diagram for Server:



Classes

- class Block
- class CatchOutput

  Python output catching.
- class Connection

Public Member Functions

- Server ()
- Server (const Server &)=delete
- Server (const std::string &socket)
- ~Server ()
- void run ()

  The only user-visible part: just instantiate a server object and start it with run().

Public Attributes

- CatchOutput catchOut
- CatchOutput catchErr
- Stopwatch server_stopwatch
- Stopwatch sympy_stopwatch

Private Types

- typedef websocketpp::server< websocketpp::config::asio > WebsocketServer
- typedef std::map< websocketpp::connection_hdl, Connection, std::owner_less< websocketpp::connection↩ _hdl > > ConnectionMap

Private Member Functions

- void init ()
- void on_socket_init (websocketpp::connection_hdl hdl, boost::asio::ip::tcp::socket &s)
- void on_message (websocketpp::connection_hdl hdl, WebsocketServer::message_ptr msg)
- void on_open (websocketpp::connection_hdl hdl)
- void on_close (websocketpp::connection_hdl hdl)
- void wait_for_job ()
- std::string run_string (const std::string &, bool handle_output=true)
- void on_block_finished (Block)

  Called by the run_block() thread upon completion of the task.
- void on_block_error (Block)
- void on_kernel_fault (Block)
- std::string architecture () const
- bool handles (const std::string &otype) const
- void send (const std::string &output, const std::string &msg_type)

  Raw code to send a string (which must be JSON formatted) as a message to the client.
- void send_json (const std::string &)
- void stop_block ()

  Halt the currently running block and prevent execution of any further blocks that may still be on the queue.
- void dispatch_message (websocketpp::connection_hdl, const std::string &json_string)

  Takes a JSON encoded message and performs the required action to process it.

Private Attributes

- WebsocketServer wserver
- std::string socket_name
- ConnectionMap connections
- std::mutex ws_mutex
- std::thread runner
- std::mutex block_available_mutex
- std::condition_variable block_available
- std::queue< Block > block_queue
- websocketpp::connection_hdl current_hdl
- uint64_t current_id
- bool started
- std::future< std::string > job
- boost::python::object main_module
- boost::python::object main_namespace
- int cells_ran

9.174.2 Member Typedef Documentation

9.174.2.1 typedef std::map<websocketpp::connection_hdl, Connection, std::owner_less<websocketpp::connection_hdl> > Server::ConnectionMap [private]

9.174.2.2 typedef websocketpp::server<websocketpp::config::asio> Server::WebsocketServer [private]

9.174.3 Constructor & Destructor Documentation

9.174.3.1 Server::Server ( )

9.174.3.2 Server::Server (const Server &) [delete]

9.174.3.3 Server::Server (const std::string & *socket*)

9.174.3.4 Server::~Server ( )

9.174.4 Member Function Documentation

9.174.4.1 std::string Server::architecture ( ) const [private]

9.174.4.2 void Server::dispatch_message (websocketpp::connection_hdl *hdl*, const std::string & *json_string*) [private]

Takes a JSON encoded message and performs the required action to process it.

Where applicable these messages are compatible with IPython's message types, http://ipython.org/ipython-doc/dev/development/messaging.html

9.174.4.3 bool Server::handles (const std::string & *otype*) const [private]

9.174.4.4 void Server::init ( ) [private]

9.174.4.5 void Server::on_block_error (Block *blk*) [private]

9.174.4.6 void Server::on_block_finished (Block *blk*) [private]

Called by the run_block() thread upon completion of the task.

This will send any output generated by printing directly to stdout or stderr from Python (so, output not generated by using the 'display' function). Indicates to the client that this block has finished executing. Will send an empty string if there has been no output 'print'ed.

9.174.4.7 void Server::on_close (websocketpp::connection_hdl *hdl*) [private]

9.174.4.8 void Server::on_kernel_fault (Block *blk*) [private]

9.174.4.9 void Server::on_message (websocketpp::connection_hdl *hdl,* WebsocketServer::message_ptr *msg*) [private]

9.174.4.10 void Server::on_open (websocketpp::connection_hdl *hdl*) [private]

9.174.4.11 void Server::on_socket_init (websocketpp::connection_hdl *hdl,* boost::asio::ip::tcp::socket & *s*) [private]

9.174.4.12 void Server::run ( )

The only user-visible part: just instantiate a server object and start it with run().

This will not return until the server has been shut down.

9.174.4.13 std::string Server::run_string (const std::string & *blk,* bool *handle_output* = true) [private]

9.174.4.14 void Server::send (const std::string & *output,* const std::string & *msg_type*) [private]

Raw code to send a string (which must be JSON formatted) as a message to the client.

Handles communication of the result back to the client in JSON format. This is always of the form

```
{ "header":   { "parent_id":     "...",
                "parent_origin": "client" | "server",
                "cell_id":       "...",
                "cell_origin":   "client" | "server"
```

}, "content": { "output": "..." }, "msg_type": "..." } msg_type can be "output", "latex", "image_png" and so on, corresponding to the possible values of DataCell::CellType.

9.174.4.15 void Server::send_json (const std::string & *msg*) [private]

9.174.4.16 void Server::stop_block ( ) [private]

Halt the currently running block and prevent execution of any further blocks that may still be on the queue.

9.174.4.17 void Server::wait_for_job ( ) [private]

9.174.5 Member Data Documentation

9.174.5.1 std::condition_variable Server::block_available [private]

9.174.5.2 std::mutex Server::block_available_mutex [private]

9.174.5.3 std::queue<Block> Server::block_queue [private]

9.174.5.4 CatchOutput Server::catchErr

9.174.5.5 CatchOutput Server::catchOut

9.174.5.6 int Server::cells_ran [private]

9.174.5.7 ConnectionMap Server::connections [private]

9.174.5.8 websocketpp::connection_hdl Server::current_hdl [private]

9.174.5.9 uint64_t Server::current_id [private]

9.174.5.10 std::future<std::string> Server::job [private]

9.174.5.11 boost::python::object Server::main_module [private]

9.174.5.12 boost::python::object Server::main_namespace [private]

9.174.5.13 std::thread Server::runner [private]

9.174.5.14 Stopwatch Server::server_stopwatch

9.174.5.15 std::string Server::socket_name [private]

9.174.5.16 bool Server::started [private]

9.174.5.17 Stopwatch Server::sympy_stopwatch

9.174.5.18 std::mutex Server::ws_mutex [private]

9.174.5.19 WebsocketServer Server::wserver [private]

The documentation for this class was generated from the following files:

- client_server/Server.hh
- client_server/Server.cc

9.175 ServerWrapper Class Reference

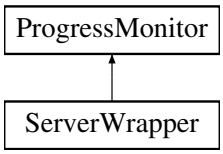
9.175.1 Description

A class which enables access to a Python-side defined Server object with ProgressMonitor interface.

It forwards calls on the group/progress members via Boost.Python to the Python defined Server object, thereby allowing the ProgressMonitor functionality to be exposed to Cadabra algorithms through a a single C++ interface, independent of whether it is implemented in the C++ Server object or in the Python one.

`#include <ServerWrapper.hh>`

Inheritance diagram for ServerWrapper:



Public Member Functions

- ServerWrapper ()
- virtual void group (std::string name="") override
- virtual void progress (int n, int total) override

Private Attributes

- boost::python::object server
- boost::python::object python_group
- boost::python::object python_progress

9.175.2 Constructor & Destructor Documentation

9.175.2.1 ServerWrapper::ServerWrapper ( )

9.175.3 Member Function Documentation

9.175.3.1 void ServerWrapper::group (std::string *name* = " ") [override], [virtual]

Reimplemented from ProgressMonitor.

9.175.3.2 void ServerWrapper::progress (int *n*, int *total*) [override], [virtual]

Reimplemented from ProgressMonitor.

9.175.4 Member Data Documentation

9.175.4.1 boost::python::object ServerWrapper::python_group [private]

9.175.4.2 boost::python::object ServerWrapper::python_progress [private]

9.175.4.3 boost::python::object ServerWrapper::server [private]

The documentation for this class was generated from the following files:

- core/ServerWrapper.hh
- core/ServerWrapper.cc

9.176 snoop::Snoop Class Reference

9.176.1 Description

Logging class with functionality to send log information to a remote server using a websocket connection.

`#include <Snoop.hh>`

Classes

- class AppEntry

  C++ representation of an run entry.
- class LogEntry

  C++ representation of a log entry.

Public Member Functions

- Snoop ()
- ~Snoop ()
- void init (const std::string &app_name, const std::string &app_version, std::string server="", std::string local_log_file="")

  Initialise the logging stream.
- std::string get_user_uuid (const std::string &app_name)

  Get a string which uniquely identifies the current user.
- Snoop & operator() (const std::string &type, std::string fl="", int loc=-1, std::string method="")

  Operator to initialise a logging entry with the type of the log message as well as (optionally) the file, line number and method.
- template<class T >
  Snoop & operator<< (const T &obj)

  Generic operator to log an object to the log message being constructed.
- Snoop & operator<< (const Flush &)

  Flush the log entry to disk/server.
- void set_sync_immediately (bool)

  Set to sync with server after every log line.
- void sync_with_server (bool from_wsthread=false)

  Ensure that the local database is synchronised with the server (this sends multiple app or log entries in one websocket message).
- void sync_runs_with_server (bool from_wsthread=false)

  As above, but only for run entries.
- void sync_logs_with_server (bool from_wsthread=false)

  As above, but only for log entries.

Protected Attributes

- std::ostringstream out_
- bool sync_immediately_
- SnoopImpl * impl
- friend SnoopImpl

9.176.2 Constructor & Destructor Documentation

9.176.2.1 Snoop::Snoop ( )

9.176.2.2 Snoop::~Snoop ( )

9.176.3 Member Function Documentation

9.176.3.1 std::string Snoop::get_user_uuid (const std::string & *app_name*)

Get a string which uniquely identifies the current user.

This is stored in ~/.config/snoop/appname.conf, and in the 'user_id' field in each LogEntry. Note that this is different from the 'uuid' field, which will change from one run to the next.

9.176.3.2 void Snoop::init (const std::string & *app_name*, const std::string & *app_version*, std::string *server* = " ", std::string *local_log_file* = " ")

Initialise the logging stream.

Should be called once at program startup, but can be called multiple times without causing problems.

9.176.3.3 Snoop & Snoop::operator() (const std::string & *type*, std::string *fl* = " ", int *loc* = -1, std::string *method* = " ")

Operator to initialise a logging entry with the type of the log message as well as (optionally) the file, line number and method.

9.176.3.4 template<class T > Snoop& snoop::Snoop::operator<< (const T & *obj*) [inline]

Generic operator to log an object to the log message being constructed.

9.176.3.5 Snoop & Snoop::operator<< (const Flush & *f*)

Flush the log entry to disk/server.

9.176.3.6 void Snoop::set_sync_immediately (bool *s*)

Set to sync with server after every log line.

9.176.3.7 void Snoop::sync_logs_with_server (bool *from_wsthread* = `false`)

As above, but only for log entries.

9.176.3.8 void Snoop::sync_runs_with_server (bool *from_wsthread* = `false`)

As above, but only for run entries.

9.176.3.9 void Snoop::sync_with_server (bool *from_wsthread* = `false`)

Ensure that the local database is synchronised with the server (this sends multiple app or log entries in one web-socket message).

Leave the bool argument at its default argument under all normal circumstances.

9.176.4 Member Data Documentation

9.176.4.1 SnoopImpl* snoop::Snoop::impl [protected]

9.176.4.2 std::ostringstream snoop::Snoop::out_ [protected]

9.176.4.3 friend snoop::Snoop::SnoopImpl [protected]

9.176.4.4 bool snoop::Snoop::sync_immediately_ [protected]

The documentation for this class was generated from the following files:

- client_server/Snoop.hh
- client_server/Snoop.cc

9.177 snoop::SnoopImpl Class Reference

`#include <SnoopPrivate.hh>`

Public Member Functions

- SnoopImpl (Snoop *)
- ~SnoopImpl ()
- Snoop & operator<< (const Flush &)
- void init (const std::string &app_name, const std::string &app_version, std::string server="", std::string local_log_file="")

  Initialise the logging stream.
- std::string get_user_uuid (const std::string &app_name)

  Get a string which uniquely identifies the current user.
- Snoop & operator() (const std::string &type, std::string fl="", int loc=-1, std::string method="")

  Operator to initialise a logging entry with the type of the log message as well as (optionally) the file, line number and method.
- void sync_with_server (bool from_wsthread=false)

  Ensure that the local database is synchronised with the server (this sends multiple app or log entries in one websocket message).
- void sync_runs_with_server (bool from_wsthread=false)

  As above, but only for run entries.
- void sync_logs_with_server (bool from_wsthread=false)

  As above, but only for log entries.
- void start_websocket_client ()

  Start the websocket client.
- void create_tables ()

  Ensure that the required tables are present in the database file.
- void obtain_uuid ()

  Obtain a uuid by finding the last AppEntry stored in the local database.
- bool store_app_entry (Snoop::AppEntry &)

  Store an app entry in the database.
- bool store_app_entry_without_lock (Snoop::AppEntry &)
- bool store_log_entry (Snoop::LogEntry &, bool avoid_server_duplicates)

  Store a log entry in the local database.
- std::vector< Snoop::AppEntry > get_app_registrations (std::string uuid_filter="")

  Return a vector of all aps registered in the database.
- void on_client_open (websocketpp::connection_hdl hdl)
- void on_client_fail (websocketpp::connection_hdl hdl)
- void on_client_close (websocketpp::connection_hdl hdl)
- void on_client_message (websocketpp::connection_hdl hdl, message_ptr msg)

Public Attributes

- Snoop * snoop_
- sqlite3 * db
- Snoop::AppEntry this_app_
- Snoop::LogEntry this_log_
- std::string server_
- sqlite3_stmt * insert_statement

  Prepared statements (only need to prepare these once).
- sqlite3_stmt * id_for_uuid_statement
- std::mutex sqlite_mutex
- WebsocketClient wsclient

  Websocket client to talk to a remote logging server.
- std::thread wsclient_thread

- std::mutex connection_mutex
- std::condition_variable connection_cv
- bool connection_is_open
- bool connection_attempt_failed
- WebsocketClient::connection_ptr connection
- websocketpp::connection_hdl our_connection_hdl

Private Attributes

- std::mutex call_mutex

9.177.1 Constructor & Destructor Documentation

9.177.1.1 SnoopImpl::SnoopImpl (Snoop * *s*)

9.177.1.2 SnoopImpl::~SnoopImpl ( )

9.177.2 Member Function Documentation

9.177.2.1 void SnoopImpl::create_tables ( )

Ensure that the required tables are present in the database file.

9.177.2.2 std::vector< Snoop::AppEntry > SnoopImpl::get_app_registrations (std::string *uuid_filter* = " ")

Return a vector of all aps registered in the database.

If the uuid filter is non-empty, will filter on the given uuid.

9.177.2.3 std::string SnoopImpl::get_user_uuid (const std::string & *app_name*)

Get a string which uniquely identifies the current user.

This is stored in ~/.config/snoop/appname.conf.

9.177.2.4 void SnoopImpl::init (const std::string & *app_name*, const std::string & *app_version*, std::string *server* = " ", std::string *local_log_file* = " ")

Initialise the logging stream.

Should be called once at program startup, but can be called multiple times without causing problems.

9.177.2.5 void SnoopImpl::obtain_uuid ( )

Obtain a uuid by finding the last AppEntry stored in the local database.

Will attempt to re-turn a previously generated uuid but will do so only if one is stored for the current pid; if no entry with the current pid is stored then a new one will always be generated.

9.177.2.6 void SnoopImpl::on_client_close (websocketpp::connection_hdl *hdl*)

9.177.2.7 void SnoopImpl::on_client_fail (websocketpp::connection_hdl *hdl*)

9.177.2.8 void SnoopImpl::on_client_message (websocketpp::connection_hdl *hdl,* message_ptr *msg*)

9.177.2.9 void SnoopImpl::on_client_open (websocketpp::connection_hdl *hdl*)

9.177.2.10 Snoop & SnoopImpl::operator() (const std::string & *type,* std::string *fl* = " ", int *loc* = -1, std::string *method* = " ")

Operator to initialise a logging entry with the type of the log message as well as (optionally) the file, line number and method.

Thread-safe: can be called from different threads at the same time.

9.177.2.11 Snoop & SnoopImpl::operator<< (const Flush &)

9.177.2.12 void SnoopImpl::start_websocket_client ( )

Start the websocket client.

This tries to connect to the server and then waits in a separate thread until the server sends us something (typically in response to something the main thread makes by calling wsclient.send).

9.177.2.13 bool SnoopImpl::store_app_entry (Snoop::AppEntry & *app*)

Store an app entry in the database.

Will update the 'id' field in the AppEntry.

9.177.2.14 bool SnoopImpl::store_app_entry_without_lock (Snoop::AppEntry & *app*)

9.177.2.15 bool SnoopImpl::store_log_entry (Snoop::LogEntry & *log_entry,* bool *avoid_server_duplicates*)

Store a log entry in the local database.

Generates its own receive_millis field (the one given gets overwritten). Will update the 'id' field in the LogEntry. Returns 'true' if the entry was stored, or 'false' if an entry with this client_log_id was already present (except when it is 0).

9.177.2.16 void SnoopImpl::sync_logs_with_server (bool *from_wsthread* = `false`)

As above, but only for log entries.

9.177.2.17 void SnoopImpl::sync_runs_with_server (bool *from_wsthread* = `false`)

As above, but only for run entries.

9.177.2.18 void SnoopImpl::sync_with_server (bool *from_wsthread* = `false`)

Ensure that the local database is synchronised with the server (this sends multiple app or log entries in one websocket message).

Leave the bool argument at its default argument under all normal circumstances.

9.177.3 Member Data Documentation

9.177.3.1 std::mutex snoop::SnoopImpl::call_mutex `[private]`

9.177.3.2 WebsocketClient::connection_ptr snoop::SnoopImpl::connection

9.177.3.3 bool snoop::SnoopImpl::connection_attempt_failed

9.177.3.4 std::condition_variable snoop::SnoopImpl::connection_cv

9.177.3.5 bool snoop::SnoopImpl::connection_is_open

9.177.3.6 std::mutex snoop::SnoopImpl::connection_mutex

9.177.3.7 sqlite3* snoop::SnoopImpl::db

9.177.3.8 sqlite3_stmt * snoop::SnoopImpl::id_for_uuid_statement

9.177.3.9 sqlite3_stmt* snoop::SnoopImpl::insert_statement

Prepared statements (only need to prepare these once).

9.177.3.10 websocketpp::connection_hdl snoop::SnoopImpl::our_connection_hdl

9.177.3.11 std::string snoop::SnoopImpl::server_

9.177.3.12 Snoop* snoop::SnoopImpl::snoop_

9.177.3.13 std::mutex snoop::SnoopImpl::sqlite_mutex

9.177.3.14 Snoop::AppEntry snoop::SnoopImpl::this_app_

9.177.3.15 Snoop::LogEntry snoop::SnoopImpl::this_log_

9.177.3.16 WebsocketClient snoop::SnoopImpl::wsclient

Websocket client to talk to a remote logging server.

9.177.3.17 std::thread snoop::SnoopImpl::wsclient_thread

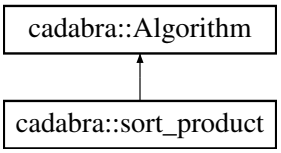
The documentation for this class was generated from the following files:

- client_server/SnoopPrivate.hh
- client_server/Snoop.cc

9.178 cadabra::sort_product Class Reference

`#include <sort_product.hh>`

Inheritance diagram for cadabra::sort_product:



Public Member Functions

- sort_product (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Private Attributes

- bool ignore_numbers_

Additional Inherited Members

9.178.1 Constructor & Destructor Documentation

9.178.1.1 sort_product::sort_product (const Kernel & *k*, Ex & *tr*)

9.178.2 Member Function Documentation

9.178.2.1 Algorithm::result_t sort_product::apply (iterator & *st*) [virtual]

Implements cadabra::Algorithm.

9.178.2.2 bool sort_product::can_apply (iterator *st*) [virtual]

Implements cadabra::Algorithm.

9.178.3 Member Data Documentation

9.178.3.1 bool cadabra::sort_product::ignore_numbers_ [private]

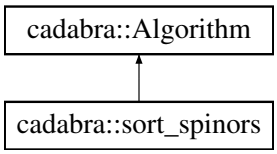
The documentation for this class was generated from the following files:

- core/algorithms/sort_product.hh
- core/algorithms/sort_product.cc

9.179 cadabra::sort_spinors Class Reference

`#include <sort_spinors.hh>`

Inheritance diagram for cadabra::sort_spinors:



Public Member Functions

- sort_spinors (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Public Attributes

- sibling_iterator one
- sibling_iterator gammamat
- sibling_iterator two

Additional Inherited Members

9.179.1 Constructor & Destructor Documentation

9.179.1.1 sort_spinors::sort_spinors (const Kernel & *k*, Ex & *e*)

9.179.2 Member Function Documentation

9.179.2.1 Algorithm::result_t sort_spinors::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.179.2.2 bool sort_spinors::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.179.3 Member Data Documentation

9.179.3.1 sibling_iterator cadabra::sort_spinors::gammamat

9.179.3.2 sibling_iterator cadabra::sort_spinors::one

9.179.3.3 sibling_iterator cadabra::sort_spinors::two

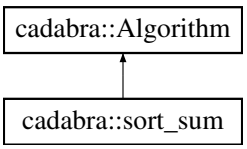
The documentation for this class was generated from the following files:

- core/algorithms/sort_spinors.hh
- core/algorithms/sort_spinors.cc

9.180 cadabra::sort_sum Class Reference

`#include <sort_sum.hh>`

Inheritance diagram for cadabra::sort_sum:



Public Member Functions

- sort_sum (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Member Functions

- bool should_swap (iterator obj, int subtree_comparison) const

Additional Inherited Members

9.180.1 Constructor & Destructor Documentation

9.180.1.1 sort_sum::sort_sum (const Kernel & *k*, Ex & *e*)

9.180.2 Member Function Documentation

9.180.2.1 Algorithm::result_t sort_sum::apply (iterator & *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.180.2.2 bool sort_sum::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.180.2.3 bool sort_sum::should_swap (iterator *obj*, int *subtree_comparison*) const [private]

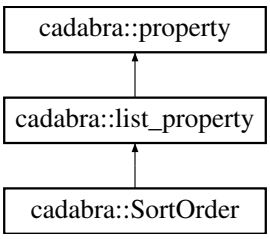
The documentation for this class was generated from the following files:

- core/algorithms/sort_sum.hh
- core/algorithms/sort_sum.cc

9.181 cadabra::SortOrder Class Reference

`#include <SortOrder.hh>`

Inheritance diagram for cadabra::SortOrder:



Public Member Functions

- virtual std::string name () const
- virtual match_t equals (const property *) const

Additional Inherited Members

9.181.1 Member Function Documentation

9.181.1.1 property::match_t SortOrder::equals (const property *) const [virtual]

Reimplemented from cadabra::property.

9.181.1.2 std::string SortOrder::name ( ) const [virtual]

Implements cadabra::property.

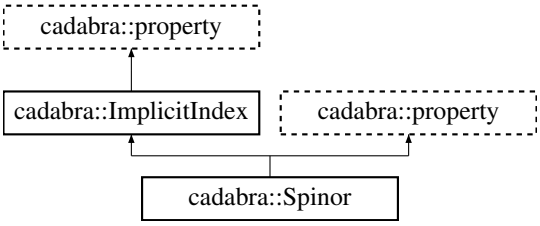
The documentation for this class was generated from the following files:

- core/properties/SortOrder.hh
- core/properties/SortOrder.cc

9.182 cadabra::Spinor Class Reference

`#include <Spinor.hh>`

Inheritance diagram for cadabra::Spinor:



Public Types

- enum { positive, negative }

Public Member Functions

- Spinor ()
- virtual ~Spinor ()
- virtual std::string name () const override
- virtual bool parse (const Kernel &, keyval_t &keyvals) override

Public Attributes

- int dimension
- bool weyl
- enum cadabra::Spinor:: { ... } chirality
- bool majorana

9.182.1 Member Enumeration Documentation

9.182.1.1 anonymous enum

Enumerator

positive

negative

9.182.2 Constructor & Destructor Documentation

9.182.2.1 Spinor::Spinor ( )

9.182.2.2 virtual cadabra::Spinor::~Spinor ( ) [inline], [virtual]

9.182.3 Member Function Documentation

9.182.3.1 std::string Spinor::name ( ) const [override], [virtual]

Implements cadabra::property.

9.182.3.2 bool Spinor::parse (const Kernel & *kernel,* keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

9.182.4 Member Data Documentation

9.182.4.1 enum { ... } cadabra::Spinor::chirality

9.182.4.2 int cadabra::Spinor::dimension

9.182.4.3 bool cadabra::Spinor::majorana

9.182.4.4 bool cadabra::Spinor::weyl

The documentation for this class was generated from the following files:

- core/properties/Spinor.hh
- core/properties/Spinor.cc

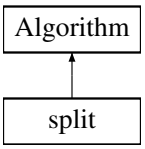
9.183 split Class Reference

9.183.1 Description

Split an expression into terms or factors.

`#include <split.hh>`

Inheritance diagram for split:



Public Member Functions

- split (const Kernel &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

9.183.2 Constructor & Destructor Documentation

9.183.2.1 split::split (const Kernel & , Ex &)

9.183.3 Member Function Documentation

9.183.3.1 Algorithm::result_t split::apply (iterator & *sum*) [virtual]

9.183.3.2 bool split::can_apply (iterator *st*) [virtual]

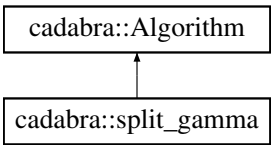
The documentation for this class was generated from the following files:

- core/algorithms/split.hh
- core/algorithms/split.cc

9.184 cadabra::split_gamma Class Reference

`#include <split_gamma.hh>`

Inheritance diagram for cadabra::split_gamma:



Public Member Functions

- split_gamma (const Kernel &, Ex &, bool on_back)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Public Attributes

- bool on_back

Additional Inherited Members

9.184.1 Constructor & Destructor Documentation

9.184.1.1 split_gamma::split_gamma (const Kernel & *k,* Ex & *e,* bool *on_back*)

9.184.2 Member Function Documentation

9.184.2.1 Algorithm::result_t split_gamma::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.184.2.2 bool split_gamma::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.184.3 Member Data Documentation

9.184.3.1 bool cadabra::split_gamma::on_back

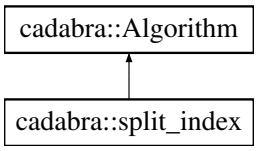
The documentation for this class was generated from the following files:

- core/algorithms/split_gamma.hh
- core/algorithms/split_gamma.cc

9.185 cadabra::split_index Class Reference

`#include <split_index.hh>`

Inheritance diagram for cadabra::split_index:



Public Member Functions

- split_index (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Private Attributes

- const Indices * full_class
- const Indices * part1_class
- const Indices * part2_class
- const Coordinate * part1_coord
- const Coordinate * part2_coord
- bool part1_is_number
- bool part2_is_number
- long num1
- long num2
- iterator part1_coord_node
- iterator part2_coord_node

Additional Inherited Members

9.185.1 Constructor & Destructor Documentation

9.185.1.1 split_index::split_index (const Kernel & *k*, Ex & *tr*, Ex & *triple*)

9.185.2 Member Function Documentation

9.185.2.1 Algorithm::result_t split_index::apply (iterator & *it*) [virtual]

Implements cadabra::Algorithm.

9.185.2.2 bool split_index::can_apply (iterator *it*) [virtual]

Implements cadabra::Algorithm.

9.185.3 Member Data Documentation

9.185.3.1 const Indices* cadabra::split_index::full_class [private]

9.185.3.2 long cadabra::split_index::num1 [private]

9.185.3.3 long cadabra::split_index::num2 [private]

9.185.3.4 const Indices * cadabra::split_index::part1_class [private]

9.185.3.5 const Coordinate* cadabra::split_index::part1_coord [private]

9.185.3.6 iterator cadabra::split_index::part1_coord_node [private]

9.185.3.7 bool cadabra::split_index::part1_is_number [private]

9.185.3.8 const Indices * cadabra::split_index::part2_class [private]

9.185.3.9 const Coordinate * cadabra::split_index::part2_coord [private]

9.185.3.10 iterator cadabra::split_index::part2_coord_node [private]

9.185.3.11 bool cadabra::split_index::part2_is_number [private]

The documentation for this class was generated from the following files:

- core/algorithms/split_index.hh
- core/algorithms/split_index.cc

9.186 Stopwatch Class Reference

`#include <Stopwatch.hh>`

Public Member Functions

- Stopwatch ()
- void reset ()
- void start ()
- void stop ()
- long seconds () const
- long useconds () const
- bool stopped () const

Private Member Functions

- void checkpoint_ () const

Private Attributes

- struct timeval tv1 tv2
- struct timezone tz
- long diffsec
- long diffusec
- bool stopped_

Friends

- std::ostream & operator<< (std::ostream &, const Stopwatch &)

9.186.1 Constructor & Destructor Documentation

9.186.1.1 Stopwatch::Stopwatch ( )

9.186.2 Member Function Documentation

9.186.2.1 void Stopwatch::checkpoint_ ( ) const [private]

9.186.2.2 void Stopwatch::reset ( )

9.186.2.3 long Stopwatch::seconds ( ) const

9.186.2.4 void Stopwatch::start ( )

9.186.2.5 void Stopwatch::stop ( )

9.186.2.6 bool Stopwatch::stopped ( ) const

9.186.2.7 long Stopwatch::useconds ( ) const

9.186.3 Friends And Related Function Documentation

9.186.3.1 std::ostream& operator<< (std::ostream & , const Stopwatch &) [friend]

9.186.4 Member Data Documentation

9.186.4.1 long Stopwatch::diffsec [mutable], [private]

9.186.4.2 long Stopwatch::diffusec [mutable], [private]

9.186.4.3 bool Stopwatch::stopped_ [private]

9.186.4.4 struct timeval tv1 Stopwatch::tv2 [private]

9.186.4.5 struct timezone Stopwatch::tz [private]

The documentation for this class was generated from the following files:

- core/Stopwatch.hh
- core/Stopwatch.cc

9.187 cadabra::str_node Class Reference

9.187.1 Description

Elementary building block for a mathematical expression.

Contains information about the way in which the node is related to the parent node, and iterators into the global list of names and rationals.

`#include <Storage.hh>`

Classes

- struct flag_t

Public Types

- enum bracket_t {
  b_round =0, b_square =1, b_curly =2, b_pointy =3,
  b_none =4, b_no =5, b_invalid =6 }
- enum parent_rel_t {
  p_sub =0, p_super =1, p_none =2, p_property =3,
  p_exponent =4, p_components =5 }

  Child nodes are related to their parent node by a so-called parent relation, which can be one of these values.

Public Member Functions

- str_node (void)
- str_node (nset_t::iterator name, bracket_t btype=b_none, parent_rel_t ptype=p_none)
- str_node (const std::string &name, bracket_t btype=b_none, parent_rel_t ptype=p_none)
- bool operator== (const str_node &) const
- bool operator< (const str_node &) const
- void flip_parent_rel ()

  Change the parent relation from sub to super and vice versa (throws error when this is not an index).
- bool is_zero () const
- bool is_identity () const
- bool is_rational () const
- bool is_unsimplified_rational () const
- bool is_integer () const
- bool is_unsimplified_integer () const
- bool is_index () const
- bool is_quoted_string () const
- bool is_command () const
- bool is_inert_command () const
- bool is_name_wildcard () const
- bool is_object_wildcard () const
- bool is_range_wildcard () const
- bool is_siblings_wildcard () const
- bool is_autodeclare_wildcard () const
- bool is_indexstar_wildcard () const
- bool is_indexplus_wildcard () const
- bool is_numbered_symbol () const
- nset_t::iterator name_only ()

Static Public Member Functions

- static bool compare_names_only (const str_node &, const str_node &)
- static bool compare_name_brack_par (const str_node &, const str_node &)
- static bool compare_name_inverse_par (const str_node &, const str_node &)

Public Attributes

- nset_t::iterator name
- rset_t::iterator multiplier
- flag_t fl

9.187.2 Member Enumeration Documentation

9.187.2.1 enum cadabra::str_node::bracket_t

Enumerator

b_round

b_square

b_curly

b_pointy

b_none

b_no

b_invalid

9.187.2.2 enum cadabra::str_node::parent_rel_t

Child nodes are related to their parent node by a so-called parent relation, which can be one of these values.

Enumerator

p_sub

p_super

p_none

p_property

p_exponent

p_components

9.187.3 Constructor & Destructor Documentation

9.187.3.1 cadabra::str_node::str_node (void)

9.187.3.2 cadabra::str_node::str_node (nset_t::iterator *name,* bracket_t *btype* = b_none, parent_rel_t *ptype* = p_none)

9.187.3.3 cadabra::str_node::str_node (const std::string & *name,* bracket_t *btype* = b_none, parent_rel_t *ptype* = p_none)

9.187.4 Member Function Documentation

9.187.4.1 bool cadabra::str_node::compare_name_brack_par (const str_node & *one,* const str_node & *two*) [static]

9.187.4.2 bool cadabra::str_node::compare_name_inverse_par (const str_node & *one,* const str_node & *two*) [static]

9.187.4.3 bool cadabra::str_node::compare_names_only (const str_node & *one,* const str_node & *two*) [static]

9.187.4.4 void cadabra::str_node::flip_parent_rel ( )

Change the parent relation from sub to super and vice versa (throws error when this is not an index).

9.187.4.5 bool cadabra::str_node::is_autodeclare_wildcard ( ) const

9.187.4.6 bool cadabra::str_node::is_command ( ) const

9.187.4.7 bool cadabra::str_node::is_identity ( ) const

9.187.4.8 bool cadabra::str_node::is_index ( ) const

9.187.4.9 bool cadabra::str_node::is_indexplus_wildcard ( ) const

9.187.4.10 bool cadabra::str_node::is_indexstar_wildcard ( ) const

9.187.4.11 bool cadabra::str_node::is_inert_command ( ) const

9.187.4.12 bool cadabra::str_node::is_integer ( ) const

9.187.4.13 bool cadabra::str_node::is_name_wildcard ( ) const

9.187.4.14 bool cadabra::str_node::is_numbered_symbol ( ) const

9.187.4.15 bool cadabra::str_node::is_object_wildcard ( ) const

9.187.4.16 bool cadabra::str_node::is_quoted_string ( ) const

9.187.4.17 bool cadabra::str_node::is_range_wildcard ( ) const

9.187.4.18 bool cadabra::str_node::is_rational ( ) const

9.187.4.19 bool cadabra::str_node::is_siblings_wildcard ( ) const

9.187.4.20 bool cadabra::str_node::is_unsimplified_integer ( ) const

9.187.4.21 bool cadabra::str_node::is_unsimplified_rational ( ) const

9.187.4.22 bool cadabra::str_node::is_zero ( ) const

9.187.4.23 nset_t::iterator cadabra::str_node::name_only ( )

9.187.4.24 bool cadabra::str_node::operator< (const **str_node** & *other*) const

9.187.4.25 bool cadabra::str_node::operator== (const **str_node** & *other*) const

9.187.5 Member Data Documentation

9.187.5.1 flag_t cadabra::str_node::fl

9.187.5.2 rset_t::iterator cadabra::str_node::multiplier

9.187.5.3 nset_t::iterator cadabra::str_node::name

The documentation for this class was generated from the following files:

- core/Storage.hh
- core/Storage.cc

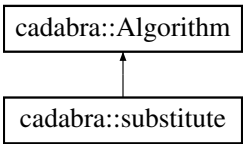
9.188 cadabra::substitute Class Reference

9.188.1 Description

Generic substitution algorithm.

`#include <substitute.hh>`

Inheritance diagram for cadabra::substitute:



Public Member Functions

- substitute (const Kernel &, Ex &tr, Ex &args)
- virtual bool can_apply (iterator st)

  Match the lhs of the replacement rule to the subtree 'st' under consideration.
- virtual result_t apply (iterator &)

Public Attributes

- Ex_comparator comparator

Private Attributes

- Ex & args
- iterator use_rule
- iterator conditions
- std::map< iterator, bool > lhs_contains_dummies
- std::map< iterator, bool > rhs_contains_dummies
- sort_product sort_product_

Additional Inherited Members

9.188.2 Constructor & Destructor Documentation

9.188.2.1 substitute::substitute (const Kernel & *k*, Ex & *tr*, Ex & *args*)

9.188.3 Member Function Documentation

9.188.3.1 Algorithm::result_t substitute::apply (iterator & *st*) [virtual]

Implements cadabra::Algorithm.

9.188.3.2 bool substitute::can_apply (iterator *st*) [virtual]

Match the lhs of the replacement rule to the subtree 'st' under consideration.

This will fill the replacement_map giving a map from objects appearing in the *rule* to what they matched to.

For F_{m n} -> G_{m n}, position-free indices and the expression F^{m n}, we would get a replacement map_m -> ^m _n -> ^n (Note how it is important that despite these indices being position-free, the rules in the replacement map do not ignore sub/superscript information). In the apply member, we then replace all of 'st' with the rhs of the rule, and scan through the rule (with a basic replacement_map.find based on simple tree comparison logic, not pattern matching) for replacements to be made.

Implements cadabra::Algorithm.

9.188.4 Member Data Documentation

9.188.4.1 Ex& cadabra::substitute::args [private]

9.188.4.2 Ex_comparator cadabra::substitute::comparator

9.188.4.3 iterator cadabra::substitute::conditions [private]

9.188.4.4 std::map<iterator, bool> cadabra::substitute::lhs_contains_dummies [private]

9.188.4.5 std::map<iterator, bool> cadabra::substitute::rhs_contains_dummies [private]

9.188.4.6 sort_product cadabra::substitute::sort_product_ [private]

9.188.4.7 iterator cadabra::substitute::use_rule [private]

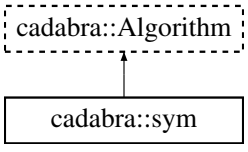
The documentation for this class was generated from the following files:

- core/algorithms/substitute.hh
- core/algorithms/substitute.cc

9.189 cadabra::sym Class Reference

```
#include <sym.hh>
```

Inheritance diagram for cadabra::sym:



Public Member Functions

- sym (const Kernel &, Ex &, Ex &, bool sign)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &it) override

Protected Member Functions

- result_t doit (iterator &, bool)

Protected Attributes

- Ex objects
- bool sign
- std::vector< unsigned int > argloc_2_treeloc
- combin::combinations< unsigned int > raw_ints

Additional Inherited Members

9.189.1 Constructor & Destructor Documentation

9.189.1.1 sym::sym (const Kernel & *k,* Ex & *tr,* Ex & *objs,* bool *sign*)

9.189.2 Member Function Documentation

9.189.2.1 Algorithm::result_t sym::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.189.2.2 bool sym::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.189.2.3 Algorithm::result_t sym::doit (iterator & *it,* bool *sign*) [protected]

9.189.3 Member Data Documentation

9.189.3.1 std::vector<unsigned int> cadabra::sym::argloc_2_treeloc [protected]

9.189.3.2 Ex cadabra::sym::objects [protected]

9.189.3.3 combin::combinations<unsigned int> cadabra::sym::raw_ints [protected]

9.189.3.4 bool cadabra::sym::sign [protected]

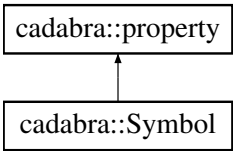
The documentation for this class was generated from the following files:

- core/algorithms/sym.hh
- core/algorithms/sym.cc

9.190 cadabra::Symbol Class Reference

`#include <Symbol.hh>`

Inheritance diagram for cadabra::Symbol:



Public Member Functions

- virtual std::string name () const

Static Public Member Functions

- static const Symbol * get (const Properties &, Ex::iterator, bool ignore_parent_rel=false)

Additional Inherited Members

9.190.1 Member Function Documentation

9.190.1.1 const Symbol * Symbol::get (const Properties & *properties*, Ex::iterator *it*, bool *ignore_parent_rel* = `false`) [static]

9.190.1.2 std::string Symbol::name ( ) const [virtual]

Implements cadabra::property.

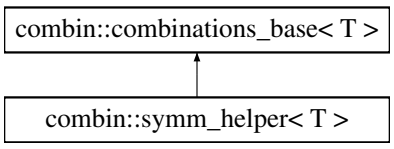
The documentation for this class was generated from the following files:

- core/properties/Symbol.hh
- core/properties/Symbol.cc

9.191 combin::symm_helper< T > Class Template Reference

`#include <Combinatorics.hh>`

Inheritance diagram for combin::symm_helper< T >:



Public Member Functions

- symm_helper (symmetriser< T > &)
- virtual void clear ()

Public Attributes

- int current_multiplicity

Protected Member Functions

- virtual void vector_generated (const std::vector< unsigned int > &)

Protected Attributes

- bool first_one
- symmetriser< T > & owner_

Additional Inherited Members

9.191.1 Constructor & Destructor Documentation

9.191.1.1 template<class T> combin::symm_helper< T >::symm_helper (symmetriser< T > & *tt*)

9.191.2 Member Function Documentation

9.191.2.1 template<class T > void combin::symm_helper< T >::clear ( ) [virtual]

Reimplemented from combin::combinations_base< T >.

9.191.2.2 template<class T > void combin::symm_helper< T >::vector_generated (const std::vector< unsigned int > & *vec*) [protected], [virtual]

Implements combin::combinations_base< T >.

9.191.3 Member Data Documentation

9.191.3.1 template<class T> int combin::symm_helper< T >::current_multiplicity

9.191.3.2 template<class T> bool combin::symm_helper< T >::first_one [protected]

9.191.3.3 template<class T> symmetriser<T>& combin::symm_helper< T >::owner_ [protected]

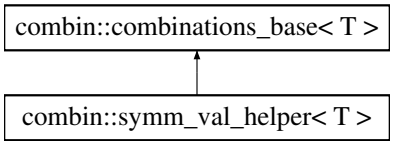
The documentation for this class was generated from the following file:

- core/Combinatorics.hh

9.192 combin::symm_val_helper< T > Class Template Reference

`#include <Combinatorics.hh>`

Inheritance diagram for combin::symm_val_helper< T >:



Public Member Functions

- symm_val_helper (symmetriser< T > &)
- virtual void clear ()

Public Attributes

- int current_multiplicity

Protected Member Functions

- virtual void vector_generated (const std::vector< unsigned int > &)

Protected Attributes

- bool first_one
- symmetriser< T > & owner_

Additional Inherited Members

9.192.1 Constructor & Destructor Documentation

9.192.1.1 template<class T> combin::symm_val_helper< T >::symm_val_helper (symmetriser< T > & *tt*)

9.192.2 Member Function Documentation

9.192.2.1 template<class T > void combin::symm_val_helper< T >::clear ( ) [virtual]

Reimplemented from combin::combinations_base< T >.

9.192.2.2 template<class T > void combin::symm_val_helper< T >::vector_generated (const std::vector< unsigned int > & *vec*) [protected], [virtual]

Implements combin::combinations_base< T >.

9.192.3 Member Data Documentation

9.192.3.1 template<class T> int combin::symm_val_helper< T >::current_multiplicity

9.192.3.2 template<class T> bool combin::symm_val_helper< T >::first_one [protected]

9.192.3.3 template<class T> symmetriser<T>& combin::symm_val_helper< T >::owner_ [protected]

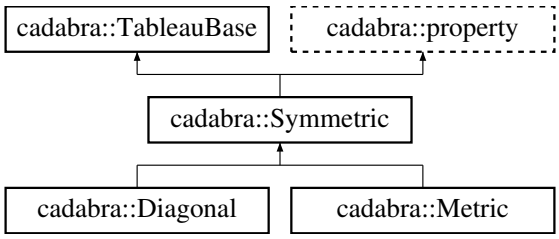
The documentation for this class was generated from the following file:

- core/Combinatorics.hh

9.193 cadabra::Symmetric Class Reference

`#include <Symmetric.hh>`

Inheritance diagram for cadabra::Symmetric:



Public Member Functions

- virtual std::string name () const override
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const override
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const override

Additional Inherited Members

9.193.1 Member Function Documentation

9.193.1.1 TableauBase::tab_t Symmetric::get_tab (const Properties & *pr,* Ex & *tr,* Ex::iterator *it,* unsigned int *num*) const [override], [virtual]

Implements cadabra::TableauBase.

9.193.1.2 std::string Symmetric::name ( ) const [override], [virtual]

Implements cadabra::property.

Reimplemented in cadabra::Metric, and cadabra::Diagonal.

9.193.1.3 unsigned int Symmetric::size (const Properties & , Ex & , Ex::iterator) const [override], [virtual]

Implements cadabra::TableauBase.

The documentation for this class was generated from the following files:

- core/properties/Symmetric.hh
- core/properties/Symmetric.cc

9.194 combin::symmetriser< T > Class Template Reference

`#include <Combinatorics.hh>`

Public Member Functions

- symmetriser ()
- void apply_symmetry (long start=-1, long end=-1)
- range_t values_to_locations (const std::vector< T > &values) const

  Convert vectors of values to vectors of locations in the original (mainly useful to create input_asym for permutation by value).
- const std::vector< T > & operator[ ] (unsigned int) const
- int signature (unsigned int) const
- void set_multiplicity (unsigned int pos, int val)
- unsigned int size () const
- void clear ()
- void collect ()

  Collect equal entries, and adjust the multiplier field accordingly.
- void remove_multiplicity_zero ()

Public Attributes

- std::vector< T > original
- unsigned int block_length
- std::vector< unsigned int > permute_blocks
- std::vector< T > value_permute
- int permutation_sign
- std::vector< unsigned int > sublengths
- range_vector_t input_asym
- range_vector_t sublengths_scattered

Private Attributes

- symm_helper< T > sh_
- symm_val_helper< T > svh_
- unsigned int current_
- std::vector< std::vector< T > > originals
- std::vector< int > multiplicity

Friends

- class symm_helper< T >
- class symm_val_helper< T >

9.194.1 Constructor & Destructor Documentation

9.194.1.1 template<class T > combin::symmetriser< T >::symmetriser ( )

9.194.2 Member Function Documentation

9.194.2.1 template<class T > void combin::symmetriser< T >::apply_symmetry (long *start* = −1, long *end* = −1)

9.194.2.2 template<class T > void combin::symmetriser< T >::clear ( )

9.194.2.3 template<class T > void combin::symmetriser< T >::collect ( )

Collect equal entries, and adjust the multiplier field accordingly.

9.194.2.4 template<class T > const std::vector< T > & combin::symmetriser< T >::operator[ ] (unsigned int *i*) const

9.194.2.5 template<class T > void combin::symmetriser< T >::remove_multiplicity_zero ( )

9.194.2.6 template<class T > void combin::symmetriser< T >::set_multiplicity (unsigned int *pos*, int *val*)

9.194.2.7 template<class T > int combin::symmetriser< T >::signature (unsigned int *i*) const

9.194.2.8 template<class T > unsigned int combin::symmetriser< T >::size ( ) const

9.194.2.9 template<class T> range_t combin::symmetriser< T >::values_to_locations (const std::vector< T > & *values*) const

Convert vectors of values to vectors of locations in the original (mainly useful to create input_asym for permutation by value).

9.194.3 Friends And Related Function Documentation

9.194.3.1 template<class T> friend class symm_helper< T > [friend]

9.194.3.2 template<class T> friend class symm_val_helper< T > [friend]

9.194.4 Member Data Documentation

9.194.4.1 template<class T> unsigned int combin::symmetriser< T >::block_length

9.194.4.2 template<class T> unsigned int combin::symmetriser< T >::current_ [private]

9.194.4.3 template<class T> range_vector_t combin::symmetriser< T >::input_asym

9.194.4.4 template<class T> std::vector<int> combin::symmetriser< T >::multiplicity [private]

9.194.4.5 template<class T> std::vector<T> combin::symmetriser< T >::original

9.194.4.6 template<class T> std::vector<std::vector<T> > combin::symmetriser< T >::originals [private]

9.194.4.7 template<class T> int combin::symmetriser< T >::permutation_sign

9.194.4.8 template<class T> std::vector<unsigned int> combin::symmetriser< T >::permute_blocks

9.194.4.9 template<class T> symm_helper<T> combin::symmetriser< T >::sh_ [private]

9.194.4.10 template<class T> std::vector<unsigned int> combin::symmetriser< T >::sublengths

9.194.4.11 template<class T> range_vector_t combin::symmetriser< T >::sublengths_scattered

9.194.4.12 template<class T> symm_val_helper<T> combin::symmetriser< T >::svh_ [private]

9.194.4.13 template<class T> std::vector<T> combin::symmetriser< T >::value_permute

The documentation for this class was generated from the following file:

- core/Combinatorics.hh

9.195 cadabra2_defaults.Sympy Class Reference

Static Private Attributes

- string __version__ = "unavailable"

9.195.1 Member Data Documentation

9.195.1.1 string cadabra2_defaults.Sympy.__version__ = "unavailable" [static], [private]

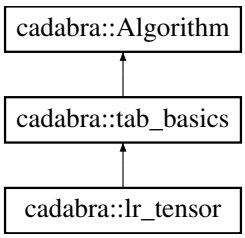
The documentation for this class was generated from the following file:

- core/cadabra2_defaults.py

9.196 cadabra::tab_basics Class Reference

`#include <tab_basics.hh>`

Inheritance diagram for cadabra::tab_basics:



Public Types

- typedef yngtab::filled_tableau< unsigned int > uinttab_t
- typedef yngtab::tableaux< uinttab_t > uinttabs_t

Public Member Functions

- tab_basics (const Kernel &, Ex &)
- void tree_to_numerical_tab (iterator, uinttab_t &)

  Convert an Ex to a numerical Young tableau, using num_to_it below.
- unsigned int find_obj (const Ex &other)
- void tabs_to_tree (uinttabs_t &, iterator, iterator, bool even_only)

  The inverse, converting tableaux to Ex objects attached as children of the iterator.
- void tabs_to_singlet_rules (uinttabs_t &, iterator)

Public Attributes

- std::vector< Ex::iterator > num_to_it

Additional Inherited Members

9.196.1 Member Typedef Documentation

9.196.1.1 typedef yngtab::filled_tableau<unsigned int> cadabra::tab_basics::uinttab_t

9.196.1.2 typedef yngtab::tableaux<uinttab_t> cadabra::tab_basics::uinttabs_t

9.196.2 Constructor & Destructor Documentation

9.196.2.1 tab_basics::tab_basics (const Kernel & *k,* Ex & *tr*)

9.196.3 Member Function Documentation

9.196.3.1 unsigned int tab_basics::find_obj (const Ex & *other*)

9.196.3.2 void tab_basics::tabs_to_singlet_rules (uinttabs_t & *tabs,* iterator *top*)

9.196.3.3 void tab_basics::tabs_to_tree (uinttabs_t & *tabs,* iterator *top,* iterator *tabpat,* bool *even_only*)

The inverse, converting tableaux to Ex objects attached as children of the iterator.

9.196.3.4 void tab_basics::tree_to_numerical_tab (iterator *tab1,* uinttab_t & *one*)

Convert an Ex to a numerical Young tableau, using num_to_it below.

9.196.4 Member Data Documentation

9.196.4.1 std::vector<Ex::iterator> cadabra::tab_basics::num_to_it

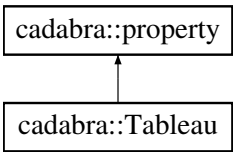
The documentation for this class was generated from the following files:

- core/algorithms/tab_basics.hh
- core/algorithms/tab_basics.cc

9.197 cadabra::Tableau Class Reference

`#include <Tableau.hh>`

Inheritance diagram for cadabra::Tableau:



Public Member Functions

- virtual ~Tableau ()
- virtual std::string name () const
- virtual bool parse (Ex &, Ex::iterator, Ex::iterator, keyval_t &)

Public Attributes

- int dimension

Additional Inherited Members

9.197.1 Constructor & Destructor Documentation

9.197.1.1 virtual cadabra::Tableau::~Tableau ( ) [inline], [virtual]

9.197.2 Member Function Documentation

9.197.2.1 std::string Tableau::name ( ) const [virtual]

Implements cadabra::property.

9.197.2.2 bool Tableau::parse (Ex & *tr,* Ex::iterator *pat,* Ex::iterator *prop,* keyval_t & *keyvals*) [virtual]

9.197.3 Member Data Documentation

9.197.3.1 int cadabra::Tableau::dimension

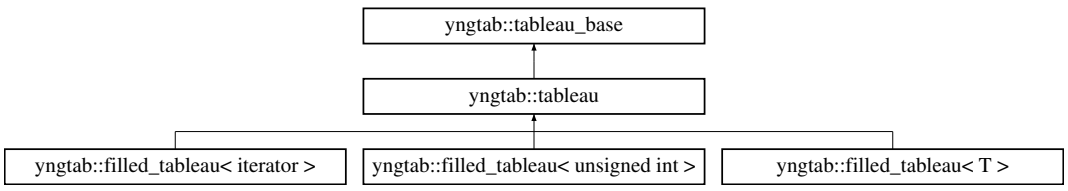
The documentation for this class was generated from the following files:

- core/properties/Tableau.hh
- core/properties/Tableau.cc

9.198 yngtab::tableau Class Reference

```
#include <YoungTab.hh>
```

Inheritance diagram for yngtab::tableau:



Public Member Functions

- virtual ~tableau ()
- virtual unsigned int number_of_rows () const
- virtual unsigned int row_size (unsigned int row) const
- virtual void add_box (unsigned int row)
- virtual void remove_box (unsigned int row)
- virtual void clear ()
- tableau & operator= (const tableau &)

Private Attributes

- std::vector< int > rows

Additional Inherited Members

9.198.1 Constructor & Destructor Documentation

9.198.1.1 yngtab::tableau::~tableau ( ) [virtual]

9.198.2 Member Function Documentation

9.198.2.1 void yngtab::tableau::add_box (unsigned int *row*) [virtual]

Implements yngtab::tableau_base.

Reimplemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, and yngtab::filled_tableau< iterator >.

9.198.2.2 void yngtab::tableau::clear ( ) [virtual]

Implements yngtab::tableau_base.

Reimplemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, and yngtab::filled_tableau< iterator >.

9.198.2.3 unsigned int yngtab::tableau::number_of_rows ( ) const [virtual]

Implements yngtab::tableau_base.

Reimplemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, and yngtab::filled_tableau< iterator >.

9.198.2.4 tableau & yngtab::tableau::operator= (const tableau & *other*)

9.198.2.5 void yngtab::tableau::remove_box (unsigned int *row*) [virtual]

Implements yngtab::tableau_base.

Reimplemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, and yngtab::filled_tableau< iterator >.

9.198.2.6 unsigned int yngtab::tableau::row_size (unsigned int *row*) const [virtual]

Implements yngtab::tableau_base.

Reimplemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, and yngtab::filled_tableau< iterator >.

9.198.3 Member Data Documentation

9.198.3.1 std::vector<int> yngtab::tableau::rows [private]

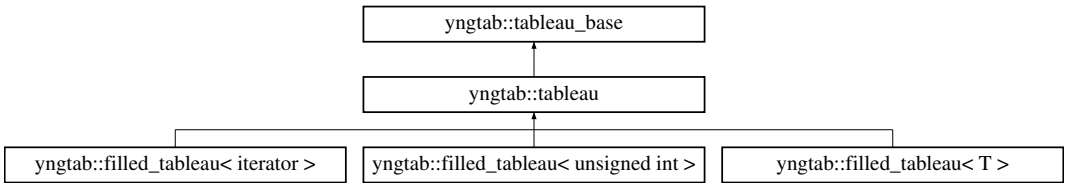
The documentation for this class was generated from the following files:

- core/YoungTab.hh
- core/YoungTab.cc

9.199 yngtab::tableau_base Class Reference

`#include <YoungTab.hh>`

Inheritance diagram for yngtab::tableau_base:



Public Member Functions

- tableau_base ()
- virtual ~tableau_base ()
- virtual unsigned int number_of_rows () const =0
- virtual unsigned int row_size (unsigned int row) const =0
- virtual unsigned int column_size (unsigned int col) const
- virtual void add_box (unsigned int row)=0
- virtual void remove_box (unsigned int row)=0
- virtual void add_row (unsigned int row_size)
- virtual void clear ()=0
- yngint_t dimension (unsigned int) const
- unsigned long hook_length (unsigned int row, unsigned int col) const
- yngint_t hook_length_prod () const

Public Attributes

- yngrat_t multiplicity
- int selfdual_column

9.199.1 Constructor & Destructor Documentation

9.199.1.1 yngtab::tableau_base::tableau_base ( )

9.199.1.2 yngtab::tableau_base::~tableau_base ( ) [virtual]

9.199.2 Member Function Documentation

9.199.2.1 virtual void yngtab::tableau_base::add_box (unsigned int *row*) [pure virtual]

Implemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, yngtab::filled_tableau< iterator >, and yngtab::tableau.

9.199.2.2 void yngtab::tableau_base::add_row (unsigned int *row_size*) [virtual]

9.199.2.3 virtual void yngtab::tableau_base::clear ( ) [pure virtual]

Implemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, yngtab::filled_tableau< iterator >, and yngtab::tableau.

9.199.2.4 unsigned int yngtab::tableau_base::column_size (unsigned int *col*) const [virtual]

9.199.2.5 yngint_t yngtab::tableau_base::dimension (unsigned int *dim*) const

9.199.2.6 unsigned long yngtab::tableau_base::hook_length (unsigned int *row,* unsigned int *col*) const

9.199.2.7 yngint_t yngtab::tableau_base::hook_length_prod ( ) const

9.199.2.8 virtual unsigned int yngtab::tableau_base::number_of_rows ( ) const [pure virtual]

Implemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, yngtab::filled_tableau< iterator >, and yngtab::tableau.

9.199.2.9 virtual void yngtab::tableau_base::remove_box (unsigned int *row*) [pure virtual]

Implemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, yngtab::filled_tableau< iterator >, and yngtab::tableau.

9.199.2.10 virtual unsigned int yngtab::tableau_base::row_size (unsigned int *row*) const [pure virtual]

Implemented in yngtab::filled_tableau< T >, yngtab::filled_tableau< unsigned int >, yngtab::filled_tableau< iterator >, and yngtab::tableau.

9.199.3 Member Data Documentation

9.199.3.1 yngrat_t yngtab::tableau_base::multiplicity

9.199.3.2 int yngtab::tableau_base::selfdual_column

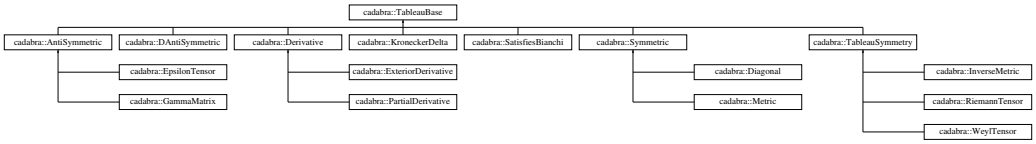
The documentation for this class was generated from the following files:

- core/YoungTab.hh
- core/YoungTab.cc

9.200 cadabra::TableauBase Class Reference

`#include <TableauBase.hh>`

Inheritance diagram for cadabra::TableauBase:



Public Types

- typedef yngtab::filled_tableau< unsigned int > tab_t

Public Member Functions

- virtual ~TableauBase ()
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const =0
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const =0
- virtual bool only_column_exchange () const
- int get_indexgroup (const Properties &, Ex &, Ex::iterator, int) const
- bool is_simple_symmetry (const Properties &, Ex &, Ex::iterator) const

9.200.1 Member Typedef Documentation

9.200.1.1 typedef yngtab::filled_tableau<unsigned int> cadabra::TableauBase::tab_t

9.200.2 Constructor & Destructor Documentation

9.200.2.1 virtual cadabra::TableauBase::~TableauBase ( ) [inline], [virtual]

9.200.3 Member Function Documentation

9.200.3.1 int TableauBase::get_indexgroup (const Properties & *pr,* Ex & *tr,* Ex::iterator *it,* int *indexnum*) const

9.200.3.2 virtual tab_t cadabra::TableauBase::get_tab (const Properties & , Ex & , Ex::iterator , unsigned *int*) const [pure virtual]

Implemented in cadabra::Derivative, cadabra::TableauSymmetry, cadabra::AntiSymmetric, cadabra::DAnti↩Symmetric, cadabra::KroneckerDelta, cadabra::PartialDerivative, cadabra::SatisfiesBianchi, and cadabra::↩Symmetric.

9.200.3.3 bool TableauBase::is_simple_symmetry (const Properties & *pr,* Ex & *tr,* Ex::iterator *it*) const

9.200.3.4 virtual bool cadabra::TableauBase::only_column_exchange ( ) const [inline], [virtual]

Reimplemented in cadabra::TableauSymmetry.

9.200.3.5 virtual unsigned int cadabra::TableauBase::size (const Properties & , Ex & , Ex::iterator) const [pure virtual]

Implemented in cadabra::Derivative, cadabra::TableauSymmetry, cadabra::AntiSymmetric, cadabra::DAnti↩Symmetric, cadabra::KroneckerDelta, cadabra::PartialDerivative, cadabra::SatisfiesBianchi, and cadabra::↩Symmetric.

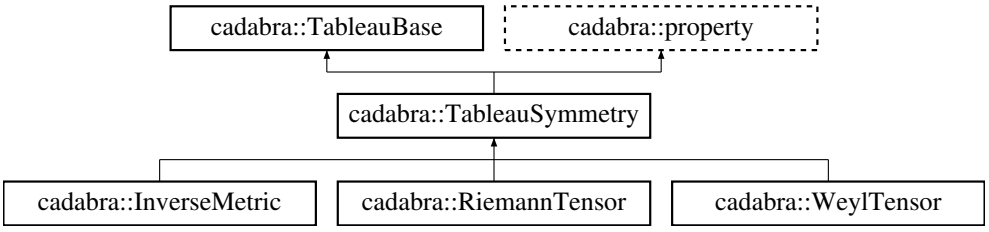
The documentation for this class was generated from the following files:

- core/properties/TableauBase.hh
- core/properties/TableauBase.cc

9.201 cadabra::TableauSymmetry Class Reference

`#include <TableauSymmetry.hh>`

Inheritance diagram for cadabra::TableauSymmetry:



Public Member Functions

- virtual ~TableauSymmetry ()
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual std::string name () const override
- virtual void latex (std::ostream &) const override

  Display the property on the stream.
- virtual unsigned int size (const Properties &, Ex &, Ex::iterator) const override
- virtual tab_t get_tab (const Properties &, Ex &, Ex::iterator, unsigned int) const override
- virtual bool only_column_exchange () const override

Public Attributes

- std::vector< tab_t > tabs

Private Attributes

- bool only_col_

Additional Inherited Members

9.201.1 Constructor & Destructor Documentation

9.201.1.1 TableauSymmetry::~TableauSymmetry ( ) [virtual]

9.201.2 Member Function Documentation

9.201.2.1 TableauBase::tab_t TableauSymmetry::get_tab (const Properties & , Ex & , Ex::iterator , unsigned int *num*) const [override], [virtual]

Implements cadabra::TableauBase.

9.201.2.2 void TableauSymmetry::latex (std::ostream & *str*) const [override], [virtual]

Display the property on the stream.

Generate a LaTeX representation of the property, assuming LaTeX is in text mode (so it needs dollar symbols to switch to maths).

Reimplemented from cadabra::property.

Reimplemented in cadabra::WeylTensor.

9.201.2.3 std::string TableauSymmetry::name ( ) const [override], [virtual]

Implements cadabra::property.

Reimplemented in cadabra::InverseMetric, cadabra::WeylTensor, and cadabra::RiemannTensor.

9.201.2.4 bool TableauSymmetry::only_column_exchange ( ) const [override], [virtual]

Reimplemented from cadabra::TableauBase.

9.201.2.5 bool TableauSymmetry::parse (const Kernel & *kernel*, keyval_t & *keyvals*) [override], [virtual]

Reimplemented from cadabra::property.

Reimplemented in cadabra::InverseMetric.

9.201.2.6 unsigned int TableauSymmetry::size (const Properties & , Ex & , Ex::iterator) const [override], [virtual]

Implements cadabra::TableauBase.

9.201.3 Member Data Documentation

9.201.3.1 bool cadabra::TableauSymmetry::only_col_ [private]

9.201.3.2 std::vector<tab_t> cadabra::TableauSymmetry::tabs

The documentation for this class was generated from the following files:

- core/properties/TableauSymmetry.hh
- core/properties/TableauSymmetry.cc

9.202 yngtab::tableaux< T > Class Template Reference

`#include <YoungTab.hh>`

Public Types

- typedef std::list< T > tableau_container_t
- typedef std::back_insert_iterator< tableau_container_t > back_insert_iterator

Public Member Functions

- yngint_t total_dimension (unsigned int dim)
- void remove_nullifying_traces ()
- bool standard_form ()

  Put the set of tableaux into standard form by using Garnir symmetries.
- void add_tableau (const T &)
- void symmetrise (const T &tabsym)
- back_insert_iterator get_back_insert_iterator ()

Public Attributes

- tableau_container_t storage

9.202.1 Member Typedef Documentation

9.202.1.1 template<class T> typedef std::back_insert_iterator<tableau_container_t> yngtab::tableaux< T >::back_insert_iterator

9.202.1.2 template<class T> typedef std::list<T> yngtab::tableaux< T >::tableau_container_t

9.202.2 Member Function Documentation

9.202.2.1 template<class T> void yngtab::tableaux< T >::add_tableau (const T & *ntab*)

9.202.2.2 template<class T > tableaux< T >::back_insert_iterator yngtab::tableaux< T >::get_back_insert_iterator ( )

9.202.2.3 template<class T > void yngtab::tableaux< T >::remove_nullifying_traces ( )

9.202.2.4 template<class T > bool yngtab::tableaux< T >::standard_form ( )

Put the set of tableaux into standard form by using Garnir symmetries.

Return value indicates whether the tableaux were already all in standard form.

9.202.2.5 template<class T> void yngtab::tableaux< T >::symmetrise (const T & *tabsym*)

9.202.2.6 template<class T > yngint_t yngtab::tableaux< T >::total_dimension (unsigned int *dim*)

9.202.3 Member Data Documentation

9.202.3.1 template<class T> tableau_container_t yngtab::tableaux< T >::storage

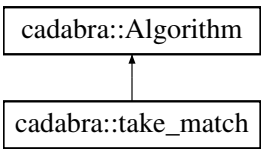
The documentation for this class was generated from the following file:

- core/YoungTab.hh

9.203 cadabra::take_match Class Reference

`#include <take_match.hh>`

Inheritance diagram for cadabra::take_match:



Public Member Functions

- take_match (const Kernel &k, Ex &e, Ex &r)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Private Attributes

- Ex & rules

Additional Inherited Members

9.203.1 Constructor & Destructor Documentation

9.203.1.1 take_match::take_match (const Kernel & *k,* Ex & *e,* Ex & *r*)

9.203.2 Member Function Documentation

9.203.2.1 Algorithm::result_t take_match::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.203.2.2 bool take_match::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.203.3 Member Data Documentation

9.203.3.1 Ex& cadabra::take_match::rules [private]

The documentation for this class was generated from the following files:

- core/algorithms/take_match.hh
- core/algorithms/take_match.cc

9.204 cadabra::exchange::tensor_type_t Struct Reference

`#include <Exchange.hh>`

Public Attributes

- nset_t::iterator name
- unsigned int number_of_indices

9.204.1 Member Data Documentation

9.204.1.1 nset_t::iterator cadabra::exchange::tensor_type_t::name

9.204.1.2 unsigned int cadabra::exchange::tensor_type_t::number_of_indices

The documentation for this struct was generated from the following file:

- core/Exchange.hh

9.205 cadabra::TerminalStream Class Reference

`#include <TerminalStream.hh>`

Public Member Functions

- TerminalStream (const Kernel &, std::ostream &)
- TerminalStream & operator<< (const Ex &)
- TerminalStream & operator<< (std::shared_ptr< Ex >)
- template<class T >
  TerminalStream & operator<< (const T &obj)
- TerminalStream & operator<< (std::ostream &(*os)(std::ostream &))

Private Attributes

- const Kernel & kernel
- std::ostream & out_

9.205.1 Constructor & Destructor Documentation

9.205.1.1 TerminalStream::TerminalStream (const Kernel & *k*, std::ostream & *s*)

9.205.2 Member Function Documentation

9.205.2.1 TerminalStream & TerminalStream::operator<< (const Ex & *ex*)

9.205.2.2 TerminalStream & TerminalStream::operator<< (std::shared_ptr< Ex > *ex*)

9.205.2.3 template<class T > TerminalStream& cadabra::TerminalStream::operator<< (const T & *obj*) [inline]

9.205.2.4 TerminalStream& cadabra::TerminalStream::operator<< (std::ostream &(*)(std::ostream &) *os*) [inline]

9.205.3 Member Data Documentation

9.205.3.1 const Kernel& cadabra::TerminalStream::kernel [private]

9.205.3.2 std::ostream& cadabra::TerminalStream::out_ [private]

The documentation for this class was generated from the following files:

- core/TerminalStream.hh
- core/TerminalStream.cc

9.206 cadabra::TeXEngine Class Reference

9.206.1 Description

TeXEngine is used to convert LaTeX strings into PNG images.

This is a two-stage process: you first 'check in' a string into the system, in exchange for a pointer to a TeXRequest object. When you are ready to retrieve the image, call 'get_pixbuf'.

If you need to generate images for more than one string, simply check them all in and then call 'convert_all' before retrieving the pixbufs. This requires only one round-trip through latex/dvipng.

```
#include <TeXEngine.hh>
```

Classes

- class TeXException
- class TeXRequest

Public Member Functions

- TeXEngine ()
- ~TeXEngine ()
- void set_geometry (int horizontal_pixels)
- void set_scale (double)
- void set_font_size (int font_size)
- std::shared_ptr< TeXRequest > checkin (const std::string &, const std::string &startwrap, const std::string &endwrap)
- std::shared_ptr< TeXRequest > modify (std::shared_ptr< TeXRequest >, const std::string &)
- void convert_all ()

  Generate images for all TeXRequests which are labelled as needing conversion.
- void invalidate_all ()

  Mark all TeXRequests as needing re-generating.
- void checkout (std::shared_ptr< TeXRequest >)

  Mark a TeXRequest as no longer being needed.
- void checkout_all ()

Public Attributes

- std::vector< std::string > latex_packages

Private Member Functions

- void erase_file (const std::string &) const
- void convert_one (std::shared_ptr< TeXRequest >)
- void convert_set (std::set< std::shared_ptr< TeXRequest > > &)
- std::string handle_latex_errors (const std::string &, int exit_code) const

Private Attributes

- std::set< std::shared_ptr< TeXRequest > > requests
- std::string preamble_string
- int horizontal_pixels_
- int font_size_
- double scale_

Static Private Attributes

- static double millimeter_per_inch = 25.4

9.206.2 Constructor & Destructor Documentation

9.206.2.1 TeXEngine::TeXEngine ( )

9.206.2.2 TeXEngine::~TeXEngine ( )

9.206.3 Member Function Documentation

9.206.3.1 std::shared_ptr< TeXEngine::TeXRequest > TeXEngine::checkin (const std::string & *txt,* const std::string & *startwrap,* const std::string & *endwrap*)

9.206.3.2 void TeXEngine::checkout (std::shared_ptr< TeXRequest > *req*)

Mark a TeXRequest as no longer being needed.

9.206.3.3 void TeXEngine::checkout_all ( )

9.206.3.4 void TeXEngine::convert_all ( )

Generate images for all TeXRequests which are labelled as needing conversion.

9.206.3.5 void TeXEngine::convert_one (std::shared_ptr< TeXRequest > *req*) [private]

9.206.3.6 void TeXEngine::convert_set (std::set< std::shared_ptr< TeXRequest > > & *reqs*) [private]

9.206.3.7 void TeXEngine::erase_file (const std::string & *nm*) const [private]

9.206.3.8 std::string TeXEngine::handle_latex_errors (const std::string & *result,* int *exit_code*) const [private]

9.206.3.9 void TeXEngine::invalidate_all ( )

Mark all TeXRequests as needing re-generating.

Use this e.g. when changing font size for the entire notebook: first invalidate_all, then convert_all.

9.206.3.10 std::shared_ptr< TeXEngine::TeXRequest > TeXEngine::modify (std::shared_ptr< TeXRequest > *req*, const std::string & *txt*)

9.206.3.11 void TeXEngine::set_font_size (int *font_size*)

9.206.3.12 void TeXEngine::set_geometry (int *horizontal_pixels*)

9.206.3.13 void TeXEngine::set_scale (double *scale*)

9.206.4 Member Data Documentation

9.206.4.1 int cadabra::TeXEngine::font_size_ [private]

9.206.4.2 int cadabra::TeXEngine::horizontal_pixels_ [private]

9.206.4.3 std::vector<std::string> cadabra::TeXEngine::latex_packages

9.206.4.4 double TeXEngine::millimeter_per_inch = 25.4 [static], [private]

9.206.4.5 std::string cadabra::TeXEngine::preamble_string [private]

9.206.4.6 std::set<std::shared_ptr<TeXRequest> > cadabra::TeXEngine::requests [private]

9.206.4.7 double cadabra::TeXEngine::scale_ [private]

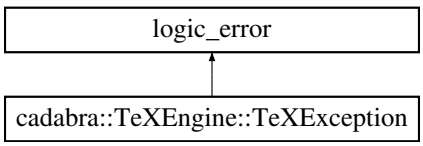
The documentation for this class was generated from the following files:

- frontend/common/TeXEngine.hh
- frontend/common/TeXEngine.cc

9.207 cadabra::TeXEngine::TeXException Class Reference

`#include <TeXEngine.hh>`

Inheritance diagram for cadabra::TeXEngine::TeXException:



Public Member Functions

- TeXException (const std::string &)

9.207.1 Constructor & Destructor Documentation

9.207.1.1 TeXEngine::TeXException::TeXException (const std::string & *str*)

The documentation for this class was generated from the following files:

- frontend/common/TeXEngine.hh
- frontend/common/TeXEngine.cc

9.208 cadabra::TeXEngine::TeXRequest Class Reference

`#include <TeXEngine.hh>`

Public Member Functions

- TeXRequest ()
- unsigned width () const
- unsigned height () const
- const std::vector< unsigned char > & image () const

Private Attributes

- std::string latex_string
- std::string start_wrap
- std::string end_wrap
- bool needs_generating
- std::vector< unsigned char > image_
- unsigned width_
- unsigned height_

Friends

- class TeXEngine

9.208.1 Constructor & Destructor Documentation

9.208.1.1 TeXEngine::TeXRequest::TeXRequest ( )

9.208.2 Member Function Documentation

9.208.2.1 unsigned TeXEngine::TeXRequest::height ( ) const

9.208.2.2 const std::vector< unsigned char > & TeXEngine::TeXRequest::image ( ) const

9.208.2.3 unsigned TeXEngine::TeXRequest::width ( ) const

9.208.3 Friends And Related Function Documentation

9.208.3.1 friend class TeXEngine [friend]

9.208.4 Member Data Documentation

9.208.4.1 std::string cadabra::TeXEngine::TeXRequest::end_wrap [private]

9.208.4.2 unsigned cadabra::TeXEngine::TeXRequest::height_ [private]

9.208.4.3 std::vector<unsigned char> cadabra::TeXEngine::TeXRequest::image_ [private]

9.208.4.4 std::string cadabra::TeXEngine::TeXRequest::latex_string [private]

9.208.4.5 bool cadabra::TeXEngine::TeXRequest::needs_generating [private]

9.208.4.6 std::string cadabra::TeXEngine::TeXRequest::start_wrap [private]

9.208.4.7 unsigned cadabra::TeXEngine::TeXRequest::width_ [private]

The documentation for this class was generated from the following files:

- frontend/common/TeXEngine.hh
- frontend/common/TeXEngine.cc

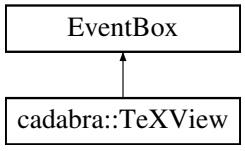
9.209 cadabra::TeXView Class Reference

9.209.1 Description

TeXView is a widget which knows how to turn a string into a LaTeX-rendered image and display that.

```
#include <TeXView.hh>
```

Inheritance diagram for cadabra::TeXView:



Public Member Functions

- TeXView (TeXEngine &, DTree::iterator, int hmargin=25)
- virtual ~TeXView ()
- void update_image ()

  Update the visible image from the pixbuf.
- void dim (bool)

  Dim the output to indicate that the result is no longer guaranteed to be correlated with the input cell from which it was derived.

Public Attributes

- std::shared_ptr< TeXEngine::TeXRequest > content
- sigc::signal1< bool, DTree::iterator > show_hide_requested
- DTree::iterator datacell
- Gtk::VBox vbox
- Gtk::HBox hbox
- Gtk::Image image
- Glib::RefPtr< Gdk::Pixbuf > pixbuf

  The actual image is stored in the image referenced by pixbuf.
- sigc::signal1< bool, std::string > tex_error

Protected Member Functions

- virtual bool on_button_release_event (GdkEventButton *) override
- virtual void on_show () override
- void convert ()

Private Attributes

- TeXEngine & engine

9.209.2 Constructor & Destructor Documentation

9.209.2.1 TeXView::TeXView (TeXEngine & *eng*, DTree::iterator *it*, int *hmargin* = 25)

9.209.2.2 TeXView::~TeXView ( ) [virtual]

9.209.3 Member Function Documentation

9.209.3.1 void TeXView::convert ( ) [protected]

9.209.3.2 void TeXView::dim (bool *d*)

Dim the output to indicate that the result is no longer guaranteed to be correlated with the input cell from which it was derived.

9.209.3.3 bool TeXView::on_button_release_event (GdkEventButton * *ev*) [override], [protected], [virtual]

9.209.3.4 void TeXView::on_show ( ) [override], [protected], [virtual]

9.209.3.5 void TeXView::update_image ( )

Update the visible image from the pixbuf.

Call this in order to propagate changes to the pixbuf (e.g. from re-running the TeXRequest) to the visible widget itself.

9.209.4 Member Data Documentation

9.209.4.1 std::shared_ptr<TeXEngine::TeXRequest> cadabra::TeXView::content

9.209.4.2 DTree::iterator cadabra::TeXView::datacell

9.209.4.3 TeXEngine& cadabra::TeXView::engine [private]

9.209.4.4 Gtk::HBox cadabra::TeXView::hbox

9.209.4.5 Gtk::Image cadabra::TeXView::image

9.209.4.6 Glib::RefPtr<Gdk::Pixbuf> cadabra::TeXView::pixbuf

The actual image is stored in the image referenced by pixbuf.

FIXME: This pointer is not yet shared among instances which show the same content.

9.209.4.7 sigc::signal1<bool, DTree::iterator> cadabra::TeXView::show_hide_requested

9.209.4.8 sigc::signal1<bool, std::string> cadabra::TeXView::tex_error

9.209.4.9 Gtk::VBox cadabra::TeXView::vbox

The documentation for this class was generated from the following files:

- frontend/gtkmm/TeXView.hh
- frontend/gtkmm/TeXView.cc

9.210 ProgressMonitor::Total Class Reference

`#include <ProgressMonitor.hh>`

Public Member Functions

- Total ()
- long time_spent_as_long () const
- bool operator== (const Total &other) const
- std::string str () const

Public Attributes

- std::string name
- size_t call_count
- std::chrono::milliseconds time_spent
- int total_steps

9.210.1 Constructor & Destructor Documentation

9.210.1.1 ProgressMonitor::Total::Total ( )

9.210.2 Member Function Documentation

9.210.2.1 bool ProgressMonitor::Total::operator== (const Total & *other*) const

9.210.2.2 std::string ProgressMonitor::Total::str ( ) const

9.210.2.3 long ProgressMonitor::Total::time_spent_as_long ( ) const

9.210.3 Member Data Documentation

9.210.3.1 size_t ProgressMonitor::Total::call_count

9.210.3.2 std::string ProgressMonitor::Total::name

9.210.3.3 std::chrono::milliseconds ProgressMonitor::Total::time_spent

9.210.3.4 int ProgressMonitor::Total::total_steps

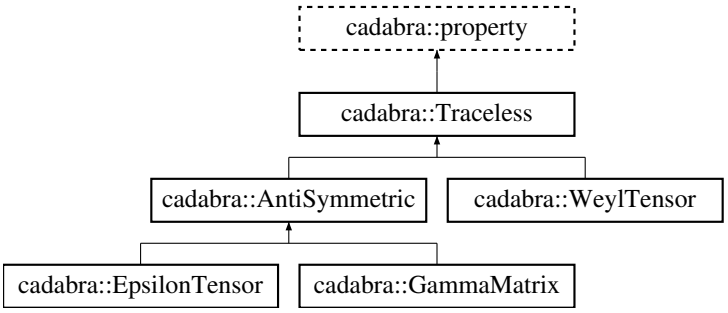
The documentation for this class was generated from the following files:

- core/ProgressMonitor.hh
- core/ProgressMonitor.cc

9.211 cadabra::Traceless Class Reference

`#include <Traceless.hh>`

Inheritance diagram for cadabra::Traceless:



Public Member Functions

- virtual ~Traceless ()
- virtual std::string name () const

Additional Inherited Members

9.211.1 Constructor & Destructor Documentation

9.211.1.1 virtual cadabra::Traceless::~Traceless ( ) [inline], [virtual]

9.211.2 Member Function Documentation

9.211.2.1 std::string Traceless::name ( ) const [virtual]

Implements cadabra::property.

Reimplemented in cadabra::EpsilonTensor, cadabra::AntiSymmetric, cadabra::WeylTensor, and cadabra::GammaMatrix.

The documentation for this class was generated from the following files:

- core/properties/Traceless.hh
- core/properties/Traceless.cc

9.212 cadabra::tree_equal_obj Class Reference

`#include <Compare.hh>`

Public Member Functions

- tree_equal_obj (const Properties ∗)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties ∗ properties

9.212.1 Constructor & Destructor Documentation

9.212.1.1 cadabra::tree_equal_obj::tree_equal_obj (const Properties ∗ *k*)

9.212.2 Member Function Documentation

9.212.2.1 bool cadabra::tree_equal_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.212.3 Member Data Documentation

9.212.3.1 const Properties∗ cadabra::tree_equal_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.213 cadabra::tree_exact_equal_mod_prel_obj Class Reference

`#include <Compare.hh>`

Public Member Functions

- tree_exact_equal_mod_prel_obj (const Properties ∗)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties ∗ properties

9.213.1 Constructor & Destructor Documentation

9.213.1.1 cadabra::tree_exact_equal_mod_prel_obj::tree_exact_equal_mod_prel_obj (const Properties *)

9.213.2 Member Function Documentation

9.213.2.1 bool cadabra::tree_exact_equal_mod_prel_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.213.3 Member Data Documentation

9.213.3.1 const Properties* cadabra::tree_exact_equal_mod_prel_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.214 cadabra::tree_exact_equal_obj Class Reference

`#include <Compare.hh>`

Public Member Functions

- tree_exact_equal_obj (const Properties *)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties * properties

9.214.1 Constructor & Destructor Documentation

9.214.1.1 cadabra::tree_exact_equal_obj::tree_exact_equal_obj (const Properties *)

9.214.2 Member Function Documentation

9.214.2.1 bool cadabra::tree_exact_equal_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.214.3 Member Data Documentation

9.214.3.1 const Properties* cadabra::tree_exact_equal_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.215 cadabra::tree_exact_less_for_indexmap_obj Class Reference

9.215.1 Description

Compare for indexmap_t.

The only comparator object that does not use properties info to lookup properties. This one compares exactly (it cannot do any matching which requires knowledge about index sets because it does not know about properties). It requires parent relations to match including at top level. It requires multipliers to match including at top level. Names with '?' or '??' suffixes are matched literally, not as patterns.

`#include <Compare.hh>`

Public Member Functions

- bool operator() (const Ex &first, const Ex &second) const

9.215.2 Member Function Documentation

9.215.2.1 bool cadabra::tree_exact_less_for_indexmap_obj::operator() (const Ex & *first,* const Ex & *second*) const

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.216 cadabra::tree_exact_less_mod_prel_obj Class Reference

`#include <Compare.hh>`

Public Member Functions

- tree_exact_less_mod_prel_obj (const Properties *)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties * properties

9.216.1 Constructor & Destructor Documentation

9.216.1.1 cadabra::tree_exact_less_mod_prel_obj::tree_exact_less_mod_prel_obj (const Properties * *p*)

9.216.2 Member Function Documentation

9.216.2.1 bool cadabra::tree_exact_less_mod_prel_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.216.3 Member Data Documentation

9.216.3.1 const Properties* cadabra::tree_exact_less_mod_prel_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.217 cadabra::tree_exact_less_no_wildcards_mod_prel_obj Class Reference

`#include <Compare.hh>`

Public Member Functions

- tree_exact_less_no_wildcards_mod_prel_obj (const Properties *)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties * properties

9.217.1 Constructor & Destructor Documentation

9.217.1.1 cadabra::tree_exact_less_no_wildcards_mod_prel_obj::tree_exact_less_no_wildcards_mod_prel_obj (const Properties *)

9.217.2 Member Function Documentation

9.217.2.1 bool cadabra::tree_exact_less_no_wildcards_mod_prel_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.217.3 Member Data Documentation

9.217.3.1 const Properties* cadabra::tree_exact_less_no_wildcards_mod_prel_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.218 cadabra::tree_exact_less_no_wildcards_obj Class Reference

9.218.1 Description

Compare two trees exactly, treat wildcard names as ordinary names.

`#include <Compare.hh>`

Public Member Functions

- tree_exact_less_no_wildcards_obj ()
- tree_exact_less_no_wildcards_obj (const Properties *)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties * properties

9.218.2 Constructor & Destructor Documentation

9.218.2.1 cadabra::tree_exact_less_no_wildcards_obj::tree_exact_less_no_wildcards_obj ( )

9.218.2.2 cadabra::tree_exact_less_no_wildcards_obj::tree_exact_less_no_wildcards_obj (const Properties *)

9.218.3 Member Function Documentation

9.218.3.1 bool cadabra::tree_exact_less_no_wildcards_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.218.4 Member Data Documentation

9.218.4.1 const Properties* cadabra::tree_exact_less_no_wildcards_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.219 cadabra::tree_exact_less_obj Class Reference

9.219.1 Description

Compare two trees exactly, i.e. including exact index names.

`#include <Compare.hh>`

Public Member Functions

- tree_exact_less_obj (const Properties ∗)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties ∗ properties

9.219.2 Constructor & Destructor Documentation

9.219.2.1 cadabra::tree_exact_less_obj::tree_exact_less_obj (const Properties ∗ *p*)

9.219.3 Member Function Documentation

9.219.3.1 bool cadabra::tree_exact_less_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.219.4 Member Data Documentation

9.219.4.1 const Properties∗ cadabra::tree_exact_less_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.220 cadabra::tree_less_modprel_obj Class Reference

`#include <Compare.hh>`

Public Member Functions

- tree_less_modprel_obj (const Properties ∗)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties ∗ properties

9.220.1 Constructor & Destructor Documentation

9.220.1.1 cadabra::tree_less_modprel_obj::tree_less_modprel_obj (const Properties * *k*)

9.220.2 Member Function Documentation

9.220.2.1 bool cadabra::tree_less_modprel_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.220.3 Member Data Documentation

9.220.3.1 const Properties* cadabra::tree_less_modprel_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.221 cadabra::tree_less_obj Class Reference

9.221.1 Description

Compare two trees by pattern logic, i.e. modulo index names.

`#include <Compare.hh>`

Public Member Functions

- tree_less_obj (const Properties *)
- bool operator() (const Ex &first, const Ex &second) const

Private Attributes

- const Properties * properties

9.221.2 Constructor & Destructor Documentation

9.221.2.1 cadabra::tree_less_obj::tree_less_obj (const Properties * *k*)

9.221.3 Member Function Documentation

9.221.3.1 bool cadabra::tree_less_obj::operator() (const Ex & *first,* const Ex & *second*) const

9.221.4 Member Data Documentation

9.221.4.1 const Properties* cadabra::tree_less_obj::properties [private]

The documentation for this class was generated from the following files:

- core/Compare.hh
- core/Compare.cc

9.222 UI Class Reference

Public Member Functions

- void run ()

9.222.1 Member Function Documentation

9.222.1.1 void UI::run ( )

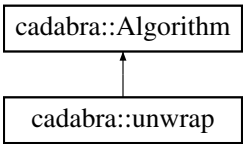
The documentation for this class was generated from the following file:

- client_server/test_client.cc

9.223 cadabra::unwrap Class Reference

`#include <unwrap.hh>`

Inheritance diagram for cadabra::unwrap:



Public Member Functions

- unwrap (const Kernel &, Ex &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Public Attributes

- std::vector< Ex > wrappers

Additional Inherited Members

9.223.1 Constructor & Destructor Documentation

9.223.1.1 unwrap::unwrap (const Kernel & *k,* Ex & *tr,* Ex & *w*)

9.223.2 Member Function Documentation

9.223.2.1 Algorithm::result_t unwrap::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.223.2.2 bool unwrap::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.223.3 Member Data Documentation

9.223.3.1 std::vector<Ex> cadabra::unwrap::wrappers

The documentation for this class was generated from the following files:

- core/algorithms/unwrap.hh
- core/algorithms/unwrap.cc

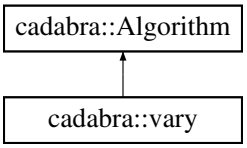
9.224 cadabra::vary Class Reference

9.224.1 Description

Generic variational derivative algorithm.

`#include <vary.hh>`

Inheritance diagram for cadabra::vary:



Public Member Functions

- vary (const Kernel &, Ex &tr, Ex &args)
- virtual bool can_apply (iterator st) override
- virtual result_t apply (iterator &) override

Private Attributes

- Ex & args

Additional Inherited Members

9.224.2 Constructor & Destructor Documentation

9.224.2.1 vary::vary (const Kernel & *k*, Ex & *tr*, Ex & *args*)

9.224.3 Member Function Documentation

9.224.3.1 Algorithm::result_t vary::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.224.3.2 bool vary::can_apply (iterator *st*) [override], [virtual]

Implements cadabra::Algorithm.

9.224.4 Member Data Documentation

9.224.4.1 Ex& cadabra::vary::args [private]

The documentation for this class was generated from the following files:

- core/algorithms/vary.hh
- core/algorithms/vary.cc

9.225 cadabra::VisualCell Class Reference

9.225.1 Description

Structure holding a pointer to one of the possible GUI widgets that can appear in a document.

`#include <VisualCell.hh>`

Public Attributes

- union {
  Gtk::VBox * document
  CodeInput * inbox
  TeXView * outbox
  ImageView * imagebox
  };

  Union of pointers to one of the possible GUI realisations of the cell types declared in DataCell::CellType.

9.225.2 Member Data Documentation

9.225.2.1 union { ... }

Union of pointers to one of the possible GUI realisations of the cell types declared in DataCell::CellType.

All of these cells below should derive from Gtk::VBox.

9.225.2.2 Gtk::VBox* cadabra::VisualCell::document

9.225.2.3 ImageView* cadabra::VisualCell::imagebox

9.225.2.4 CodeInput* cadabra::VisualCell::inbox

9.225.2.5 TeXView* cadabra::VisualCell::outbox

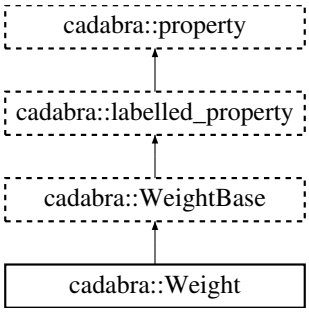
The documentation for this class was generated from the following file:

- frontend/gtkmm/VisualCell.hh

9.226 cadabra::Weight Class Reference

`#include <Weight.hh>`

Inheritance diagram for cadabra::Weight:



Public Member Functions

- virtual multiplier_t value (const Kernel &, Ex::iterator, const std::string &forcedlabel) const override
- virtual bool parse (const Kernel &, keyval_t &) override
- virtual std::string unnamed_argument () const override
- virtual std::string name () const override

Private Attributes

- multiplier_t value_

Additional Inherited Members

9.226.1 Member Function Documentation

9.226.1.1 std::string Weight::name ( ) const [override], [virtual]

Implements cadabra::property.

9.226.1.2 bool Weight::parse (const Kernel & *k,* keyval_t & *kv*) [override], [virtual]

Reimplemented from cadabra::labelled_property.

9.226.1.3 virtual std::string cadabra::Weight::unnamed_argument ( ) const [inline], [override], [virtual]

Reimplemented from cadabra::property.

9.226.1.4 multiplier_t Weight::value (const Kernel & *pr,* Ex::iterator , const std::string & *forcedlabel*) const [override], [virtual]

Implements cadabra::WeightBase.

9.226.2 Member Data Documentation

9.226.2.1 multiplier_t cadabra::Weight::value_ [private]

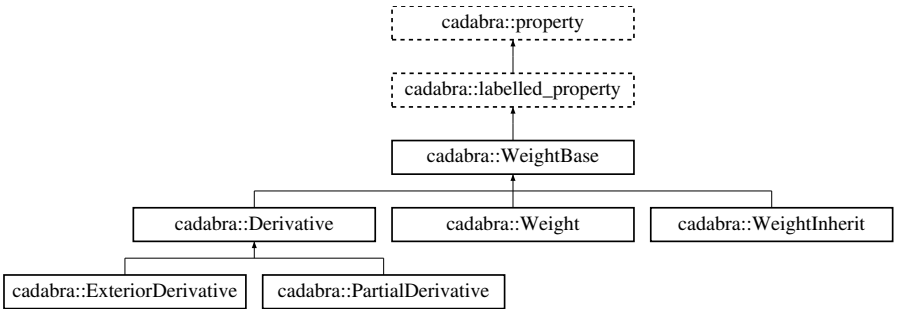
The documentation for this class was generated from the following files:

- core/properties/Weight.hh
- core/properties/Weight.cc

9.227 cadabra::WeightBase Class Reference

`#include <WeightBase.hh>`

Inheritance diagram for cadabra::WeightBase:



Public Member Functions

- virtual multiplier_t value (const Kernel &, Ex::iterator, const std::string &forcedlabel) const =0

Additional Inherited Members

9.227.1 Member Function Documentation

9.227.1.1 virtual multiplier_t cadabra::WeightBase::value (const Kernel & , Ex::iterator , const std::string & *forcedlabel*) const [pure virtual]

Implemented in cadabra::Derivative, cadabra::WeightInherit, and cadabra::Weight.

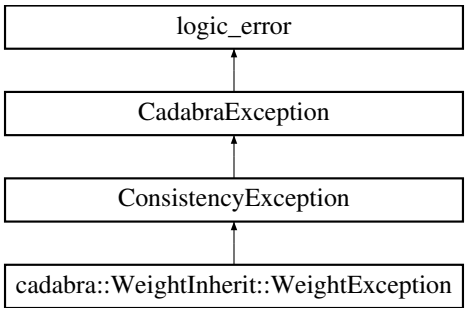
The documentation for this class was generated from the following file:

- core/properties/WeightBase.hh

9.228 cadabra::WeightInherit::WeightException Class Reference

`#include <WeightInherit.hh>`

Inheritance diagram for cadabra::WeightInherit::WeightException:



Public Member Functions

- WeightException (const std::string &)

9.228.1 Constructor & Destructor Documentation

9.228.1.1 WeightInherit::WeightException::WeightException (const std::string & *str*)

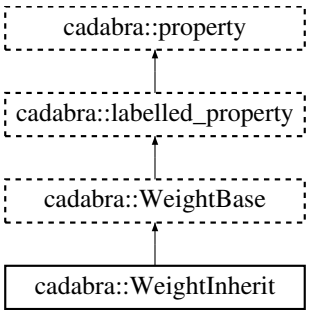
The documentation for this class was generated from the following files:

- core/properties/WeightInherit.hh
- core/properties/WeightInherit.cc

9.229 cadabra::WeightInherit Class Reference

`#include <WeightInherit.hh>`

Inheritance diagram for cadabra::WeightInherit:



Classes

- class WeightException

Public Types

- enum { multiplicative, additive, power }

Public Member Functions

- virtual bool parse (const Kernel &, keyval_t &) override
- virtual multiplier_t value (const Kernel &, Ex::iterator, const std::string &forcedlabel) const override
- virtual std::string unnamed_argument () const override
- virtual std::string name () const override

Public Attributes

- enum cadabra::WeightInherit:: { ... } combination_type
- multiplier_t value_self

9.229.1 Member Enumeration Documentation

9.229.1.1 anonymous enum

Enumerator

multiplicative

additive

power

9.229.2 Member Function Documentation

9.229.2.1 std::string WeightInherit::name ( ) const [override], [virtual]

Implements cadabra::property.

9.229.2.2 bool WeightInherit::parse (const Kernel & *k,* keyval_t & *kv*) [override], [virtual]

Reimplemented from cadabra::labelled_property.

9.229.2.3 virtual std::string cadabra::WeightInherit::unnamed_argument ( ) const [inline], [override], [virtual]

Reimplemented from cadabra::property.

9.229.2.4 multiplier_t WeightInherit::value (const Kernel & *kernel,* Ex::iterator *it,* const std::string & *forcedlabel*) const [override], [virtual]

Implements cadabra::WeightBase.

9.229.3 Member Data Documentation

9.229.3.1 enum { ... } cadabra::WeightInherit::combination_type

9.229.3.2 multiplier_t cadabra::WeightInherit::value_self

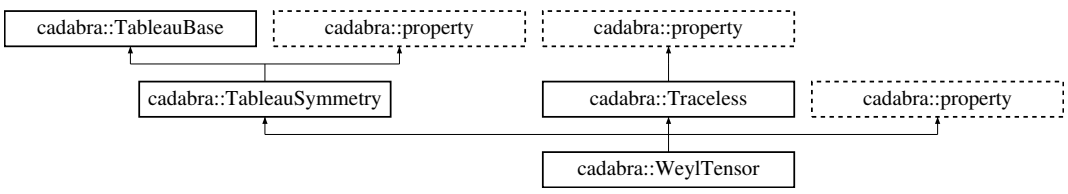
The documentation for this class was generated from the following files:

- core/properties/WeightInherit.hh
- core/properties/WeightInherit.cc

9.230 cadabra::WeylTensor Class Reference

`#include <WeylTensor.hh>`

Inheritance diagram for cadabra::WeylTensor:



Public Member Functions

- WeylTensor ()
- virtual std::string name () const override
- virtual void validate (const Kernel &, const Ex &) const override
- virtual void latex (std::ostream &) const override

  Display the property on the stream.

Additional Inherited Members

9.230.1 Constructor & Destructor Documentation

9.230.1.1 WeylTensor::WeylTensor ( )

9.230.2 Member Function Documentation

9.230.2.1 void WeylTensor::latex (std::ostream & *str*) const [override], [virtual]

Display the property on the stream.

Generate a LaTeX representation of the property, assuming LaTeX is in text mode (so it needs dollar symbols to switch to maths).

Reimplemented from cadabra::property.

9.230.2.2 std::string WeylTensor::name ( ) const [override], [virtual]

Implements cadabra::property.

9.230.2.3 void WeylTensor::validate (const Kernel & *kernel,* const Ex & *pat*) const [override], [virtual]

Reimplemented from cadabra::property.

The documentation for this class was generated from the following files:

- core/properties/WeylTensor.hh
- core/properties/WeylTensor.cc

9.231 X Struct Reference

9.231.1 Description

Mockup types.

The documentation for this struct was generated from the following file:

- core/ptest.cc

9.232 Y Struct Reference

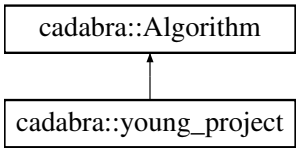
The documentation for this struct was generated from the following file:

- core/ptest.cc

9.233 cadabra::young_project Class Reference

`#include <young_project.hh>`

Inheritance diagram for cadabra::young_project:



Public Types

- typedef yngtab::filled_tableau< unsigned int > pos_tab_t
- typedef yngtab::filled_tableau< iterator > name_tab_t

Public Member Functions

- young_project (const Kernel &, Ex &)
- young_project (const Kernel &, Ex &, const std::vector< int > &shape, const std::vector< int > &indices)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Public Attributes

- pos_tab_t tab
- name_tab_t nametab
- combin::symmetriser< unsigned int > sym
- combin::combinations< unsigned int >::permuted_sets_t asym_ranges
- bool remove_traces

Private Member Functions

- iterator nth_index_node (iterator, unsigned int)

Additional Inherited Members

9.233.1 Member Typedef Documentation

9.233.1.1 typedef yngtab::filled_tableau<iterator> cadabra::young_project::name_tab_t

9.233.1.2 typedef yngtab::filled_tableau<unsigned int> cadabra::young_project::pos_tab_t

9.233.2 Constructor & Destructor Documentation

9.233.2.1 young_project::young_project (const Kernel & *k*, Ex & *tr*)

9.233.2.2 young_project::young_project (const Kernel & *k*, Ex & *tr*, const std::vector< int > & *shape*, const std::vector< int > & *indices*)

9.233.3 Member Function Documentation

9.233.3.1 Algorithm::result_t young_project::apply (iterator & *it*) [virtual]

Implements cadabra::Algorithm.

9.233.3.2 bool young_project::can_apply (iterator *it*) [virtual]

Implements cadabra::Algorithm.

9.233.3.3 Ex::iterator young_project::nth_index_node (iterator *it*, unsigned int *num*) [private]

9.233.4 Member Data Documentation

9.233.4.1 combin::combinations<unsigned int>::permuted_sets_t cadabra::young_project::asym_ranges

9.233.4.2 name_tab_t cadabra::young_project::nametab

9.233.4.3 bool cadabra::young_project::remove_traces

9.233.4.4 combin::symmetriser<unsigned int> cadabra::young_project::sym

9.233.4.5 pos_tab_t cadabra::young_project::tab

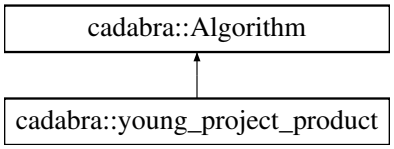
The documentation for this class was generated from the following files:

- core/algorithms/young_project.hh
- core/algorithms/young_project.cc

9.234 cadabra::young_project_product Class Reference

`#include <young_project_product.hh>`

Inheritance diagram for cadabra::young_project_product:



Public Member Functions

- young_project_product (const Kernel &, Ex &)
- virtual bool can_apply (iterator) override
- virtual result_t apply (iterator &) override

Additional Inherited Members

9.234.1 Constructor & Destructor Documentation

9.234.1.1 young_project_product::young_project_product (const Kernel & *k*, Ex & *tr*)

9.234.2 Member Function Documentation

9.234.2.1 Algorithm::result_t young_project_product::apply (iterator & *it*) [override], [virtual]

Implements cadabra::Algorithm.

9.234.2.2 bool young_project_product::can_apply (iterator *it*) [override], [virtual]

Implements cadabra::Algorithm.

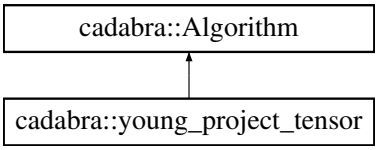
The documentation for this class was generated from the following files:

- core/algorithms/young_project_product.hh
- core/algorithms/young_project_product.cc

9.235 cadabra::young_project_tensor Class Reference

`#include <young_project_tensor.hh>`

Inheritance diagram for cadabra::young_project_tensor:



Public Member Functions

- young_project_tensor (const Kernel &, Ex &, bool)
- virtual bool can_apply (iterator)
- virtual result_t apply (iterator &)

Public Attributes

- bool modulo_monoterm
- combin::symmetriser< unsigned int > sym

Private Attributes

- const TableauBase * tb

Additional Inherited Members

9.235.1 Constructor & Destructor Documentation

9.235.1.1 young_project_tensor::young_project_tensor (const Kernel & *k,* Ex & *tr,* bool *modmono*)

9.235.2 Member Function Documentation

9.235.2.1 Algorithm::result_t young_project_tensor::apply (iterator & *it*) [virtual]

Implements cadabra::Algorithm.

9.235.2.2 bool young_project_tensor::can_apply (iterator *it*) [virtual]

Implements cadabra::Algorithm.

9.235.3 Member Data Documentation

9.235.3.1 bool cadabra::young_project_tensor::modulo_monoterm

9.235.3.2 combin::symmetriser<unsigned int> cadabra::young_project_tensor::sym

9.235.3.3 const TableauBase* cadabra::young_project_tensor::tb [private]

The documentation for this class was generated from the following files:

- core/algorithms/young_project_tensor.hh
- core/algorithms/young_project_tensor.cc

9.236 Z Struct Reference

Public Member Functions

- Z ()

9.236.1 Constructor & Destructor Documentation

9.236.1.1 Z::Z ( )

The documentation for this struct was generated from the following file:

- core/ptest.cc

Chapter 10

File Documentation

10.1 client_server/Actions.cc File Reference

```
#include "Actions.hh"
#include "DataCell.hh"
#include "DocumentThread.hh"
#include "GUIBase.hh"
#include <iostream>
```

10.2 client_server/Actions.hh File Reference

```
#include "DataCell.hh"
#include "DocumentThread.hh"
#include <memory>
```

Classes

- class cadabra::ActionBase

  All actions derive from the ActionBase object, which defines the interface they need to implement.
- class cadabra::ActionAddCell

  Add a cell to the notebook.
- class cadabra::ActionPositionCursor

  Position the cursor relative to the indicated cell.
- class cadabra::ActionSetRunStatus

  Update the running status of the indicated cell.
- class cadabra::ActionRemoveCell

  Remove a cell and all its child cells from the document.
- class cadabra::ActionSplitCell

  Split a cell into two separate cells, at the point of the cursor.
- class cadabra::ActionInsertText

  Add a text string (can be just a single character) at the point of the cursor.
- class cadabra::ActionEraseText

  Remove a text string starting at the indicated position, and with the indicated length, from the indicated cell.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.3 client_server/cadabra-server.cc File Reference

```
#include "Snoop.hh"
#include "Server.hh"
```

Functions

- int main ()

10.3.1 Function Documentation

10.3.1.1 int main ( )

10.4 client_server/cadabra2cadabra.cc File Reference

```
#include <fstream>
#include <iostream>
#include "DataCell.hh"
```

Functions

- int main (int argc, char **argv)

10.4.1 Function Documentation

10.4.1.1 int main (int *argc*, char ** *argv*)

10.5 client_server/cadabra2html.cc File Reference

```
#include <fstream>
#include <iostream>
#include "DataCell.hh"
```

Functions

- int main (int argc, char **argv)

10.5.1 Function Documentation

10.5.1.1 int main (int *argc*, char ** *argv*)

10.6 client_server/ComputeThread.cc File Reference

```
#include <string>
#include <iostream>
#include <sstream>
#include "ComputeThread.hh"
#include "DocumentThread.hh"
#include "GUIBase.hh"
#include "Actions.hh"
#include "popen2.hh"
#include <sys/types.h>
#include <signal.h>
#include <glibmm/spawn.h>
```

Typedefs

- typedef websocketpp::client< websocketpp::config::asio_client > client
- using SlotSpawnChildSetup = sigc::slot< void()>

10.6.1 Typedef Documentation

10.6.1.1 typedef websocketpp::client<websocketpp::config::asio_client> client

10.6.1.2 using SlotSpawnChildSetup = sigc::slot<void()>

10.7 client_server/ComputeThread.hh File Reference

```
#include <websocketpp/config/asio_no_tls_client.hpp>
#include <websocketpp/client.hpp>
#include <websocketpp/common/thread.hpp>
#include <websocketpp/common/functional.hpp>
#include <thread>
#include <glibmm/spawn.h>
#include "DataCell.hh"
```

Classes

- class cadabra::ComputeThread

  Base class which talks to the server and sends Action objects back to the DocumentThread.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Typedefs

- typedef websocketpp::client< websocketpp::config::asio_client > WSClient
- typedef websocketpp::config::asio_client::message_type::ptr message_ptr
- typedef websocketpp::lib::lock_guard< websocketpp::lib::mutex > scoped_lock

10.7.1 Typedef Documentation

10.7.1.1 typedef websocketpp::config::asio_client::message_type::ptr message_ptr

10.7.1.2 typedef websocketpp::lib::lock_guard<websocketpp::lib::mutex> scoped_lock

10.7.1.3 typedef websocketpp::client<websocketpp::config::asio_client> WSClient

10.8 client_server/DataCell.cc File Reference

```
#include "Config.hh"
#include "DataCell.hh"
#include <sstream>
#include <fstream>
#include <boost/regex.hpp>
#include <boost/filesystem.hpp>
#include <boost/algorithm/string.hpp>
#include <iostream>
```

10.9 client_server/DataCell.hh File Reference

```
#include <string>
#include <mutex>
#include "tree.hh"
#include "json/json.h"
```

Classes

- class cadabra::DataCell

  DataCells are the basic building blocks for a document.
- class cadabra::DataCell::id_t

  Each cell is identified by a serial number 'id' which is used to keep track of it across network calls, and a bool indicating whether the client or the server has created this cell.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Typedefs

- typedef tree< DataCell > cadabra::DTree

Functions

- std::string cadabra::JSON_serialise (const DTree &)

  Serialise a document into .cj format, which is a JSON version of the document tree.
- void cadabra::JSON_recurse (const DTree &, DTree::iterator, Json::Value &)
- void cadabra::JSON_deserialise (const std::string &, DTree &)

  Load a document from .cj format, i.e. the inverse of the above.
- void cadabra::JSON_in_recurse (DTree &doc, DTree::iterator loc, const Json::Value &cells)
- std::string cadabra::export_as_HTML (const DTree &doc, bool for_embedding=false, std::string title="")

  Export a document to a single self-contained HTML file containing inline CSS.
- void cadabra::HTML_recurse (const DTree &doc, DTree::iterator it, std::ostringstream &str, const std::string &preamble_string, bool for_embedding=false, std::string title="")
- std::string cadabra::latex_to_html (const std::string &)

  Convert various LaTeX constructions to HTML-with-Mathjax, e.g.
- std::string cadabra::export_as_LaTeX (const DTree &doc)

  Export a document to a single self-contained LaTeX file.
- void cadabra::LaTeX_recurse (const DTree &doc, DTree::iterator it, std::ostringstream &str, const std::string &preamble_string)
- std::string cadabra::export_as_python (const DTree &doc)

  Export a document to a python-like file (converting text cells to comments and python cells to python code, dropping output cells).
- void cadabra::python_recurse (const DTree &doc, DTree::iterator it, std::ostringstream &str)

10.10 client_server/DocumentThread.cc File Reference

```
#include "Actions.hh"
#include "DocumentThread.hh"
#include "GUIBase.hh"
#include <typeinfo>
#include <iostream>
#include <set>
#include <string>
#include <fstream>
#include <boost/config.hpp>
#include <boost/program_options/detail/config_file.hpp>
#include <boost/program_options/parsers.hpp>
#include <unistd.h>
#include <sys/types.h>
#include <glibmm/miscutils.h>
#include "Snoop.hh"
#include "Config.hh"
```

10.11 client_server/DocumentThread.hh File Reference

```
#include <queue>
#include <mutex>
#include <stack>
#include "DataCell.hh"
#include "tree.hh"
```

Classes

- class cadabra::DocumentThread

  A base class with all the logic to manipulate a Cadabra notebook document.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.12 client_server/GUIBase.hh File Reference

```
#include <deque>
#include "DataCell.hh"
```

Classes

- class cadabra::GUIBase

  Abstract base class with methods that need to be implemented by any GUI.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.13 client_server/popen2.cc File Reference

```
#include "popen2.hh"
#include <sys/wait.h>
#include <stdexcept>
```

Macros

- #define READ 0
- #define WRITE 1

Functions

- FILE * popen2 (std::string command, std::string type, int &pid)
- int pclose2 (FILE *fp, pid_t pid)

10.13.1 Macro Definition Documentation

10.13.1.1 #define READ 0

10.13.1.2 #define WRITE 1

10.13.2 Function Documentation

10.13.2.1 int pclose2 (FILE * *fp*, pid_t *pid*)

10.13.2.2 FILE* popen2 (std::string *command*, std::string *type*, int & *pid*)

10.14 client_server/popen2.hh File Reference

```
#include <unistd.h>
#include <errno.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <string>
```

Functions

- FILE * popen2 (std::string command, std::string type, int &pid)
- int pclose2 (FILE *fp, pid_t pid)

10.14.1 Function Documentation

10.14.1.1 int pclose2 (FILE * *fp*, pid_t *pid*)

10.14.1.2 FILE* popen2 (std::string *command*, std::string *type*, int & *pid*)

10.15 client_server/regexp_tester.cc File Reference

```
#include <iostream>
#include <string>
#include <regex>
```

Functions

- int main (int argc, char **argv)

10.15.1 Function Documentation

10.15.1.1 int main (int *argc,* char ** *argv*)

10.16 client_server/Server.cc File Reference

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <thread>
#include <boost/regex.hpp>
#include <boost/uuid/uuid_generators.hpp>
#include <boost/uuid/uuid_io.hpp>
#include <boost/algorithm/string/replace.hpp>
#include <json/json.h>
#include "Snoop.hh"
#include "CdbPython.hh"
#include "Server.hh"
#include "SympyCdb.hh"
```

Functions

- boost::python::list ProgressMonitor_totals_helper (ProgressMonitor &self)
- int quit (void *)

10.16.1 Function Documentation

10.16.1.1 boost::python::list ProgressMonitor_totals_helper (ProgressMonitor & *self*)

10.16.1.2 int quit (void *)

10.17 client_server/Server.hh File Reference

```
#include <string>
#include <boost/uuid/uuid.hpp>
#include <websocketpp/server.hpp>
#include <websocketpp/config/asio_no_tls.hpp>
#include <websocketpp/common/functional.hpp>
#include <future>
#include <boost/python.hpp>
#include "Stopwatch.hh"
#include "ProgressMonitor.hh"
```

Classes

- class Server

  Object representing a Cadabra server, capable of receiving messages on a websocket, running Python code, and sending output back to the client.
- class Server::CatchOutput

  Python output catching.
- class Server::Connection
- class Server::Block

10.18 client_server/Snoop.cc File Reference

```
#include "Snoop.hh"
#include "SnoopPrivate.hh"
#include <iostream>
#include <string.h>
#include <regex>
#include <chrono>
#include <ctime>
#include <sys/utsname.h>
#include <stdint.h>
#include <json/json.h>
#include <set>
#include <unistd.h>
#include <sys/types.h>
#include <pwd.h>
#include <glibmm/miscutils.h>
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/json_parser.hpp>
#include <boost/config.hpp>
#include <boost/program_options/detail/config_file.hpp>
#include <boost/program_options/parsers.hpp>
#include <boost/uuid/uuid.hpp>
#include <boost/uuid/uuid_generators.hpp>
#include <boost/uuid/uuid_io.hpp>
```

Macros

- #define BOOST_SPIRIT_THREADSAFE

Typedefs

- using u64_millis = std::chrono::duration< uint64_t, std::milli >

Functions

- std::string safestring (const unsigned char *c)

10.18.1 Macro Definition Documentation

10.18.1.1 #define BOOST_SPIRIT_THREADSAFE

10.18.2 Typedef Documentation

10.18.2.1 using u64_millis = std::chrono::duration<uint64_t, std::milli>

10.18.3 Function Documentation

10.18.3.1 std::string safestring (const unsigned char * c)

10.19 client_server/Snoop.hh File Reference

```
#include <string>
#include <sstream>
#include <sqlite3.h>
#include <stdint.h>
#include <mutex>
#include <json/json.h>
#include <thread>
```

Classes

- class snoop::Flush
- class snoop::Snoop

  Logging class with functionality to send log information to a remote server using a websocket connection.
- class snoop::Snoop::AppEntry

  C++ representation of an run entry.
- class snoop::Snoop::LogEntry

  C++ representation of a log entry.

Namespaces

- snoop

Macros

- #define LOC __FILE__, __LINE__, __func__

Variables

- Flush snoop::flush
- Snoop snoop::log
- const char snoop::info [ ] ="info"
- const char snoop::warn [ ] ="warning"
- const char snoop::error [ ] ="error"
- const char snoop::fatal [ ] ="fatal"
- const char snoop::email [ ] ="email"

10.19.1 Macro Definition Documentation

10.19.1.1 #define LOC __FILE__, __LINE__, __func__

10.20 client_server/SnoopPrivate.hh File Reference

```
#include <websocketpp/config/asio_no_tls_client.hpp>
#include <websocketpp/client.hpp>
#include <websocketpp/common/thread.hpp>
#include <websocketpp/common/functional.hpp>
#include <mutex>
#include <condition_variable>
#include <thread>
```

Classes

- class snoop::SnoopImpl

Namespaces

- snoop

Typedefs

- typedef websocketpp::client< websocketpp::config::asio_client > WebsocketClient
- typedef websocketpp::config::asio_client::message_type::ptr message_ptr

10.20.1 Typedef Documentation

10.20.1.1 typedef websocketpp::config::asio_client::message_type::ptr message_ptr

10.20.1.2 typedef websocketpp::client<websocketpp::config::asio_client> WebsocketClient

10.21 client_server/test_client.cc File Reference

```
#include "Client.hh"
#include <thread>
#include <system_error>
```

Classes

- class MyClient
- class UI

Functions

- int main (int, char **)

Variables

- MyClient client
- UI ui

10.21.1 Function Documentation

10.21.1.1 int main (int , char **)

10.21.2 Variable Documentation

10.21.2.1 MyClient client

10.21.2.2 UI ui

10.22 client_server/test_talk_to_server.cc File Reference

```
#include <websocketpp/client.hpp>
#include <websocketpp/config/asio_no_tls_client.hpp>
#include <websocketpp/common/functional.hpp>
```

Typedefs

- typedef websocketpp::client< websocketpp::config::asio_client > client
- typedef websocketpp::config::asio_client::message_type::ptr message_ptr

Functions

- void on_open (client *c, websocketpp::connection_hdl hdl)
- void on_message (client *c, websocketpp::connection_hdl hdl, message_ptr msg)
- int main (int argc, char **argv)

Variables

- bool stopit =false

10.22.1 Typedef Documentation

10.22.1.1 typedef websocketpp::client<websocketpp::config::asio_client> client

10.22.1.2 typedef websocketpp::config::asio_client::message_type::ptr message_ptr

10.22.2 Function Documentation

10.22.2.1 int main (int *argc,* char ** *argv*)

10.22.2.2 void on_message (client * *c,* websocketpp::connection_hdl *hdl,* message_ptr *msg*)

10.22.2.3 void on_open (client * *c,* websocketpp::connection_hdl *hdl*)

10.22.3 Variable Documentation

10.22.3.1 bool stopit =false

10.23 core/Algorithm.cc File Reference

```
#include <stddef.h>
#include "Algorithm.hh"
#include "Storage.hh"
#include "Props.hh"
#include "Cleanup.hh"
#include <typeinfo>
#include <cxxabi.h>
#include "properties/Derivative.hh"
#include "properties/Indices.hh"
#include "properties/Coordinate.hh"
#include "properties/Symbol.hh"
#include "properties/DependsBase.hh"
#include <sstream>
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.24 core/Algorithm.hh File Reference

```
#include "Stopwatch.hh"
#include "Storage.hh"
#include "Compare.hh"
#include "Props.hh"
#include "Exceptions.hh"
#include "Kernel.hh"
#include "IndexIterator.hh"
#include "ProgressMonitor.hh"
#include <map>
#include <fstream>
#include <cstddef>
```

Classes

- class cadabra::Algorithm

  Base class for all algorithms, containing generic routines and in particular the logic for index classification.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.25 core/algorithms/canonicalise.cc File Reference

```
#include "Cleanup.hh"
#include "Exchange.hh"
#include "Functional.hh"
#include "algorithms/canonicalise.hh"
#include "modules/xperm_new.h"
#include "properties/Traceless.hh"
#include "properties/Diagonal.hh"
#include "properties/Derivative.hh"
#include "properties/AntiCommuting.hh"
```

10.26 core/algorithms/canonicalise.hh File Reference

```
#include "Algorithm.hh"
#include "properties/Indices.hh"
#include <vector>
```

Classes

- class cadabra::canonicalise

  Canonicalise the index structure of a tensorial expression.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.27 core/algorithms/collect_components.cc File Reference

```
#include "algorithms/collect_components.hh"
#include "algorithms/evaluate.hh"
```

10.28 core/algorithms/collect_components.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::collect_components

  Collect terms inside a sum, merging their substitution rules.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.29 core/algorithms/collect_factors.cc File Reference

```
#include "Props.hh"
#include "Compare.hh"
#include "Cleanup.hh"
#include "algorithms/collect_factors.hh"
#include "algorithms/collect_terms.hh"
#include "properties/Symbol.hh"
```

10.30 core/algorithms/collect_factors.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::collect_factors

  Collect symbolically equal factors in a product.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.31 core/algorithms/collect_terms.cc File Reference

```
#include "algorithms/collect_terms.hh"
```

10.32 core/algorithms/collect_terms.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::collect_terms

  Collect symbolically equal terms in a sum.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.33 core/algorithms/combine.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/combine.hh"
#include "properties/Matrix.hh"
```

10.34 core/algorithms/combine.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::combine

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.35 core/algorithms/complete.cc File Reference

```
#include "algorithms/complete.hh"
#include "Functional.hh"
#include "properties/Metric.hh"
#include "properties/InverseMetric.hh"
#include "SympyCdb.hh"
```

10.36 core/algorithms/complete.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::complete

  Complete a set of coordinate rules so that they also cover related tensors.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.37 core/algorithms/component.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class component

10.38 core/algorithms/decompose_product.cc File Reference

```
#include "Cleanup.hh"
#include "IndexIterator.hh"
#include "Exceptions.hh"
#include "algorithms/decompose_product.hh"
#include "properties/Integer.hh"
```

10.39 core/algorithms/decompose_product.hh File Reference

```
#include "algorithms/young_project.hh"
#include "properties/TableauBase.hh"
#include "properties/Indices.hh"
```

Classes

- class cadabra::decompose_product

  Decompose a product of tensors by applying Young projectors.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.40 core/algorithms/distribute.cc File Reference

```
#include "distribute.hh"
#include "Cleanup.hh"
#include "properties/Distributable.hh"
#include "algorithms/flatten_product.hh"
```

10.41 core/algorithms/distribute.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::distribute

  Distribute factors over a sum, that is, $(A+B)C \to AC+BC$.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.42 core/algorithms/drop_weight.cc File Reference

```
#include "algorithms/drop_weight.hh"
#include "Cleanup.hh"
#include "Props.hh"
#include "properties/Weight.hh"
#include "properties/WeightInherit.hh"
```

10.43 core/algorithms/drop_weight.hh File Reference

```
#include "Algorithm.hh"
#include "properties/WeightInherit.hh"
#include "properties/Weight.hh"
#include <string>
```

Classes

- class cadabra::drop_keep_weight
- class cadabra::drop_weight
- class cadabra::keep_weight

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.44 core/algorithms/einsteinify.cc File Reference

```
#include "algorithms/einsteinify.hh"
```

10.45 core/algorithms/einsteinify.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::einsteinify

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.46 core/algorithms/eliminate_kronecker.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/eliminate_kronecker.hh"
#include "properties/KroneckerDelta.hh"
#include "properties/Integer.hh"
```

10.47 core/algorithms/eliminate_kronecker.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::eliminate_kronecker

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.48 core/algorithms/eliminate_metric.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/eliminate_metric.hh"
#include "properties/Metric.hh"
#include "properties/InverseMetric.hh"
```

10.49 core/algorithms/eliminate_metric.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::eliminate_converter
- class cadabra::eliminate_metric

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.50 core/algorithms/epsilon_to_delta.cc File Reference

```
#include "algorithms/epsilon_to_delta.hh"
#include "algorithms/reduce_delta.hh"
#include "properties/EpsilonTensor.hh"
#include "properties/Metric.hh"
```

10.51 core/algorithms/epsilon_to_delta.hh File Reference

```
#include "Algorithm.hh"
#include <vector>
```

Classes

- class cadabra::epsilon_to_delta

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.52 core/algorithms/evaluate.cc File Reference

```
#include "Functional.hh"
#include "Cleanup.hh"
#include "Permutations.hh"
#include "SympyCdb.hh"
#include "algorithms/evaluate.hh"
#include "algorithms/substitute.hh"
#include "properties/EpsilonTensor.hh"
#include "properties/PartialDerivative.hh"
#include "properties/Coordinate.hh"
#include "properties/Depends.hh"
#include "properties/Accent.hh"
#include <functional>
```

10.53 core/algorithms/evaluate.hh File Reference

```
#include "Compare.hh"
#include "Algorithm.hh"
#include "properties/Indices.hh"
```

Classes

- class cadabra::evaluate

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.54 core/algorithms/expand.cc File Reference

```
#include "Cleanup.hh"
#include "Exceptions.hh"
#include "algorithms/expand.hh"
#include "properties/Matrix.hh"
```

10.55 core/algorithms/expand.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::expand

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.56 core/algorithms/expand_delta.cc File Reference

```
#include "algorithms/expand_delta.hh"
#include "properties/KroneckerDelta.hh"
#include "properties/TableauBase.hh"
#include "Cleanup.hh"
```

10.57 core/algorithms/expand_delta.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::expand_delta

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.58 core/algorithms/expand_diracbar.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/expand_diracbar.hh"
#include "properties/DiracBar.hh"
#include "properties/Spinor.hh"
#include "properties/GammaMatrix.hh"
```

10.59 core/algorithms/expand_diracbar.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::expand_diracbar

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.60 core/algorithms/expand_power.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/expand_power.hh"
```

10.61 core/algorithms/expand_power.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::expand_power

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.62 core/algorithms/factor_in.cc File Reference

```
#include "Functional.hh"
#include "algorithms/factor_in.hh"
```

10.63 core/algorithms/factor_in.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::factor_in

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.64 core/algorithms/factor_out.cc File Reference

```
#include "Cleanup.hh"
#include "Functional.hh"
#include "algorithms/factor_out.hh"
#include "algorithms/sort_product.hh"
#include <map>
```

10.65 core/algorithms/factor_out.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::factor_out

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.66 core/algorithms/fierz.cc File Reference

```
#include "Cleanup.hh"
#include "Exceptions.hh"
#include "algorithms/fierz.hh"
#include "properties/GammaMatrix.hh"
#include "properties/Integer.hh"
#include "properties/DiracBar.hh"
```

10.67 core/algorithms/fierz.hh File Reference

```
#include "Algorithm.hh"
#include "properties/Indices.hh"
#include "properties/Spinor.hh"
```

Classes

- class cadabra::fierz

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.68 core/algorithms/flatten_product.cc File Reference

```
#include "algorithms/flatten_product.hh"
#include "properties/PartialDerivative.hh"
```

10.69 core/algorithms/flatten_product.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::flatten_product

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.70 core/algorithms/flatten_sum.cc File Reference

```
#include "algorithms/flatten_sum.hh"
```

10.71 core/algorithms/flatten_sum.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::flatten_sum

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.72 core/algorithms/indexsort.cc File Reference

```
#include "IndexIterator.hh"
#include "properties/TableauSymmetry.hh"
#include "algorithms/indexsort.hh"
```

10.73 core/algorithms/indexsort.hh File Reference

```
#include "Algorithm.hh"
#include "properties/TableauBase.hh"
```

Classes

- class cadabra::indexsort
- class cadabra::indexsort::less_indexed_treenode

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.74 core/algorithms/integrate_by_parts.cc File Reference

```
#include <iterator>
#include "algorithms/integrate_by_parts.hh"
#include "properties/Derivative.hh"
#include "Cleanup.hh"
```

10.75 core/algorithms/integrate_by_parts.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::integrate_by_parts

  Integrate by parts away from the indicated derivative object.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.76 core/algorithms/join_gamma.cc File Reference

```
#include "Storage.hh"
#include "algorithms/join_gamma.hh"
#include "Exceptions.hh"
#include "Cleanup.hh"
#include "properties/Integer.hh"
```

10.77 core/algorithms/join_gamma.hh File Reference

```
#include "Algorithm.hh"
#include "properties/GammaMatrix.hh"
```

Classes

- class cadabra::join_gamma

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.78 core/algorithms/keep_terms.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/keep_terms.hh"
```

10.79 core/algorithms/keep_terms.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::keep_terms

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.80 core/algorithms/lr_tensor.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/lr_tensor.hh"
#include "properties/Tableau.hh"
#include "properties/FilledTableau.hh"
```

10.81 core/algorithms/lr_tensor.hh File Reference

```
#include "Algorithm.hh"
#include "algorithms/tab_basics.hh"
```

Classes

- class cadabra::lr_tensor

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.82 core/algorithms/map_sympy.cc File Reference

```
#include "algorithms/map_sympy.hh"
#include "properties/Coordinate.hh"
#include "properties/Symbol.hh"
#include "SympyCdb.hh"
```

Macros

- #define DEBUG 1

10.82.1 Macro Definition Documentation

10.82.1.1 #define DEBUG 1

10.83 core/algorithms/map_sympy.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::map_sympy

  Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.84 core/algorithms/order.cc File Reference

```
#include "algorithms/order.hh"
#include "Combinatorics.hh"
```

10.85 core/algorithms/order.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::order

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.86 core/algorithms/product_rule.cc File Reference

```
#include "Compare.hh"
#include "Cleanup.hh"
#include "algorithms/product_rule.hh"
#include "properties/Derivative.hh"
#include "properties/DifferentialForm.hh"
```

10.87 core/algorithms/product_rule.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::product_rule

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.88 core/algorithms/reduce_delta.cc File Reference

```
#include "Exceptions.hh"
#include "algorithms/reduce_delta.hh"
#include "properties/Integer.hh"
#include "properties/KroneckerDelta.hh"
```

10.89 core/algorithms/reduce_delta.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::reduce_delta

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.90 core/algorithms/rename_dummies.cc File Reference

```
#include "algorithms/rename_dummies.hh"
#include "properties/Indices.hh"
#include "Exceptions.hh"
```

10.91 core/algorithms/rename_dummies.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::rename_dummies

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.92 core/algorithms/replace_match.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/replace_match.hh"
#include "algorithms/substitute.hh"
```

10.93 core/algorithms/replace_match.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::replace_match

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.94 core/algorithms/rewrite_indices.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/rewrite_indices.hh"
#include "properties/Derivative.hh"
#include "properties/Indices.hh"
```

10.95 core/algorithms/rewrite_indices.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::rewrite_indices

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.96 core/algorithms/sort_product.cc File Reference

```
#include "algorithms/sort_product.hh"
```

10.97 core/algorithms/sort_product.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::sort_product

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.98 core/algorithms/sort_spinors.cc File Reference

```
#include "algorithms/sort_spinors.hh"
#include "properties/GammaMatrix.hh"
#include "properties/Spinor.hh"
#include "properties/DiracBar.hh"
#include "properties/SortOrder.hh"
```

10.99 core/algorithms/sort_spinors.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::sort_spinors

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.100 core/algorithms/sort_sum.cc File Reference

```
#include "properties/SortOrder.hh"
#include "algorithms/sort_sum.hh"
```

10.101 core/algorithms/sort_sum.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::sort_sum

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.102 core/algorithms/split.cc File Reference

```
#include "split.hh"
```

10.103 core/algorithms/split.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class split

  Split an expression into terms or factors.

10.104 core/algorithms/split_gamma.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/split_gamma.hh"
#include "algorithms/join_gamma.hh"
```

10.105 core/algorithms/split_gamma.hh File Reference

```
#include "Algorithm.hh"
#include "properties/GammaMatrix.hh"
```

Classes

- class cadabra::split_gamma

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.106 core/algorithms/split_index.cc File Reference

```
#include "algorithms/split_index.hh"
```

10.107 core/algorithms/split_index.hh File Reference

```
#include "Algorithm.hh"
#include "properties/Indices.hh"
#include "properties/Coordinate.hh"
```

Classes

- class cadabra::split_index

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.108 core/algorithms/substitute.cc File Reference

```
#include <sstream>
#include "Cleanup.hh"
#include "Functional.hh"
#include "algorithms/substitute.hh"
#include "properties/Indices.hh"
```

10.109 core/algorithms/substitute.hh File Reference

```
#include "Algorithm.hh"
#include "algorithms/sort_product.hh"
```

Classes

- class cadabra::substitute

  Generic substitution algorithm.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.110 core/algorithms/sym.cc File Reference

```
#include "algorithms/sym.hh"
```

10.111 core/algorithms/sym.hh File Reference

```
#include "Algorithm.hh"
#include <vector>
#include "Combinatorics.hh"
```

Classes

- class cadabra::sym

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.112 core/algorithms/tab_basics.cc File Reference

```
#include "algorithms/tab_basics.hh"
```

10.113 core/algorithms/tab_basics.hh File Reference

```
#include "Algorithm.hh"
#include "YoungTab.hh"
```

Classes

- class cadabra::tab_basics

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.114 core/algorithms/take_match.cc File Reference

```
#include "algorithms/take_match.hh"
#include "algorithms/substitute.hh"
#include "Cleanup.hh"
```

10.115 core/algorithms/take_match.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::take_match

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.116 core/algorithms/unwrap.cc File Reference

```
#include "Compare.hh"
#include "Cleanup.hh"
#include "algorithms/unwrap.hh"
#include "properties/Derivative.hh"
#include "properties/Accent.hh"
#include "properties/DiracBar.hh"
#include "properties/Spinor.hh"
#include "properties/GammaMatrix.hh"
#include "properties/DependsBase.hh"
```

10.117 core/algorithms/unwrap.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::unwrap

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.118 core/algorithms/vary.cc File Reference

```
#include "Cleanup.hh"
#include "Exceptions.hh"
#include "algorithms/vary.hh"
#include "algorithms/substitute.hh"
#include "algorithms/product_rule.hh"
#include "properties/Derivative.hh"
#include "properties/Accent.hh"
```

10.119 core/algorithms/vary.hh File Reference

```
#include "Algorithm.hh"
#include "algorithms/sort_product.hh"
```

Classes

- class cadabra::vary

  Generic variational derivative algorithm.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.120 core/algorithms/young_project.cc File Reference

```
#include "IndexIterator.hh"
#include "algorithms/young_project.hh"
```

10.121 core/algorithms/young_project.hh File Reference

```
#include "Algorithm.hh"
#include "YoungTab.hh"
```

Classes

- class cadabra::young_project

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.122 core/algorithms/young_project_product.cc File Reference

```
#include "Cleanup.hh"
#include "algorithms/young_project_product.hh"
#include "algorithms/young_project_tensor.hh"
#include "algorithms/distribute.hh"
#include "algorithms/collect_terms.hh"
#include "algorithms/canonicalise.hh"
```

10.123 core/algorithms/young_project_product.hh File Reference

```
#include "Algorithm.hh"
```

Classes

- class cadabra::young_project_product

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.124 core/algorithms/young_project_tensor.cc File Reference

```
#include "IndexIterator.hh"
#include "Exceptions.hh"
#include "Cleanup.hh"
#include "properties/Indices.hh"
#include "properties/Integer.hh"
#include "algorithms/young_project_tensor.hh"
#include "algorithms/collect_terms.hh"
#include "algorithms/indexsort.hh"
```

10.125 core/algorithms/young_project_tensor.hh File Reference

```
#include "Algorithm.hh"
#include "properties/TableauBase.hh"
```

Classes

- class cadabra::young_project_tensor

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.126 core/Bridge.cc File Reference

```
#include "PythonCdb.hh"
#include "Bridge.hh"
#include "algorithms/collect_terms.hh"
```

Functions

- void pull_in (std::shared_ptr< Ex > ex)

  Replace any objects of the form '@(...)' in the expression tree by the python expression '...' if it exists.

10.127 core/Bridge.hh File Reference

```
#include "Storage.hh"
```

Functions

- void pull_in (std::shared_ptr< cadabra::Ex > ex)

  Replace any objects of the form '@(...)' in the expression tree by the python expression '...' if it exists.

10.128 core/cadabra2 File Reference

10.129 core/cadabra2_defaults.py File Reference

Classes

- class cadabra2_defaults.Sympy
- class cadabra2_defaults.Server

Namespaces

- cadabra2_defaults

Functions

- def cadabra2_defaults.display (obj, delay_send=False)
- def cadabra2_defaults._displayhook (arg)
- def cadabra2_defaults.post_process (__cdbkernel__, ex)

Variables

- cadabra2_defaults.__cdbkernel__ = cadabra2.__cdbkernel__
- cadabra2_defaults.sympy
- string cadabra2_defaults.mopen = "\\begin{dmath*}{}"
- string cadabra2_defaults.mclose = "\\end{dmath*}"
- cadabra2_defaults.server
- bool cadabra2_defaults.have_matplotlib = True
- cadabra2_defaults.remember_display_hook = sys.displayhook
- cadabra2_defaults.displayhook

10.130 core/cadabra2python.cc File Reference

```
#include <iostream>
#include <CdbPython.hh>
#include <fstream>
```

Functions

- int main (int argc, char **argv)

10.130.1 Function Documentation

10.130.1.1 int main (int *argc*, char ** *argv*)

10.131 core/CdbPython.cc File Reference

```
#include "CdbPython.hh"
#include <boost/regex.hpp>
#include <boost/algorithm/string/replace.hpp>
```

10.132 core/CdbPython.hh File Reference

```
#include <string>
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- std::string cadabra::escape_quotes (const std::string &)
- std::string cadabra::cdb2python (const std::string &)
- std::string cadabra::convert_line (const std::string &, std::string &lhs, std::string &rhs, std::string &indent)

10.133 core/Cleanup.cc File Reference

```
#include "Cleanup.hh"
#include "Functional.hh"
#include "Algorithm.hh"
#include "algorithms/collect_terms.hh"
#include "properties/Diagonal.hh"
#include "properties/ExteriorDerivative.hh"
#include "properties/KroneckerDelta.hh"
#include "properties/NumericalFlat.hh"
#include "properties/PartialDerivative.hh"
#include "properties/ImaginaryI.hh"
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- void cadabra::cleanup_dispatch (const Kernel &k, Ex &, Ex::iterator &it)

  Central cleanup dispatch routine, which calls the other cleanup functions defined later.
- void cadabra::check_index_consistency (const Kernel &k, Ex &tr, Ex::iterator it)

  Generic tool to check for index inconsistencies independent of running any algorithm.
- bool cadabra::cleanup_productlike (const Kernel &k, Ex &, Ex::iterator &it)

  Individual node cleanup routines.
- bool cadabra::cleanup_sumlike (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::push_down_multiplier (const Kernel &k, Ex &tr, Ex::iterator it)

  Given a node with a non-unit multiplier, push this multiplier down the tree if the node is not allowed to have a non-unit multiplier.
- bool cadabra::cleanup_components (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_partialderivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_derivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_numericalflat (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_diagonal (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_kronecker (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_exterior_derivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- void cadabra::cleanup_dispatch_deep (const Kernel &k, Ex &, dispatcher_t disp=&cleanup_dispatch)

  More general cleanup of an entire tree.
- void cadabra::cleanup_dispatch_deep (const Kernel &k, Ex &tr, Ex::iterator &top, dispatcher_t dispatch)

10.134 core/Cleanup.hh File Reference

```
#include "Storage.hh"
#include "Kernel.hh"
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Typedefs

- typedef void(∗ cadabra::dispatcher_t) (const Kernel &k, Ex &, Ex::iterator &it)

Functions

- void cadabra::cleanup_dispatch (const Kernel &k, Ex &, Ex::iterator &it)

  Central cleanup dispatch routine, which calls the other cleanup functions defined later.
- void cadabra::cleanup_dispatch_deep (const Kernel &k, Ex &, dispatcher_t disp=&cleanup_dispatch)

  More general cleanup of an entire tree.
- void cadabra::cleanup_dispatch_deep (const Kernel &k, Ex &tr, Ex::iterator &top, dispatcher_t dispatch)
- bool cadabra::cleanup_productlike (const Kernel &k, Ex &, Ex::iterator &it)

  Individual node cleanup routines.
- bool cadabra::cleanup_sumlike (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_derivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_partialderivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_components (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_numericalflat (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_diagonal (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_kronecker (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::cleanup_exterior_derivative (const Kernel &k, Ex &tr, Ex::iterator &it)
- bool cadabra::push_down_multiplier (const Kernel &k, Ex &tr, Ex::iterator it)

  Given a node with a non-unit multiplier, push this multiplier down the tree if the node is not allowed to have a non-unit multiplier.
- void cadabra::check_index_consistency (const Kernel &k, Ex &tr, Ex::iterator it)

  Generic tool to check for index inconsistencies independent of running any algorithm.

10.135 core/CMakeLists.txt File Reference

Functions

- cmake_minimum_required (VERSION 2.8) project(Cadabra) set(CMAKE_MODULE_PATH $
- print (site.getsitepackages()[0])
- OUTPUT_VARIABLE PYTHON_SITE_PATH OUTPUT_STRIP_TRAILING_WHITESPACE endif () message("-- Python site path at $
- install (FILES"${PROJECT_BINARY_DIR}/cadabra2.so""${PROJECT_SOURCE_DIR}/cadabra2_↩ defaults.py"DESTINATION"${PYTHON_SITE_PATH}") configure_file("$

10.135.1 Function Documentation

10.135.1.1 cmake_minimum_required (VERSION 2. *8*)

10.135.1.2 OUTPUT_VARIABLE PYTHON_SITE_PATH OUTPUT_STRIP_TRAILING_WHITESPACE endif ( )

10.135.1.3 install (FILES"${PROJECT_BINARY_DIR}/cadabra2.so""${PROJECT_SOURCE_DIR}/cadabra2_defaults.py"DESTI↩ NATION"${PYTHON_SITE_PATH}")

10.135.1.4 print (site. *getsitepackages()[0]*)

10.136 client_server/CMakeLists.txt File Reference

Functions

- cmake_minimum_required (VERSION 2.8) project(CadabraServer) if(POLICY CMP0042) cmake_policy(SET CMP0042 NEW) endif(POLICY CMP0042)() set(CMAKE_MODULE_PATH $

10.136.1 Function Documentation

10.136.1.1 cmake_minimum_required (VERSION 2. *8*)

10.137 frontend/common/CMakeLists.txt File Reference

Functions

- cmake_minimum_required (VERSION 2.8) project(TeXEngine) if(POLICY CMP0042) cmake_policy(SET C↩ MP0042 NEW) endif(POLICY CMP0042) if(CMAKE_COMPILER_IS_GNUCC OR CMAKE_COMPILER_I↩ S_GNUCXX) set(CMAKE_CXX_FLAGS"$

10.137.1 Function Documentation

10.137.1.1 cmake_minimum_required (VERSION 2. *8*)

10.138 frontend/gtkmm/CMakeLists.txt File Reference

10.139 core/Combinatorics.cc File Reference

```
#include "Combinatorics.hh"
```

10.140 core/Combinatorics.hh File Reference

```
#include <vector>
#include <cassert>
#include <algorithm>
#include <iomanip>
#include <iostream>
#include <map>
```

Classes

- class combin::combinations_base< T >
- class combin::combinations< T >
- class combin::symmetriser< T >
- class combin::symm_helper< T >
- class combin::symm_val_helper< T >
- class combin::symmetriser< T >

Namespaces

- combin

Typedefs

- typedef std::vector< unsigned int > combin::range_t
- typedef std::vector< range_t > combin::range_vector_t
- typedef std::vector< int > combin::weights_t

Functions

- unsigned long combin::factorial (unsigned int x)
- long combin::vector_sum (const std::vector< int > &)

  sum of elements
- unsigned long combin::vector_prod (const std::vector< unsigned int > &)

  product of elements
- unsigned long combin::vector_prod_fact (const std::vector< unsigned int > &)

  product of factorials of elements
- bool combin::operator== (const std::vector< unsigned int > &, const std::vector< unsigned int > &)
- long combin::hash (const std::vector< unsigned int > &)

  compute a hash value for a vector of unsigned ints
- int combin::determine_intersection_ranges (const range_vector_t &prod, const range_vector_t &indv, range_vector_t &target)
- template<class iterator1 , class iterator2 >
  int combin::ordersign (iterator1 b1, iterator1 e1, iterator2 b2, iterator2 e2, int stepsize=1)
- template<class iterator1 >
  int combin::ordersign (iterator1 b1, iterator1 e1)
- template<class T >
  T combin::fact (T x)
- template<class T >
  std::ostream & combin::operator<< (std::ostream &str, const symmetriser< T > &sym)

10.141 core/Compare.cc File Reference

```
#include <typeinfo>
#include "Compare.hh"
#include "Algorithm.hh"
#include <sstream>
#include "pcrecpp.h"
#include "properties/Indices.hh"
#include "properties/Coordinate.hh"
#include "properties/ImplicitIndex.hh"
#include "properties/SelfCommutingBehaviour.hh"
#include "properties/CommutingAsSum.hh"
#include "properties/CommutingAsProduct.hh"
#include "properties/CommutingBehaviour.hh"
#include "properties/DifferentialForm.hh"
#include "properties/DiracBar.hh"
#include "properties/Integer.hh"
#include "properties/SortOrder.hh"
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Macros

- #define DEBUG(ln)

Functions

- int cadabra::subtree_compare (const Properties ∗, Ex::iterator one, Ex::iterator two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2, bool literal_wildcards=false)

  Basic building block subtree comparison function for tensors without dummy indices, which determines whether two tensor subtrees are equal up to the names of indices.
- bool cadabra::tree_less (const Properties ∗, const Ex &one, const Ex &two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2)

  Various comparison functions, some exact, some with pattern logic.
- bool cadabra::tree_equal (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier)
- bool cadabra::tree_exact_less (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool cadabra::tree_exact_equal (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool cadabra::subtree_less (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier)
- bool cadabra::subtree_equal (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier)
- bool cadabra::subtree_exact_less (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_↩ prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool cadabra::subtree_exact_equal (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod↩ _prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool operator< (const cadabra::Ex::iterator &i1, const cadabra::Ex::iterator &i2)
- bool operator< (const cadabra::Ex &e1, const cadabra::Ex &e2)
- std::ostream & operator<< (std::ostream &s, cadabra::Ex_comparator::useprops_t up)

10.141.1 Macro Definition Documentation

10.141.1.1 #define DEBUG(*ln*)

10.141.2 Function Documentation

10.141.2.1 bool operator< (const cadabra::Ex::iterator & *i1*, const cadabra::Ex::iterator & *i2*)

10.141.2.2 bool operator< (const **cadabra::Ex** & *e1*, const **cadabra::Ex** & *e2*)

10.141.2.3 std::ostream& operator<< (std::ostream & *s*, **cadabra::Ex_comparator::useprops_t** *up*)

10.142 core/Compare.hh File Reference

```
#include "Storage.hh"
#include "Props.hh"
#include "properties/Indices.hh"
```

Classes

- class cadabra::tree_less_obj

  Compare two trees by pattern logic, i.e. modulo index names.
- class cadabra::tree_less_modprel_obj
- class cadabra::tree_equal_obj
- class cadabra::tree_exact_less_obj

  Compare two trees exactly, i.e. including exact index names.
- class cadabra::tree_exact_less_mod_prel_obj
- class cadabra::tree_exact_equal_obj
- class cadabra::tree_exact_equal_mod_prel_obj
- class cadabra::tree_exact_less_for_indexmap_obj

  Compare for indexmap_t.
- class cadabra::tree_exact_less_no_wildcards_obj

  Compare two trees exactly, treat wildcard names as ordinary names.
- class cadabra::tree_exact_less_no_wildcards_mod_prel_obj
- class cadabra::Ex_comparator

  A generic tree comparison class which will take into account index contractions and will also keep track of a replacement list for all types of cadabra wildcards.
- class cadabra::Ex_is_equivalent

  Basic comparison operator for tree iterators, so we can use them as keys in maps.
- class cadabra::Ex_is_less

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- int cadabra::subtree_compare (const Properties ∗, Ex::iterator one, Ex::iterator two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2, bool literal_wildcards=false)

  Basic building block subtree comparison function for tensors without dummy indices, which determines whether two tensor subtrees are equal up to the names of indices.
- bool cadabra::tree_less (const Properties ∗, const Ex &one, const Ex &two, int mod_prel=-2, bool checksets=true, int compare_multiplier=-2)

  Various comparison functions, some exact, some with pattern logic.
- bool cadabra::tree_equal (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier)
- bool cadabra::tree_exact_less (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool cadabra::tree_exact_equal (const Properties ∗properties, const Ex &one, const Ex &two, int mod_prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool cadabra::subtree_less (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier)
- bool cadabra::subtree_equal (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_prel, bool checksets, int compare_multiplier)
- bool cadabra::subtree_exact_less (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod_↩ prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool cadabra::subtree_exact_equal (const Properties ∗properties, Ex::iterator one, Ex::iterator two, int mod↩ _prel, bool checksets, int compare_multiplier, bool literal_wildcards)
- bool operator< (const cadabra::Ex::iterator &, const cadabra::Ex::iterator &)
- bool operator< (const cadabra::Ex &, const cadabra::Ex &)
- std::ostream & operator<< (std::ostream &, cadabra::Ex_comparator::useprops_t up)

10.142.1 Function Documentation

10.142.1.1 bool operator< (const cadabra::Ex::iterator & , const cadabra::Ex::iterator &)

10.142.1.2 bool operator< (const cadabra::Ex & , const cadabra::Ex &)

10.142.1.3 std::ostream& operator<< (std::ostream & , cadabra::Ex_comparator::useprops_t *up*)

10.143 core/DisplayBase.cc File Reference

```
#include "DisplayBase.hh"
```

10.144 core/DisplayBase.hh File Reference

```
#include "Props.hh"
#include "Storage.hh"
```

Classes

- class cadabra::DisplayBase

  Base class for all display classes.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.145 core/DisplaySympy.cc File Reference

```
#include "Algorithm.hh"
#include "Functional.hh"
#include "DisplaySympy.hh"
#include "properties/Depends.hh"
#include "properties/Accent.hh"
```

10.146 core/DisplaySympy.hh File Reference

```
#include "DisplayBase.hh"
#include <ostream>
#include <map>
#include <set>
```

Classes

- class cadabra::DisplaySympy

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Typedefs

- typedef uint32_t kunichar

Functions

- const char * cadabra::unichar (kunichar c)

10.146.1 Typedef Documentation

10.146.1.1 typedef uint32_t kunichar

10.147 core/DisplayTerminal.cc File Reference

```
#include "Algorithm.hh"
#include "DisplayTerminal.hh"
#include "properties/Accent.hh"
```

10.148 core/DisplayTerminal.hh File Reference

```
#include "Props.hh"
#include "Storage.hh"
#include "DisplayBase.hh"
#include <ostream>
#include <map>
```

Classes

- class cadabra::DisplayTerminal

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Typedefs

- typedef uint32_t kunichar

Functions

- const char * cadabra::unichar (kunichar c)

10.148.1 Typedef Documentation

10.148.1.1 typedef uint32_t kunichar

10.149 core/DisplayTeX.cc File Reference

```
#include "Symbols.hh"
#include "DisplayTeX.hh"
#include "Algorithm.hh"
#include "properties/LaTeXForm.hh"
#include "properties/Derivative.hh"
#include "properties/Accent.hh"
```

Macros

- #define nbsp " "
- #define zwnbsp ""

Functions

- int k_unichar_to_utf8 (kunichar c, char *buf)
- const char * unichar (kunichar c)

10.149.1 Macro Definition Documentation

10.149.1.1 #define nbsp " "

10.149.1.2 #define zwnbsp ""

10.149.2 Function Documentation

10.149.2.1 int k_unichar_to_utf8 (kunichar *c*, char * *buf*)

10.149.2.2 const char* unichar (kunichar *c*)

10.150 core/DisplayTeX.hh File Reference

```
#include "Props.hh"
#include "Storage.hh"
#include "DisplayBase.hh"
#include <ostream>
```

Classes

- class cadabra::DisplayTeX

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Typedefs

- typedef uint32_t kunichar

Functions

- const char * cadabra::unichar (kunichar c)

10.150.1 Typedef Documentation

10.150.1.1 typedef uint32_t kunichar

10.151 core/Dummies.hh File Reference

```
#include "algorithm.hh"
#include "props.hh"
```

Classes

- class rename_dummies

10.152 core/echokernel.py File Reference

Classes

- class echokernel.EchoKernel

Namespaces

- echokernel

Variables

- echokernel.kernel_class

10.153 core/Exceptions.cc File Reference

```
#include "Exceptions.hh"
#include <iostream>
```

10.154 core/Exceptions.hh File Reference

```
#include <stdexcept>
#include <string>
```

Classes

- class CadabraException
- class ConsistencyException
- class ParseException
- class InterruptionException
- class ArgumentException

  Exception thrown when arguments to an algorithm or property are not correct.
- class NonScalarException

  Exception thrown when something requires that an expression is a pure scalar (i.e.
- class RuntimeException

  Exception thrown when an algorithm cannot complete, e.g.
- class InternalError

  Exception thrown when there the Cadabra code detects an internal inconsistency; this should never happen, but is better than bailing out with an assert.
- class NotYetImplemented

  Exception thrown when there the Cadabra code detects an internal inconsistency; this should never happen, but is better than bailing out with an assert.

10.155 core/Exchange.cc File Reference

```
#include <map>
#include "Exchange.hh"
#include "Compare.hh"
#include "Algorithm.hh"
#include "properties/DiracBar.hh"
```

Functions

- bool operator< (const exchange::tensor_type_t &one, const exchange::tensor_type_t &two)

10.155.1 Function Documentation

10.155.1.1 bool operator< (const exchange::tensor_type_t & *one*, const exchange::tensor_type_t & *two*)

10.156 core/Exchange.hh File Reference

```
#include <vector>
#include "Storage.hh"
#include "properties/GammaMatrix.hh"
#include "properties/Spinor.hh"
#include "properties/Traceless.hh"
#include "properties/GammaTraceless.hh"
#include "properties/SelfCommutingBehaviour.hh"
#include "properties/CommutingBehaviour.hh"
```

Classes

- class cadabra::exchange
- struct cadabra::exchange::identical_tensors_t
- struct cadabra::exchange::tensor_type_t

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- bool cadabra::operator< (const exchange::tensor_type_t &one, const exchange::tensor_type_t &two)

10.157 core/Functional.cc File Reference

```
#include "Functional.hh"
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- void cadabra::do_list (const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f)

  Apply a function on every element of a list, or if the iterator 'it' does not point to a list, only on that single element.
- int cadabra::list_size (const Ex &tr, Ex::iterator it)

  For lists as defined above for 'do_list', return their size (in case you really need to know the size before iterating over the elements).
- Ex::iterator cadabra::do_subtree (const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f)

  Apply a function on every node in the tree at and below the given node, depth-first.
- Ex::iterator cadabra::find_in_subtree (const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f, bool including_head=true)

  Returns an iterator to the first element for which 'f' returns 'true', or 'tr.end()'.
- Ex::iterator cadabra::find_in_list (const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f)

  Returns an iterator to the first element for which 'f' does not return tr.end().
- Ex cadabra::make_list (Ex el)

  Ensure that the tree is a list, even if it contains only a single element.

10.158 core/Functional.hh File Reference

```
#include <functional>
#include "Storage.hh"
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- void cadabra::do_list (const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f)

  Apply a function on every element of a list, or if the iterator 'it' does not point to a list, only on that single element.
- int cadabra::list_size (const Ex &tr, Ex::iterator it)

  For lists as defined above for 'do_list', return their size (in case you really need to know the size before iterating over the elements).
- Ex::iterator cadabra::do_subtree (const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f)

  Apply a function on every node in the tree at and below the given node, depth-first.
- Ex::iterator cadabra::find_in_list (const Ex &tr, Ex::iterator it, std::function< Ex::iterator(Ex::iterator)> f)

  Returns an iterator to the first element for which 'f' does not return tr.end().
- Ex::iterator cadabra::find_in_subtree (const Ex &tr, Ex::iterator it, std::function< bool(Ex::iterator)> f, bool including_head=true)

  Returns an iterator to the first element for which 'f' returns 'true', or 'tr.end()'.
- Ex cadabra::make_list (Ex el)

  Ensure that the tree is a list, even if it contains only a single element.

10.159 core/IndexIterator.cc File Reference

```
#include "IndexIterator.hh"
#include "properties/IndexInherit.hh"
```

10.160 core/IndexIterator.hh File Reference

```
#include "Storage.hh"
#include "Props.hh"
```

Classes

- class cadabra::index_iterator

  An iterator which iterates over indices even if they are at lower levels, i.e.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.161 core/Kernel.cc File Reference

```
#include "Kernel.hh"
#include "PreClean.hh"
#include "Cleanup.hh"
#include "Parser.hh"
```

10.162 core/Kernel.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::Kernel

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.163 core/Parser.cc File Reference

```
#include "Parser.hh"
#include "PreProcessor.hh"
#include <sstream>
```

Functions

- std::istream & operator>> (std::istream &str, cadabra::Parser &pa)

10.163.1 Function Documentation

10.163.1.1 std::istream& operator>> (std::istream & *str,* cadabra::Parser & *pa*)

10.164 core/Parser.hh File Reference

```
#include <string>
#include <vector>
#include <iostream>
#include "Storage.hh"
```

Classes

- class cadabra::Parser

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- std::istream & operator>> (std::istream &, cadabra::Parser &)

10.164.1 Function Documentation

10.164.1.1 std::istream& operator>> (std::istream & , cadabra::Parser &)

10.165 core/passing.cc File Reference

```
#include <iostream>
```

Functions

- void fun (int *&p)
- int main ()

Variables

- int k =5

10.165.1 Function Documentation

10.165.1.1 void fun (int *& *p*)

10.165.1.2 int main ( )

10.165.2 Variable Documentation

10.165.2.1 int k =5

10.166 core/Permutations.hh File Reference

```
#include <stdexcept>
#include <vector>
```

Classes

- class PermutationException

  Generic permutation group material.
- class Perm
- class Images

10.167 core/PreClean.cc File Reference

```
#include "PreClean.hh"
#include "Cleanup.hh"
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- void cadabra::pre_clean_dispatch (const Kernel &kernel, Ex &ex, Ex::iterator &it)
- void cadabra::pre_clean_dispatch_deep (const Kernel &k, Ex &tr)
- void cadabra::cleanup_updown (const Kernel &k, Ex &, Ex::iterator &it)

  Cleanup for individual node types.
- void cadabra::cleanup_rational (const Kernel &k, Ex &tr, Ex::iterator &st)
- void cadabra::cleanup_frac (const Kernel &k, Ex &tr, Ex::iterator &st)
- void cadabra::cleanup_sub (const Kernel &k, Ex &tr, Ex::iterator &it)
- void cadabra::cleanup_indexbracket (const Kernel &k, Ex &, Ex::iterator &it)

  Convert parser output which indicates an indexbracket to an actual indexbracket node.

10.168 core/PreClean.hh File Reference

```
#include "Algorithm.hh"
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- void cadabra::pre_clean_dispatch (const Kernel &kernel, Ex &ex, Ex::iterator &it)
- void cadabra::pre_clean_dispatch_deep (const Kernel &k, Ex &tr)
- void cadabra::cleanup_updown (const Kernel &k, Ex &, Ex::iterator &it)

  Cleanup for individual node types.
- void cadabra::cleanup_rational (const Kernel &k, Ex &tr, Ex::iterator &st)
- void cadabra::cleanup_frac (const Kernel &k, Ex &tr, Ex::iterator &st)
- void cadabra::cleanup_sub (const Kernel &k, Ex &tr, Ex::iterator &it)
- void cadabra::cleanup_indexbracket (const Kernel &k, Ex &, Ex::iterator &it)

  Convert parser output which indicates an indexbracket to an actual indexbracket node.

10.169 core/PreProcessor.cc File Reference

```
#include <ctype.h>
#include <stdexcept>
#include <sstream>
#include "PreProcessor.hh"
```

Functions

- std::istream & operator>> (std::istream &s, preprocessor &p)
- std::ostream & operator<< (std::ostream &s, const preprocessor &p)

10.169.1 Function Documentation

10.169.1.1 std::ostream& operator<< (std::ostream & *s*, const preprocessor & *p*)

10.169.1.2 std::istream& operator>> (std::istream & *s*, preprocessor & *p*)

10.170 core/PreProcessor.hh File Reference

```
#include <iostream>
#include <string>
#include <vector>
```

Classes

- class preprocessor

  Preprocessing class which takes infix mathematical notation with all sorts of maths shortcuts and transforms it into a string which is properly formatted in prefix notation.
- class preprocessor::accu_t

Enumerations

- enum {
  tok_arrow =0xac, tok_unequals =0xad, tok_wedge =0xae, tok_pow =0xaf,
  tok_set_option =0xb0, tok_declare =0xb1, tok_sequence =0xb2, tok_siblings =0xb3 }

Functions

- std::ostream & operator<< (std::ostream &, const preprocessor &)
- std::istream & operator>> (std::istream &, preprocessor &)

10.170.1 Enumeration Type Documentation

10.170.1.1 anonymous enum

Enumerator

tok_arrow

tok_unequals

tok_wedge

tok_pow

tok_set_option

tok_declare

tok_sequence

tok_siblings

10.170.2 Function Documentation

10.170.2.1 std::ostream& operator<< (std::ostream & , const preprocessor &)

10.170.2.2 std::istream& operator>> (std::istream & , preprocessor &)

10.171 core/ProgressMonitor.cc File Reference

```
#include "ProgressMonitor.hh"
#include <iostream>
#include <sstream>
```

10.172 core/ProgressMonitor.hh File Reference

```
#include <string>
#include <chrono>
#include <stack>
#include <map>
#include <vector>
```

Classes

- class ProgressMonitor
- class ProgressMonitor::Total
- class ProgressMonitor::Block

10.173 core/properties/Accent.cc File Reference

```
#include "properties/Accent.hh"
```

10.174 core/properties/Accent.hh File Reference

```
#include "properties/IndexInherit.hh"
#include "properties/NumericalFlat.hh"
```

Classes

- class cadabra::Accent

  Turns a symbol into an accent.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.175 core/properties/AntiCommuting.cc File Reference

```
#include "properties/AntiCommuting.hh"
```

10.176 core/properties/AntiCommuting.hh File Reference

```
#include "properties/CommutingBehaviour.hh"
```

Classes

- class cadabra::AntiCommuting

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.177 core/properties/AntiSymmetric.cc File Reference

```
#include "Algorithm.hh"
#include "properties/AntiSymmetric.hh"
```

10.178 core/properties/AntiSymmetric.hh File Reference

```
#include "properties/TableauBase.hh"
#include "properties/Traceless.hh"
```

Classes

- class cadabra::AntiSymmetric

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.179 core/properties/Commuting.cc File Reference

```
#include "properties/Commuting.hh"
```

10.180 core/properties/Commuting.hh File Reference

```
#include "properties/CommutingBehaviour.hh"
```

Classes

- class cadabra::Commuting

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.181 core/properties/CommutingAsProduct.cc File Reference

```
#include "properties/CommutingAsProduct.hh"
```

10.182 core/properties/CommutingAsProduct.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::CommutingAsProduct

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.183 core/properties/CommutingAsSum.cc File Reference

```
#include "properties/CommutingAsSum.hh"
```

10.184 core/properties/CommutingAsSum.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::CommutingAsSum

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.185 core/properties/CommutingBehaviour.cc File Reference

```
#include "properties/CommutingBehaviour.hh"
```

10.186 core/properties/CommutingBehaviour.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::CommutingBehaviour

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.187 core/properties/Coordinate.cc File Reference

```
#include "properties/Coordinate.hh"
```

10.188 core/properties/Coordinate.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::Coordinate

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.189 core/properties/DAntiSymmetric.cc File Reference

```
#include "properties/DAntiSymmetric.hh"
```

10.190 core/properties/DAntiSymmetric.hh File Reference

```
#include "properties/TableauBase.hh"
```

Classes

- class cadabra::DAntiSymmetric

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.191 core/properties/Depends.cc File Reference

```
#include "properties/Depends.hh"
#include "properties/Indices.hh"
#include "properties/Coordinate.hh"
#include "properties/Derivative.hh"
#include "properties/Accent.hh"
#include "Exceptions.hh"
#include "Kernel.hh"
```

10.192 core/properties/Depends.hh File Reference

```
#include "properties/DependsBase.hh"
```

Classes

- class cadabra::Depends

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.193 core/properties/DependsBase.hh File Reference

```
#include "Props.hh"
#include "Kernel.hh"
```

Classes

- class cadabra::DependsBase

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.194 core/properties/DependsInherit.cc File Reference

```
#include "properties/DependsInherit.hh"
```

10.195 core/properties/DependsInherit.hh File Reference

```
#include "properties/DependsBase.hh"
```

Classes

- class cadabra::DependsInherit

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.196 core/properties/Derivative.cc File Reference

```
#include "properties/Derivative.hh"
#include "Props.hh"
#include "Kernel.hh"
```

10.197 core/properties/Derivative.hh File Reference

```
#include "properties/CommutingAsProduct.hh"
#include "properties/NumericalFlat.hh"
#include "properties/WeightBase.hh"
#include "properties/TableauBase.hh"
#include "properties/Distributable.hh"
#include "properties/DependsInherit.hh"
#include "properties/IndexInherit.hh"
#include "properties/SortOrder.hh"
```

Classes

- class cadabra::Derivative

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.198 core/properties/Diagonal.cc File Reference

```
#include "properties/Diagonal.hh"
```

10.199 core/properties/Diagonal.hh File Reference

```
#include "Props.hh"
#include "properties/Symmetric.hh"
```

Classes

- class cadabra::Diagonal

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.200 core/properties/DifferentialForm.cc File Reference

```
#include "DifferentialForm.hh"
```

10.201 core/properties/DifferentialForm.hh File Reference

```
#include "Props.hh"
#include "properties/ImplicitIndex.hh"
#include "properties/IndexInherit.hh"
#include "properties/DifferentialFormBase.hh"
```

Classes

- class cadabra::DifferentialForm

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.202 core/properties/DifferentialFormBase.hh File Reference

```
#include "Props.hh"
#include "properties/ImplicitIndex.hh"
#include "properties/IndexInherit.hh"
```

Classes

- class cadabra::DifferentialFormBase

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.203 core/properties/DiracBar.cc File Reference

```
#include "DiracBar.hh"
```

10.204 core/properties/DiracBar.hh File Reference

```
#include "properties/Accent.hh"
#include "properties/Distributable.hh"
```

Classes

- class cadabra::DiracBar

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.205 core/properties/Distributable.cc File Reference

```
#include "Distributable.hh"
```

10.206 core/properties/Distributable.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::Distributable

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.207 core/properties/EpsilonTensor.cc File Reference

```
#include "properties/EpsilonTensor.hh"
```

10.208 core/properties/EpsilonTensor.hh File Reference

```
#include "properties/AntiSymmetric.hh"
```

Classes

- class cadabra::EpsilonTensor

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.209 core/properties/ExteriorDerivative.cc File Reference

```
#include "Exceptions.hh"
#include "ExteriorDerivative.hh"
```

10.210 core/properties/ExteriorDerivative.hh File Reference

```
#include "Props.hh"
#include "properties/Derivative.hh"
#include "properties/DifferentialFormBase.hh"
```

Classes

- class cadabra::ExteriorDerivative

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.211 core/properties/FilledTableau.cc File Reference

```
#include "properties/FilledTableau.hh"
```

10.212 core/properties/FilledTableau.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::FilledTableau

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.213 core/properties/GammaMatrix.cc File Reference

```
#include "GammaMatrix.hh"
#include "Kernel.hh"
```

10.214 core/properties/GammaMatrix.hh File Reference

```
#include "properties/Matrix.hh"
#include "properties/AntiSymmetric.hh"
```

Classes

- class cadabra::GammaMatrix

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.215 core/properties/GammaTraceless.cc File Reference

```
#include "GammaTraceless.hh"
```

10.216 core/properties/GammaTraceless.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::GammaTraceless

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.217 core/properties/ImaginaryI.cc File Reference

```
#include "properties/ImaginaryI.hh"
```

10.218 core/properties/ImaginaryI.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::ImaginaryI

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.219 core/properties/ImplicitIndex.cc File Reference

```
#include "properties/ImplicitIndex.hh"
#include "Exceptions.hh"
```

10.220 core/properties/ImplicitIndex.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::ImplicitIndex

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.221 core/properties/IndexInherit.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::IndexInherit

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.222 core/properties/Indices.cc File Reference

```
#include "properties/Indices.hh"
#include "Exceptions.hh"
#include "Functional.hh"
```

10.223 core/properties/Indices.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::Indices

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.224 core/properties/Integer.cc File Reference

```
#include "properties/Integer.hh"
#include "Cleanup.hh"
#include "Kernel.hh"
#include "algorithms/collect_terms.hh"
```

10.225 core/properties/Integer.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::Integer

  Property indicating that a symbolic object always takes integer values.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.226 core/properties/Integral.hh File Reference

```
#include "Props.hh"
```

Classes

- class Integral

10.227 core/properties/InverseMetric.cc File Reference

```
#include "Exceptions.hh"
#include "properties/InverseMetric.hh"
```

10.228 core/properties/InverseMetric.hh File Reference

```
#include "Props.hh"
#include "properties/TableauSymmetry.hh"
```

Classes

- class cadabra::InverseMetric

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.229 core/properties/KroneckerDelta.cc File Reference

```
#include "properties/KroneckerDelta.hh"
#include "Exceptions.hh"
```

10.230 core/properties/KroneckerDelta.hh File Reference

```
#include "properties/TableauBase.hh"
```

Classes

- class cadabra::KroneckerDelta

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.231 core/properties/LaTeXForm.cc File Reference

```
#include "properties/LaTeXForm.hh"
```

10.232 core/properties/LaTeXForm.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::LaTeXForm

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.233 core/properties/Matrix.cc File Reference

```
#include "Matrix.hh"
```

10.234 core/properties/Matrix.hh File Reference

```
#include "properties/ImplicitIndex.hh"
```

Classes

- class cadabra::Matrix

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.235 core/properties/Metric.cc File Reference

```
#include "Exceptions.hh"
#include "properties/Metric.hh"
```

10.236 core/properties/Metric.hh File Reference

```
#include "Props.hh"
#include "properties/Symmetric.hh"
```

Classes

- class cadabra::Metric

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.237 core/properties/NonCommuting.cc File Reference

```
#include "properties/NonCommuting.hh"
```

10.238 core/properties/NonCommuting.hh File Reference

```
#include "properties/CommutingBehaviour.hh"
```

Classes

- class cadabra::NonCommuting

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.239 core/properties/NumericalFlat.cc File Reference

```
#include "properties/NumericalFlat.hh"
```

10.240 core/properties/NumericalFlat.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::NumericalFlat

  Property indicating that an operator is numerically flat, so that numerical factors in the argument can be taken outside.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.241 core/properties/PartialDerivative.cc File Reference

```
#include "Algorithm.hh"
#include "properties/PartialDerivative.hh"
```

10.242 core/properties/PartialDerivative.hh File Reference

```
#include "properties/Derivative.hh"
```

Classes

- class cadabra::PartialDerivative

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.243 core/properties/RiemannTensor.cc File Reference

```
#include "IndexIterator.hh"
#include "Exceptions.hh"
#include "Kernel.hh"
#include "Algorithm.hh"
#include "properties/RiemannTensor.hh"
```

10.244 core/properties/RiemannTensor.hh File Reference

```
#include "properties/TableauSymmetry.hh"
```

Classes

- class cadabra::RiemannTensor

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.245 core/properties/SatisfiesBianchi.cc File Reference

```
#include "properties/SatisfiesBianchi.hh"
#include "IndexIterator.hh"
```

10.246 core/properties/SatisfiesBianchi.hh File Reference

```
#include "properties/TableauBase.hh"
```

Classes

- class cadabra::SatisfiesBianchi

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.247 core/properties/SelfAntiCommuting.cc File Reference

```
#include "properties/SelfAntiCommuting.hh"
```

10.248 core/properties/SelfAntiCommuting.hh File Reference

```
#include "properties/SelfCommutingBehaviour.hh"
```

Classes

- class cadabra::SelfAntiCommuting

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.249 core/properties/SelfCommuting.cc File Reference

```
#include "properties/SelfCommuting.hh"
```

10.250 core/properties/SelfCommuting.hh File Reference

```
#include "properties/SelfCommutingBehaviour.hh"
```

Classes

- class cadabra::SelfCommuting

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.251 core/properties/SelfCommutingBehaviour.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::SelfCommutingBehaviour

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.252 core/properties/SelfNonCommuting.cc File Reference

```
#include "properties/SelfNonCommuting.hh"
```

10.253 core/properties/SelfNonCommuting.hh File Reference

```
#include "properties/SelfCommutingBehaviour.hh"
```

Classes

- class cadabra::SelfNonCommuting

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.254 core/properties/SortOrder.cc File Reference

```
#include "properties/SortOrder.hh"
```

10.255 core/properties/SortOrder.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::SortOrder

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.256 core/properties/Spinor.cc File Reference

```
#include "Spinor.hh"
#include "Exceptions.hh"
#include "Kernel.hh"
```

10.257 core/properties/Spinor.hh File Reference

```
#include "properties/ImplicitIndex.hh"
```

Classes

- class cadabra::Spinor

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.258 core/properties/Symbol.cc File Reference

```
#include "properties/Symbol.hh"
```

10.259 core/properties/Symbol.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::Symbol

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.260 core/properties/Symmetric.cc File Reference

```
#include "properties/Symmetric.hh"
```

10.261 core/properties/Symmetric.hh File Reference

```
#include "properties/TableauBase.hh"
```

Classes

- class cadabra::Symmetric

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.262 core/properties/Tableau.cc File Reference

```
#include "properties/Tableau.hh"
```

10.263 core/properties/Tableau.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::Tableau

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.264 core/properties/TableauBase.cc File Reference

```
#include "properties/TableauBase.hh"
```

10.265 core/properties/TableauBase.hh File Reference

```
#include "Props.hh"
#include "YoungTab.hh"
```

Classes

- class cadabra::TableauBase

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.266 core/properties/TableauSymmetry.cc File Reference

```
#include "properties/TableauSymmetry.hh"
#include "IndexIterator.hh"
#include "Kernel.hh"
```

10.267 core/properties/TableauSymmetry.hh File Reference

```
#include "properties/TableauBase.hh"
#include <vector>
```

Classes

- class cadabra::TableauSymmetry

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.268 core/properties/Traceless.cc File Reference

```
#include "Traceless.hh"
```

10.269 core/properties/Traceless.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::Traceless

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.270 core/properties/Weight.cc File Reference

```
#include "Weight.hh"
```

10.271 core/properties/Weight.hh File Reference

```
#include "properties/WeightBase.hh"
```

Classes

- class cadabra::Weight

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.272 core/properties/WeightBase.hh File Reference

```
#include "Props.hh"
```

Classes

- class cadabra::WeightBase

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.273 core/properties/WeightInherit.cc File Reference

```
#include "WeightInherit.hh"
#include "Kernel.hh"
```

10.274 core/properties/WeightInherit.hh File Reference

```
#include "Exceptions.hh"
#include "properties/WeightBase.hh"
```

Classes

- class cadabra::WeightInherit
- class cadabra::WeightInherit::WeightException

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.275 core/properties/WeylTensor.cc File Reference

```
#include "IndexIterator.hh"
#include "Exceptions.hh"
#include "Kernel.hh"
#include "Algorithm.hh"
#include "properties/WeylTensor.hh"
```

10.276 core/properties/WeylTensor.hh File Reference

```
#include "properties/TableauSymmetry.hh"
#include "properties/Traceless.hh"
```

Classes

- class cadabra::WeylTensor

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.277 core/Props.cc File Reference

```
#include "Props.hh"
#include "Exceptions.hh"
#include "Compare.hh"
#include <stdlib.h>
#include <typeinfo>
#include <iostream>
#include <sstream>
#include "properties/Indices.hh"
```

10.278 core/Props.hh File Reference

```
#include <map>
#include <list>
#include "Storage.hh"
```

Classes

- class cadabra::pattern
- class cadabra::keyval_t

  Arguments to properties get parsed into a keyval_t structure.
- class cadabra::property

  Base class for all properties, handling argument parsing and defining the interface.
- class cadabra::labelled_property
- class cadabra::list_property

  Something cannot be both a list property and a normal property at the same time, so we can safely inherit without virtual.
- class cadabra::Inherit< T >

  If a property X derives from Inherit<Y>, and get<Y> is called on an object which has an X property (but no Y property), then the get<Y> will look at the non-index child of the object to see if that has a Y property.
- class cadabra::PropertyInherit

  PropertyInherit is like Inherit<T> for all properties.
- class cadabra::Properties

  Class holding a collection of properties attached to expressions.
- class cadabra::Properties::registered_property_map_t

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.279 core/ptest.cc File Reference

```
#include <boost/python.hpp>
```

Classes

- struct X

  Mockup types.
- struct Y
- struct Z

Functions

- X * make_x_and_inject_y ()

  Auxiliary function that will create X and inject an Y object as 'y' into the caller's frame.
- void fun ()
- BOOST_PYTHON_MODULE (example)

10.279.1 Function Documentation

10.279.1.1 BOOST_PYTHON_MODULE (example)

10.279.1.2 void fun ( )

10.279.1.3 X* make_x_and_inject_y ( )

Auxiliary function that will create X and inject an Y object as 'y' into the caller's frame.

10.280 core/PythonCdb.cc File Reference

```
#include <memory>
#include "PythonCdb.hh"
#include "SympyCdb.hh"
#include "Parser.hh"
#include "Bridge.hh"
#include "Exceptions.hh"
#include "DisplayTeX.hh"
#include "DisplaySympy.hh"
#include "DisplayTerminal.hh"
#include "Cleanup.hh"
#include "PreClean.hh"
#include "PythonException.hh"
#include "ProgressMonitor.hh"
#include <boost/python/implicit.hpp>
#include <boost/parameter/preprocessor.hpp>
#include <boost/mpl/vector.hpp>
#include <boost/python/enum.hpp>
#include <boost/python/def.hpp>
#include <boost/python/module.hpp>
#include <boost/python/stl_iterator.hpp>
#include <boost/python/slice.hpp>
#include <boost/python/suite/indexing/vector_indexing_suite.hpp>
#include <boost/algorithm/string/replace.hpp>
#include <sstream>
#include "properties/Accent.hh"
#include "properties/AntiCommuting.hh"
#include "properties/AntiSymmetric.hh"
#include "properties/Commuting.hh"
#include "properties/Coordinate.hh"
#include "properties/Depends.hh"
#include "properties/DependsInherit.hh"
#include "properties/Derivative.hh"
#include "properties/DifferentialForm.hh"
#include "properties/DiracBar.hh"
#include "properties/GammaMatrix.hh"
#include "properties/CommutingAsProduct.hh"
#include "properties/CommutingAsSum.hh"
#include "properties/DAntiSymmetric.hh"
#include "properties/Diagonal.hh"
#include "properties/Distributable.hh"
#include "properties/EpsilonTensor.hh"
#include "properties/ExteriorDerivative.hh"
#include "properties/FilledTableau.hh"
#include "properties/ImaginaryI.hh"
#include "properties/ImplicitIndex.hh"
#include "properties/Indices.hh"
#include "properties/IndexInherit.hh"
#include "properties/Integer.hh"
#include "properties/InverseMetric.hh"
#include "properties/KroneckerDelta.hh"
#include "properties/LaTeXForm.hh"
#include "properties/Matrix.hh"
#include "properties/Metric.hh"
#include "properties/NonCommuting.hh"
#include "properties/NumericalFlat.hh"
#include "properties/PartialDerivative.hh"
#include "properties/RiemannTensor.hh"
#include "properties/SatisfiesBianchi.hh"
#include "properties/SelfAntiCommuting.hh"
#include "properties/SelfCommuting.hh"
#include "properties/SelfNonCommuting.hh"
#include "properties/SortOrder.hh"
```

Classes

- class iterable_converter

Functions

- template<typename T >
  std::vector< T > to_std_vector (const boost::python::list &iterable)
- boost::python::list ProgressMonitor_totals_helper (ProgressMonitor &self)
- boost::python::list terms (const Ex &ex)
- Ex lhs (const Ex &ex)
- Ex rhs (const Ex &ex)
- Ex Ex_getslice (Ex &ex, boost::python::slice slice)
- Ex Ex_getitem (Ex &ex, int index)
- void Ex_setitem (Ex &ex, int index, Ex val)
- size_t Ex_len (Ex &ex)
- std::string Ex_str_ (const Ex &ex)

  Generate the Python str() and repr() representation of the Ex object.
- std::string Ex_latex_ (const Ex &ex)

  The Python 'print' function always calls the 'str' member on objects to be printed.
- std::string Ex_repr_ (const Ex &ex)
- boost::python::object Ex_to_Sympy (const Ex &ex)

  Convert a Cadabra 'Ex' to a Sympy expression.
- std::shared_ptr< Ex > fetch_from_python (const std::string &nm)

  Fetch an Ex object from the Python side using its Python identifier.
- bool __eq__Ex_Ex (const Ex &one, const Ex &other)

  Comparison operator for Ex objects in Python.
- bool __eq__Ex_int (const Ex &one, int other)

  Comparison operator for Ex objects in Python.
- std::shared_ptr< Ex > make_Ex_from_string (const std::string &ex_, bool make_ref=true)
- std::shared_ptr< Ex > construct_Ex_from_string (const std::string &ex_)
- std::shared_ptr< Ex > construct_Ex_from_string_2 (const std::string &ex_, bool add_ref)
- std::shared_ptr< Ex > make_Ex_from_int (int num, bool make_ref=true)
- std::shared_ptr< Ex > construct_Ex_from_int (int num)
- std::shared_ptr< Ex > construct_Ex_from_int_2 (int num, bool add_ref)
- Ex operator+ (const Ex &ex1, const Ex &ex2)

  Add two expressions, adding a top-level node if required.
- Ex operator- (const Ex &ex1, const Ex &ex2)

  Subtract two expressions, adding a top-level node if required.
- std::string init_ipython ()
- boost::python::list list_properties ()
- std::string print_tree (Ex *ex)
- PyObject * createExceptionClass (const char *name, PyObject *baseTypeObj=PyExc_Exception)
- void translate_ArgumentException (const ArgumentException &x)
- void translate_ConsistencyException (const ConsistencyException &x)
- void translate_ParseException (const ParseException &e)
- void translate_RuntimeException (const RuntimeException &e)
- void translate_NonScalarException (const NonScalarException &e)
- void translate_InternalError (const InternalError &e)

- void translate_NotYetImplemented (const NotYetImplemented &e)
- Kernel * get_kernel_from_scope ()

  Get a pointer to the currently visible kernel.
- Kernel * create_scope ()

  Setup of kernels in current scope, callable from Python.
- Kernel * create_scope_from_global ()
- Kernel * create_empty_scope ()
- void inject_defaults (Kernel *k)

  *Inject properties directly into the Kernel, even if the kernel is not yet on the Python stack (needed when we create a new local scope: in this case we create the kernel and pass it back to be turned into local **cdbkernel** by Python, but we want to populate the kernel with defaults before we hand it back).*
- template<class F >
  Ex * dispatch_base (Ex &ex, F &algo, bool deep, bool repeat, unsigned int depth, bool pre_order)
- Ex * map_sympy_wrapper (Ex &ex, std::string head)
- void call_post_process (Kernel &kernel, Ex &ex)

  Run the post-process Python function (if defined) on the given expression.
- template<class F >
  Ex * dispatch_ex (Ex &ex, bool deep, bool repeat, unsigned int depth)
- template<class F , typename Arg1 >
  Ex * dispatch_ex (Ex &ex, Arg1 arg, bool deep, bool repeat, unsigned int depth)
- template<class F , typename Arg1 , typename Arg2 >
  Ex * dispatch_ex (Ex &ex, Arg1 arg1, Arg2 arg2, bool deep, bool repeat, unsigned int depth)
- template<class F , typename... Args>
  Ex * dispatch_string (const std::string &ex, Args...args, bool deep, bool repeat, unsigned int depth)
- template<class F >
  void def_algo_1 (const std::string &name)
- template<class P >
  void def_prop ()
- BOOST_PYTHON_MODULE (cadabra2)

Variables

- bool output_ipython =false
- bool post_process_enabled =true
- PyObject * ParseExceptionType = NULL
- PyObject * ArgumentExceptionType = 0
- PyObject * ConsistencyExceptionType = 0
- PyObject * RuntimeExceptionType = 0
- PyObject * NonScalarExceptionType = NULL
- PyObject * InternalErrorType = NULL
- PyObject * NotYetImplementedType = NULL
- ProgressMonitor * pm =0

10.280.1 Function Documentation

10.280.1.1 BOOST_PYTHON_MODULE (cadabra2)

10.280.1.2 std::shared_ptr<Ex> construct_Ex_from_int (int *num*)

10.280.1.3 std::shared_ptr<Ex> construct_Ex_from_int_2 (int *num*, bool *add_ref*)

10.280.1.4 std::shared_ptr<Ex> construct_Ex_from_string (const std::string & *ex_*)

10.280.1.5 std::shared_ptr<Ex> construct_Ex_from_string_2 (const std::string & *ex_*, bool *add_ref*)

10.280.1.6 Kernel* create_empty_scope ( )

10.280.1.7 Kernel* create_scope_from_global ( )

10.280.1.8 PyObject* createExceptionClass (const char * *name*, PyObject * *baseTypeObj* = `PyExc_Exception`)

10.280.1.9 template<class F > void def_algo_1 (const std::string & *name*)

10.280.1.10 template<class P > void def_prop ( )

10.280.1.11 template<class F > Ex* dispatch_base (Ex & *ex*, F & *algo*, bool *deep*, bool *repeat*, unsigned int *depth*, bool *pre_order*)

10.280.1.12 template<class F > Ex* dispatch_ex (Ex & *ex*, bool *deep*, bool *repeat*, unsigned int *depth*)

10.280.1.13 template<class F , typename Arg1 > Ex* dispatch_ex (Ex & *ex*, Arg1 *arg*, bool *deep*, bool *repeat*, unsigned int *depth*)

10.280.1.14 template<class F , typename Arg1 , typename Arg2 > Ex* dispatch_ex (Ex & *ex*, Arg1 *arg1*, Arg2 *arg2*, bool *deep*, bool *repeat*, unsigned int *depth*)

10.280.1.15 template<class F , typename... Args> Ex* dispatch_string (const std::string & *ex*, Args... *args*, bool *deep*, bool *repeat*, unsigned int *depth*)

10.280.1.16 Ex Ex_getitem (Ex & *ex*, int *index*)

10.280.1.17 Ex Ex_getslice (Ex & *ex*, boost::python::slice *slice*)

10.280.1.18 size_t Ex_len (Ex & *ex*)

10.280.1.19 std::string Ex_repr_ (const Ex & *ex*)

10.280.1.20 void Ex_setitem (Ex & *ex*, int *index*, Ex *val*)

10.280.1.21 std::shared_ptr<Ex> fetch_from_python (const std::string & *nm*)

Fetch an Ex object from the Python side using its Python identifier.

10.280.1.22 std::string init_ipython ( )

10.280.1.23 Ex lhs (const Ex & *ex*)

10.280.1.24 boost::python::list list_properties ( )

10.280.1.25 std::shared_ptr<Ex> make_Ex_from_int (int *num,* bool *make_ref* = `true`)

10.280.1.26 std::shared_ptr<Ex> make_Ex_from_string (const std::string & *ex_,* bool *make_ref* = `true`)

10.280.1.27 Ex* map_sympy_wrapper (Ex & *ex,* std::string *head*)

10.280.1.28 std::string print_tree (Ex * *ex*)

10.280.1.29 boost::python::list ProgressMonitor_totals_helper (ProgressMonitor & *self*)

10.280.1.30 Ex rhs (const Ex & *ex*)

10.280.1.31 boost::python::list terms (const Ex & *ex*)

10.280.1.32 template<typename T > std::vector<T> to_std_vector (const boost::python::list & *iterable*)

10.280.1.33 void translate_ArgumentException (const ArgumentException & *x*)

10.280.1.34 void translate_ConsistencyException (const ConsistencyException & *x*)

10.280.1.35 void translate_InternalError (const InternalError & *e*)

10.280.1.36 void translate_NonScalarException (const NonScalarException & *e*)

10.280.1.37 void translate_NotYetImplemented (const NotYetImplemented & *e*)

10.280.1.38 void translate_ParseException (const ParseException & *e*)

10.280.1.39 void translate_RuntimeException (const RuntimeException & *e*)

10.280.2 Variable Documentation

10.280.2.1 PyObject* ArgumentExceptionType = 0

10.280.2.2 PyObject* ConsistencyExceptionType = 0

10.280.2.3 PyObject* InternalErrorType = NULL

10.280.2.4 PyObject* NonScalarExceptionType = NULL

10.280.2.5 PyObject* NotYetImplementedType = NULL

10.280.2.6 bool output_ipython =false

10.280.2.7 PyObject* ParseExceptionType = NULL

10.280.2.8 ProgressMonitor* pm =0

10.280.2.9 bool post_process_enabled =true

10.280.2.10 PyObject* RuntimeExceptionType = 0

10.281 core/PythonCdb.hh File Reference

```
#include <boost/python.hpp>
#include <stdexcept>
#include "Storage.hh"
#include "Kernel.hh"
#include "Algorithm.hh"
```

Classes

- class BaseProperty

  Helper class to ensure that all Python property objects derive from the same base class.
- class Property< T >

  Property is a templated wrapper around a C++ property object.

Functions

- bool __eq__Ex_Ex (const cadabra::Ex &, const cadabra::Ex &)

  Comparison operator for Ex objects in Python.
- bool __eq__Ex_int (const cadabra::Ex &, int)

  Comparison operator for Ex objects in Python.
- std::shared_ptr< cadabra::Ex > fetch_from_python (const std::string &nm)

  Fetch an Ex object from the Python side using its Python identifier.
- std::string Ex_str_ (const cadabra::Ex &)

  Generate the Python str() and repr() representation of the Ex object.
- std::string Ex_repr_ (const cadabra::Ex &)
- std::string Ex_latex_ (const cadabra::Ex &)

  The Python 'print' function always calls the 'str' member on objects to be printed.
- boost::python::object Ex_to_Sympy (const cadabra::Ex &)

  Convert a Cadabra 'Ex' to a Sympy expression.
- cadabra::Ex operator+ (const cadabra::Ex &ex1, const cadabra::Ex &ex2)

  Add two expressions, adding a top-level node if required.
- cadabra::Ex operator- (const cadabra::Ex &ex1, const cadabra::Ex &ex2)

  Subtract two expressions, adding a top-level node if required.
- cadabra::Kernel * create_scope ()

Setup of kernels in current scope, callable from Python.

- cadabra::Kernel * create_scope_from_global ()
- cadabra::Kernel * create_empty_scope ()
- void inject_defaults (cadabra::Kernel *)

  *Inject properties directly into the Kernel, even if the kernel is not yet on the Python stack (needed when we create a new local scope: in this case we create the kernel and pass it back to be turned into local **cdbkernel** by Python, but we want to populate the kernel with defaults before we hand it back).*
- cadabra::Kernel * get_kernel_from_scope ()

  Get a pointer to the currently visible kernel.
- void call_post_process (cadabra::Kernel &, cadabra::Ex &ex)

  Run the post-process Python function (if defined) on the given expression.

10.281.1 Function Documentation

10.281.1.1 cadabra::Kernel* create_empty_scope ( )

10.281.1.2 cadabra::Kernel* create_scope_from_global ( )

10.281.1.3 std::string Ex_repr_ (const cadabra::Ex &)

10.281.1.4 std::shared_ptr<cadabra::Ex> fetch_from_python (const std::string & *nm*)

Fetch an Ex object from the Python side using its Python identifier.

10.282 core/PythonException.cc File Reference

```
#include <boost/python.hpp>
#include <stdexcept>
#include "PythonException.hh"
```

Functions

- std::string parse_python_exception ()

10.282.1 Function Documentation

10.282.1.1 std::string parse_python_exception ( )

10.283 core/PythonException.hh File Reference

```
#include <string>
```

Functions

- std::string parse_python_exception ()

10.283.1 Function Documentation

10.283.1.1 std::string parse_python_exception ( )

10.284 core/ServerWrapper.cc File Reference

```
#include "ServerWrapper.hh"
#include <boost/python.hpp>
```

10.285 core/ServerWrapper.hh File Reference

```
#include "ProgressMonitor.hh"
#include <boost/python/implicit.hpp>
```

Classes

- class ServerWrapper

  A class which enables access to a Python-side defined Server object with ProgressMonitor interface.

10.286 core/Stopwatch.cc File Reference

```
#include "Stopwatch.hh"
#include <assert.h>
```

Functions

- std::ostream & operator<< (std::ostream &os, const Stopwatch &mt)

10.286.1 Function Documentation

10.286.1.1 std::ostream& operator<< (std::ostream & *os*, const Stopwatch & *mt*)

10.287 core/Stopwatch.hh File Reference

```
#include <sys/time.h>
#include <unistd.h>
#include <signal.h>
#include <iostream>
```

Classes

- class Stopwatch

Functions

- std::ostream & operator<< (std::ostream &, const Stopwatch &)

10.287.1 Function Documentation

10.287.1.1 std::ostream& operator<< (std::ostream & , const Stopwatch &)

10.288 core/Storage.cc File Reference

```
#include "Storage.hh"
#include "Combinatorics.hh"
#include "Exceptions.hh"
#include <iomanip>
#include <sstream>
#include <pcrecpp.h>
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Functions

- long cadabra::to_long (multiplier_t mul)
- std::string cadabra::to_string (long num)
- void cadabra::multiply (rset_t::iterator &, multiplier_t)

  Helper functions for manipulation of multipliers.
- void cadabra::add (rset_t::iterator &num, multiplier_t fac)
- void cadabra::zero (rset_t::iterator &num)
- void cadabra::one (rset_t::iterator &num)
- void cadabra::flip_sign (rset_t::iterator &num)
- void cadabra::half (rset_t::iterator &num)
- std::ostream & operator<< (std::ostream &str, const cadabra::Ex &ex)

Variables

- nset_t cadabra::name_set
- rset_t cadabra::rat_set

10.288.1 Function Documentation

10.288.1.1 std::ostream& operator<< (std::ostream & *str,* const cadabra::Ex & *ex*)

10.289 core/Storage.hh File Reference

```
#include <cstddef>
#include <iostream>
#include <gmpxx.h>
#include <string>
#include <vector>
#include <set>
#include <map>
#include <stdint.h>
#include <assert.h>
#include <initializer_list>
#include "tree.hh"
```

Classes

- class cadabra::str_node

  Elementary building block for a mathematical expression.
- struct cadabra::str_node::flag_t
- class cadabra::Ex

  Basic storage class for symbolic mathememattical expressions.
- class cadabra::nset_it_less

  Compare two nset iterators by comparing the strings to which they point.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

Typedefs

- typedef mpq_class cadabra::multiplier_t
- typedef std::set< std::string > cadabra::nset_t
- typedef std::set< multiplier_t > cadabra::rset_t
- typedef uintptr_t cadabra::hashval_t

Functions

- long cadabra::to_long (multiplier_t mul)
- std::string cadabra::to_string (long num)
- void cadabra::multiply (rset_t::iterator &, multiplier_t)

  Helper functions for manipulation of multipliers.
- void cadabra::add (rset_t::iterator &num, multiplier_t fac)
- void cadabra::zero (rset_t::iterator &num)
- void cadabra::one (rset_t::iterator &num)
- void cadabra::flip_sign (rset_t::iterator &num)
- void cadabra::half (rset_t::iterator &num)
- template<typename T >
  bool cadabra::is_in (const T &val, const std::initializer_list< T > &list)

  Bare output operator for Ex objects, mainly to provide a simple way to generate debugging output.
- std::ostream & operator<< (std::ostream &, const cadabra::Ex &)

10.289.1 Function Documentation

10.289.1.1 std::ostream& operator<< (std::ostream & , const cadabra::Ex &)

10.290 core/Symbols.cc File Reference

```
#include "Symbols.hh"
```

10.291 core/Symbols.hh File Reference

```
#include <set>
#include <string>
```

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.
- cadabra::symbols

Variables

- const std::set< std::string > cadabra::symbols::greek

10.292 core/SympyCdb.cc File Reference

```
#include <boost/python.hpp>
#include "Functional.hh"
#include "SympyCdb.hh"
#include "PreClean.hh"
#include "Cleanup.hh"
#include "Parser.hh"
#include "Kernel.hh"
#include "DisplaySympy.hh"
#include "algorithms/substitute.hh"
```

10.293 core/SympyCdb.hh File Reference

```
#include "Props.hh"
#include "Storage.hh"
#include "Kernel.hh"
#include "Stopwatch.hh"
```

Namespaces

- sympy

Functions

- cadabra::Ex * sympy::map_sympy (const cadabra::Kernel &, cadabra::Ex &, const std::vector< std::string > &wrap, const std::string &args, const std::string &method)

  Functionality to act with Sympy on all scalar parts of an expression, and keep the result in-place.
- cadabra::Ex::iterator sympy::apply (const cadabra::Kernel &, cadabra::Ex &, cadabra::Ex::iterator &, const std::vector< std::string > &wrap, const std::string &args, const std::string &method)

  Functionality to act with Sympy functions on (parts of) Cadabra Ex expressions and read the result back into the same Ex.
- cadabra::Ex sympy::invert_matrix (const cadabra::Kernel &, cadabra::Ex &ex, cadabra::Ex &rules)

  Use Sympy to invert a matrix, given a set of rules determining its sparse components.

10.294 core/TerminalStream.cc File Reference

```
#include "DisplayTerminal.hh"
#include "TerminalStream.hh"
```

10.295 core/TerminalStream.hh File Reference

```
#include "Kernel.hh"
#include <ostream>
```

Classes

- class cadabra::TerminalStream

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.296 core/test_permutations.cc File Reference

```
#include "Permutations.hh"
#include <iostream>
```

Functions

- int main ()

10.296.1 Function Documentation

10.296.1.1 int main ( )

10.297 core/test_preprocessor.cc File Reference

```
#include <stdlib.h>
#include <stdexcept>
#include <iostream>
#include <unistd.h>
#include <sstream>
#include "PreProcessor.hh"
```

Functions

- bool testit (const std::string &input, const std::string &output)
- int main (int argc, char **argv)

10.297.1 Function Documentation

10.297.1.1 int main (int *argc,* char ** *argv*)

10.297.1.2 bool testit (const std::string & *input,* const std::string & *output*)

10.298 core/YoungTab.cc File Reference

```
#include "YoungTab.hh"
#include <cassert>
```

Namespaces

- yngtab

  Generic Young tableaux routines.

Functions

- std::ostream & yngtab::operator<< (std::ostream &str, const tableau &tab)
- template<>
  void yngtab::add_box (tableau &tab1, unsigned int row1, const tableau &tab2, unsigned int row2, unsigned int col2)

10.299 core/YoungTab.hh File Reference

```
#include <cstddef>
#include <iostream>
#include <iterator>
#include <vector>
#include <list>
#include <gmpxx.h>
#include "Combinatorics.hh"
```

Classes

- class yngtab::tableau_base
- class yngtab::tableau
- class yngtab::tableaux< T >
- class yngtab::filled_tableau< T >
- class yngtab::filled_tableau< T >::iterator_base
- class yngtab::filled_tableau< T >::in_column_iterator

  An iterator which stays inside a given column of a tableau.
- class yngtab::filled_tableau< T >::iterator

  An iterator over all boxes of a tableau, left to right, top to bottom.
- class yngtab::tableaux< T >

Namespaces

- yngtab

  Generic Young tableaux routines.

Typedefs

- typedef mpz_class yngint_t
- typedef mpq_class yngrat_t
- typedef filled_tableau< std::pair< int, int > > yngtab::keeptrack_tab_t

Functions

- bool yngtab::legal_box (const std::vector< std::pair< int, int > > &prev, const std::vector< std::pair< int, int > > &ths, int colpos, int trypos)
- template<class T , class OutputIterator >
  void yngtab::LR_tensor (const tableaux< T > &tabs1, const T &tab2, unsigned int maxrows, OutputIterator out, bool alltabs=false)
- template<class T1 , class T2 >
  void yngtab::add_box (T1 &tab1, unsigned int row1, const T2 &tab2, unsigned int row2, unsigned int col2)
- template<class T1 >
  void yngtab::add_box (T1 &tab1, unsigned int row1, const tableau &tab2, unsigned int row2, unsigned int col2)
- template<class Tab , class OutputIterator >
  void yngtab::LR_add_box (const Tab &tab2, Tab &newtab, unsigned int currow2, unsigned int curcol2, unsigned int startrow, unsigned int maxrows, OutputIterator outit, keeptrack_tab_t &Ycurrent, bool alltabs)

- template<class Tab , class OutputIterator >
  void yngtab::LR_tensor (const Tab &tab1, const Tab &tab2, unsigned int maxrows, OutputIterator outit, bool alltabs=false)
- template<class T , class OutputIterator >
  void yngtab::LR_tensor (const tableaux< T > &, bool symmetric, unsigned int maxrows, OutputIterator outit)
- std::ostream & yngtab::operator<< (std::ostream &str, const tableau &tab)
- template<class T >
  std::ostream & yngtab::operator<< (std::ostream &, const tableaux< T > &)
- template<class T >
  std::ostream & yngtab::operator<< (std::ostream &, const filled_tableau< T > &)

10.299.1 Typedef Documentation

10.299.1.1 typedef mpz_class yngint_t

10.299.1.2 typedef mpq_class yngrat_t

10.300 doc/main.md File Reference

10.301 doc/modules.dox File Reference

10.302 frontend/common/exec-stream.cc File Reference

```
#include "exec-stream.h"
#include <list>
#include <vector>
#include <algorithm>
#include <stdexcept>
#include <errno.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <signal.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <pthread.h>
```

Macros

- #define HELPERS_H "posix/exec-stream-helpers.h"
- #define HELPERS_CPP "posix/exec-stream-helpers.cpp"
- #define IMPL_CPP "posix/exec-stream-impl.cpp"

10.302.1 Macro Definition Documentation

10.302.1.1 #define HELPERS_CPP "posix/exec-stream-helpers.cpp"

10.302.1.2 #define HELPERS_H "posix/exec-stream-helpers.h"

10.302.1.3 #define IMPL_CPP "posix/exec-stream-impl.cpp"

10.303 frontend/common/exec-stream.h File Reference

```
#include <string>
#include <stdexcept>
#include <istream>
#include <ostream>
#include <vector>
```

Classes

- class exec_stream_t
- class exec_stream_t::error_t
- class exec_stream_t::next_arg_t
- class exec_stream_t::next_arg_impl_t< iterator >

10.304 frontend/common/test_tex.cc File Reference

```
#include "TeXEngine.hh"
```

Functions

- int main ()

10.304.1 Function Documentation

10.304.1.1 int main ( )

10.305 frontend/common/TeXEngine.cc File Reference

```
#include "TeXEngine.hh"
#include "Config.hh"
#include <iostream>
#include <boost/filesystem.hpp>
#include <boost/algorithm/string.hpp>
#include "exec-stream.h"
#include "lodepng.h"
#include <fstream>
#include <sstream>
#include <unistd.h>
```

10.306 frontend/common/TeXEngine.hh File Reference

```
#include <stdexcept>
#include <vector>
#include <set>
#include <string>
#include <memory>
```

Classes

- class cadabra::TeXEngine

  TeXEngine is used to convert LaTeX strings into PNG images.
- class cadabra::TeXEngine::TeXException
- class cadabra::TeXEngine::TeXRequest

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.307 frontend/gtkmm/Cadabra.cc File Reference

```
#include "Cadabra.hh"
#include <signal.h>
#include <fstream>
#include <gtkmm/messagedialog.h>
#include <gtkmm/entry.h>
#include <gtkmm/main.h>
#include "Snoop.hh"
#include "Config.hh"
```

Functions

- void signal_handler (int signal)

Variables

- cadabra::NotebookWindow * signal_window

10.307.1 Function Documentation

10.307.1.1 void signal_handler (int *signal*)

10.307.2 Variable Documentation

10.307.2.1 cadabra::NotebookWindow* signal_window

10.308 frontend/gtkmm/Cadabra.hh File Reference

```
#include <gtkmm/application.h>
#include <gtkmm/grid.h>
#include "NotebookWindow.hh"
#include "ComputeThread.hh"
```

Classes

- class Cadabra

  The Cadabra notebook application.

10.309 frontend/gtkmm/CodeInput.cc File Reference

```
#include "CodeInput.hh"
#include <gdkmm/general.h>
#include <gtkmm/messagedialog.h>
#include <gdk/gdkkeysyms.h>
#include <iostream>
```

Functions

- std::string trim (const std::string &s)

10.309.1 Function Documentation

10.309.1.1 std::string trim (const std::string & *s*)

10.310 frontend/gtkmm/CodeInput.hh File Reference

```
#include "DataCell.hh"
#include <gtkmm/box.h>
#include <gtkmm/textview.h>
#include <gtkmm/separator.h>
```

Classes

- class cadabra::CodeInput

  A text cell editor widget with support for editing Python and LaTeX.
- class cadabra::CodeInput::exp_input_tv

  The actual text widget used by CodeInput.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.311 frontend/gtkmm/ImageView.cc File Reference

```
#include "ImageView.hh"
#include <giomm/memoryinputstream.h>
#include <glibmm/base64.h>
#include <iostream>
#include <fstream>
```

10.312 frontend/gtkmm/ImageView.hh File Reference

```
#include <gtkmm/box.h>
#include <gtkmm/image.h>
```

Classes

- class cadabra::ImageView

  An image viewing widget.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.313 frontend/gtkmm/main.cc File Reference

```
#include "Cadabra.hh"
#include <gtkmm/application.h>
#include <gtkmm/cssprovider.h>
#include <gtkmm/label.h>
#include <glibmm/spawn.h>
```

Functions

- int main (int argc, char **argv)

10.313.1 Function Documentation

10.313.1.1 int main (int *argc,* char ** *argv*)

10.314 frontend/gtkmm/NotebookCanvas.cc File Reference

```
#include "NotebookCanvas.hh"
#include "NotebookWindow.hh"
#include <gtkmm/box.h>
#include <iostream>
```

10.315 frontend/gtkmm/NotebookCanvas.hh File Reference

```
#include <map>
#include <gtkmm/paned.h>
#include <gtkmm/scrolledwindow.h>
#include <gtkmm/separator.h>
#include <gtkmm/eventbox.h>
#include "VisualCell.hh"
```

Classes

- class cadabra::NotebookCanvas

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.316 frontend/osx/Cadabra/Cadabra/NotebookCanvas.hh File Reference

Classes

- class cadabra::NotebookCanvas

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.317 frontend/gtkmm/NotebookWindow.cc File Reference

```
#include <iostream>
#include "Actions.hh"
#include "Cadabra.hh"
#include "Config.hh"
#include "NotebookWindow.hh"
#include "DataCell.hh"
#include <gtkmm/box.h>
#include <gtkmm/filechooserdialog.h>
#include <gtkmm/messagedialog.h>
#include <gtkmm/aboutdialog.h>
#include <gtkmm/radioaction.h>
#include <gtkmm/scrollbar.h>
#include <fstream>
#include <gtkmm/main.h>
#include "Snoop.hh"
```

10.318 frontend/gtkmm/NotebookWindow.hh File Reference

```
#include <gtkmm/window.h>
#include <gtkmm/box.h>
#include <gtkmm/progressbar.h>
#include <gtkmm/spinner.h>
#include <gtkmm/label.h>
#include <gtkmm/stock.h>
#include <gtkmm/button.h>
#include <gtkmm/uimanager.h>
#include <gtkmm/cssprovider.h>
#include <glibmm/dispatcher.h>
#include <giomm/settings.h>
#include <thread>
#include <mutex>
#include "DocumentThread.hh"
#include "ComputeThread.hh"
#include "GUIBase.hh"
#include "NotebookCanvas.hh"
#include "../common/TeXEngine.hh"
```

Classes

- class cadabra::NotebookWindow

  Each notebook has one main window which controls it.
- class cadabra::NotebookWindow::Prefs

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.319 frontend/osx/Cadabra/Cadabra/NotebookWindow.hh File Reference

```
#include <stdio.h>
#include "DocumentThread.hh"
#include "GUIBase.hh"
#include "NotebookController.h"
#include "NotebookCanvas.hh"
```

Classes

- class cadabra_osx::NotebookWindow

  Objective-C++ class implementing DocumentThread and providing an OS-X notebook interface.

Namespaces

- cadabra_osx

10.320 frontend/gtkmm/Preferences.hh File Reference

```
#include <gtkmm/window.h>
#include <gtkmm/box.h>
```

Classes

- class PreferencesWindow

10.321 frontend/gtkmm/TeXView.cc File Reference

```
#include "TeXView.hh"
#include <iostream>
```

10.322 frontend/gtkmm/TeXView.hh File Reference

```
#include <gtkmm/eventbox.h>
#include <gtkmm/box.h>
#include <gtkmm/image.h>
#include "DataCell.hh"
#include "../common/TeXEngine.hh"
```

Classes

- class cadabra::TeXView

  TeXView is a widget which knows how to turn a string into a LaTeX-rendered image and display that.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.323 frontend/gtkmm/VisualCell.hh File Reference

```
#include "TeXView.hh"
#include "ImageView.hh"
#include "CodeInput.hh"
#include "DataCell.hh"
```

Classes

- class cadabra::VisualCell

  Structure holding a pointer to one of the possible GUI widgets that can appear in a document.

Namespaces

- cadabra

  Class to display expressions in a format that Sympy can parse.

10.324 frontend/osx/Cadabra/Cadabra/AppDelegate.h File Reference

```
#import <Cocoa/Cocoa.h>
#include "NotebookController.h"
```

Classes

- class AppDelegate

10.325 frontend/osx/Cadabra/Cadabra/AppDelegate.mm File Reference

```
#import "AppDelegate.h"
```

10.326 frontend/osx/Cadabra/Cadabra/main.m File Reference

```
#import <Cocoa/Cocoa.h>
```

Functions

- int main (int argc, const char ∗argv[ ])

10.326.1 Function Documentation

10.326.1.1 int main (int *argc,* const char ∗ *argv[ ]*)

10.327 frontend/osx/Cadabra/Cadabra/NotebookCanvas.mm File Reference

```
#import <Foundation/Foundation.h>
```

10.328 frontend/osx/Cadabra/Cadabra/NotebookWindow.mm File Reference

```
#include "NotebookWindow.hh"
#import <Cocoa/Cocoa.h>
```

10.329 frontend/osx/Cadabra/NotebookController.h File Reference

```
#import <Cocoa/Cocoa.h>
```

Classes

- class NotebookController

10.330 frontend/osx/Cadabra/NotebookController.mm File Reference

```
#import "NotebookController.h"
#include <thread>
#include "ComputeThread.hh"
#include "NotebookWindow.hh"
```

Variables

- cadabra::ComputeThread * compute
- std::thread * compute_thread

10.330.1 Variable Documentation

10.330.1.1 cadabra::ComputeThread* compute

Initial value:

```
{
    cadabra::NotebookWindow *nw
```

10.330.1.2 std::thread* compute_thread

Index

__cdbkernel__
    cadabra2_defaults, 54
__eq__Ex_Ex
    Python Core, 31
__eq__Ex_int
    Python Core, 31
__version__
    cadabra2_defaults::Sympy, 339
_displayhook
    cadabra2_defaults, 54
~ActionRemoveCell
    cadabra::ActionRemoveCell, 71
~ActionSplitCell
    cadabra::ActionSplitCell, 74
~Algorithm
    cadabra::Algorithm, 79
~AntiSymmetric
    cadabra::AntiSymmetric, 86
~Cadabra
    Cadabra, 92
~ComputeThread
    cadabra::ComputeThread, 115
~DAntiSymmetric
    cadabra::DAntiSymmetric, 120
~Derivative
    cadabra::Derivative, 129
~Distributable
    cadabra::Distributable, 145
~EpsilonTensor
    cadabra::EpsilonTensor, 161
~FilledTableau
    cadabra::FilledTableau, 197
~GammaTraceless
    cadabra::GammaTraceless, 202
~ImageView
    cadabra::ImageView, 208
~Inherit
    cadabra::Inherit, 220
~Integer
    cadabra::Integer, 221
~Integral
    Integral, 222
~Kernel
    cadabra::Kernel, 233
~KroneckerDelta
    cadabra::KroneckerDelta, 236
~Matrix
    cadabra::Matrix, 245
~NotebookCanvas
    cadabra::NotebookCanvas, 252
~NotebookWindow
    cadabra::NotebookWindow, 257
~PartialDerivative
    cadabra::PartialDerivative, 270
~ProgressMonitor
    ProgressMonitor, 280
~Server
    Server, 304
~Snoop
    snoop::Snoop, 309
~SnoopImpl
    snoop::SnoopImpl, 312
~Spinor
    cadabra::Spinor, 319
~Tableau
    cadabra::Tableau, 341
~TableauBase
    cadabra::TableauBase, 346
~TableauSymmetry
    cadabra::TableauSymmetry, 347
~TeXEngine
    cadabra::TeXEngine, 353
~TeXView
    cadabra::TeXView, 357
~Traceless
    cadabra::Traceless, 360
~combinations
    combin::combinations, 104
~combinations_base
    combin::combinations_base, 106
~error_t
    exec_stream_t::error_t, 162
~exec_stream_t
    exec_stream_t, 180
~filled_tableau
    yngtab::filled_tableau, 194
~next_arg_t
    exec_stream_t::next_arg_t, 249
~property
    cadabra::property, 287
~registered_property_map_t
    cadabra::Properties::registered_property_map_t, 291
~tableau
    yngtab::tableau, 342
~tableau_base
    yngtab::tableau_base, 344

accu

preprocessor::accu_t, 62
accu_t
preprocessor::accu_t, 62
accus
preprocessor, 277
ActionAddCell
cadabra::ActionAddCell, 64
cadabra::DocumentThread, 150
ActionEraseText
cadabra::ActionEraseText, 67
cadabra::DocumentThread, 150
ActionInsertText
cadabra::ActionInsertText, 68
cadabra::DocumentThread, 150
ActionPositionCursor
cadabra::ActionPositionCursor, 70
cadabra::DocumentThread, 150
ActionRemoveCell
cadabra::ActionRemoveCell, 71
cadabra::DocumentThread, 150
ActionSetRunStatus
cadabra::ActionSetRunStatus, 72
cadabra::DocumentThread, 150
ActionSplitCell
cadabra::ActionSplitCell, 74
cadabra::DocumentThread, 150
ActionStack
cadabra::DocumentThread, 148
actiongroup
cadabra::NotebookWindow, 262
add
cadabra, 49
add_box
yngtab, 59
yngtab::filled_tableau, 194
yngtab::tableau, 342
yngtab::tableau_base, 344
add_cell
cadabra::GUIBase, 203
cadabra::NotebookWindow, 258
cadabra_osx::NotebookWindow, 254
add_row
yngtab::tableau_base, 344
add_tableau
yngtab::tableaux, 349
additive
cadabra::WeightInherit, 374
advance
cadabra::Parser, 269
after
cadabra::ActionAddCell, 63
after_tree_change
MyClient, 247
Algorithm
cadabra::Algorithm, 79
algorithm
cadabra::DocumentThread, 148
Algorithms, 35
all_cells_nonrunning
cadabra::ComputeThread, 115
always
cadabra::Ex_comparator, 173
anticomm
cadabra::order, 266
app_name
snoop::Snoop::AppEntry, 88
app_version
snoop::Snoop::AppEntry, 88
AppDelegate, 86
AppDelegate(), 87
notebookController, 87
window, 87
AppEntry
snoop::Snoop::AppEntry, 88
append_prod_
cadabra::join_gamma, 230
apply
cadabra::Algorithm, 79
cadabra::canonicalise, 94
cadabra::collect_components, 99
cadabra::collect_factors, 100
cadabra::collect_terms, 102
cadabra::combine, 109
cadabra::complete, 113
cadabra::decompose_product, 124
cadabra::distribute, 146
cadabra::drop_weight, 153
cadabra::einsteinify, 155
cadabra::eliminate_converter, 156
cadabra::eliminate_kronecker, 157
cadabra::epsilon_to_delta, 159
cadabra::evaluate, 163
cadabra::expand, 183
cadabra::expand_delta, 184
cadabra::expand_diracbar, 185
cadabra::expand_power, 186
cadabra::factor_in, 188
cadabra::factor_out, 190
cadabra::fierz, 191
cadabra::flatten_product, 199
cadabra::flatten_sum, 200
cadabra::indexsort, 217
cadabra::integrate_by_parts, 223
cadabra::join_gamma, 230
cadabra::keep_terms, 232
cadabra::keep_weight, 233
cadabra::lr_tensor, 242
cadabra::map_sympy, 244
cadabra::order, 266
cadabra::product_rule, 279
cadabra::reduce_delta, 290
cadabra::rename_dummies, 293
cadabra::replace_match, 294
cadabra::rewrite_indices, 295
cadabra::sort_product, 315
cadabra::sort_spinors, 316

cadabra::sort_sum, 317
cadabra::split_gamma, 321
cadabra::split_index, 323
cadabra::substitute, 329
cadabra::sym, 331
cadabra::take_match, 350
cadabra::unwrap, 368
cadabra::vary, 369
cadabra::young_project, 378
cadabra::young_project_product, 379
cadabra::young_project_tensor, 380
component, 114
Perm, 272
rename_dummies, 292
Scalar algebra, 34
split, 321
apply_deep
cadabra::Algorithm, 79
apply_generic
cadabra::Algorithm, 79
apply_once
cadabra::Algorithm, 80
apply_pre_order
cadabra::Algorithm, 80
apply_symmetry
combin::symmetriser, 337
architecture
cadabra2_defaults::Server, 301
Server, 304
arg
cadabra::Ex, 168
cadabra::drop_keep_weight, 152
arg_size
cadabra::Ex, 168
arg_to_num
cadabra::Ex, 168
argloc_2_treeloc
cadabra::sym, 331
args
cadabra::substitute, 330
cadabra::vary, 370
ArgumentException, 89
ArgumentException, 89
py_what, 89
ArgumentExceptionType
PythonCdb.cc, 475
asym_ranges
cadabra::decompose_product, 125
cadabra::young_project, 378
away_from
cadabra::integrate_by_parts, 223

b_curly
cadabra::str_node, 326
b_invalid
cadabra::str_node, 326
b_no
cadabra::str_node, 326
b_none
cadabra::str_node, 326
b_pointy
cadabra::str_node, 326
b_round
cadabra::str_node, 326
b_square
cadabra::str_node, 326
BOOST_PYTHON_MODULE
ptest.cc, 469
PythonCdb.cc, 473
BOOST_SPIRIT_THREADSAFE
Snoop.cc, 392
back_insert_iterator
yngtab::tableaux, 349
banner
echokernel::EchoKernel, 154
BaseProperty, 89
before
cadabra::ActionAddCell, 63
before_tree_change
MyClient, 247
begin
cadabra::index_iterator, 214
cadabra::keyval_t, 235
yngtab::filled_tableau, 195
begin_column
yngtab::filled_tableau, 195
begin_index
cadabra::Algorithm, 80
Block
ProgressMonitor::Block, 90
Server::Block, 91
block_available
Server, 306
block_available_mutex
Server, 306
block_length
combin::combinations_base, 107
combin::symmetriser, 338
block_queue
Server, 306
bottomline
cadabra::NotebookCanvas, 252
box_row
yngtab::filled_tableau, 194
bracket
cadabra::str_node::flag_t, 198
preprocessor::accu_t, 62
bracket_level
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 144
bracket_strings_
preprocessor, 276
bracket_t
cadabra::str_node, 326
buffer
cadabra::CodeInput, 98
build_visual_representation

cadabra::DocumentThread, 149

Cadabra, 91
~Cadabra, 92
Cadabra, 92
compute, 92
compute_thread, 92
create, 92
on_activate, 92
on_open, 92
open_help, 92
cadabra, 41
add, 49
cdb2python, 49
check_index_consistency, 49
cleanup_components, 49
cleanup_derivative, 50
cleanup_diagonal, 50
cleanup_dispatch, 50
cleanup_dispatch_deep, 50
cleanup_exterior_derivative, 50
cleanup_frac, 50
cleanup_indexbracket, 50
cleanup_kronecker, 50
cleanup_numericalflat, 50
cleanup_partialderivative, 50
cleanup_productlike, 50
cleanup_rational, 50
cleanup_sub, 51
cleanup_sumlike, 51
cleanup_updown, 51
convert_line, 51
DTree, 49
dispatcher_t, 49
escape_quotes, 51
export_as_HTML, 51
export_as_LaTeX, 51
export_as_python, 51
flip_sign, 51
HTML_recurse, 51
half, 51
hashval_t, 49
JSON_deserialise, 51
JSON_in_recurse, 51
JSON_recurse, 52
JSON_serialise, 52
LaTeX_recurse, 52
latex_to_html, 52
list_size, 52
multiplier_t, 49
name_set, 53
nset_t, 49
one, 52
operator<, 52
pre_clean_dispatch, 52
pre_clean_dispatch_deep, 52
push_down_multiplier, 52
python_recurse, 52
rat_set, 53
rset_t, 49
subtree_equal, 53
subtree_exact_equal, 53
subtree_exact_less, 53
subtree_less, 53
to_long, 53
to_string, 53
tree_equal, 53
tree_exact_equal, 53
tree_exact_less, 53
tree_less, 53
unichar, 53
zero, 53
cadabra-server.cc
main, 384
Cadabra.cc
signal_handler, 488
signal_window, 488
cadabra2_defaults, 53
__cdbkernel__, 54
_displayhook, 54
display, 54
displayhook, 54
have_matplotlib, 54
mclose, 54
mopen, 54
post_process, 54
remember_display_hook, 54
server, 54
sympy, 54
cadabra2_defaults.Server, 301
cadabra2_defaults.Sympy, 338
cadabra2_defaults::Server
architecture, 301
handles, 301
send, 301
test, 301
cadabra2_defaults::Sympy
__version__, 339
cadabra2cadabra.cc
main, 384
cadabra2html.cc
main, 385
cadabra2python.cc
main, 423
cadabra::Accent, 61
name, 61
cadabra::ActionAddCell, 63
ActionAddCell, 64
after, 63
before, 63
child, 63
child_num, 64
execute, 64
newcell, 64
newref, 64
pos, 64
Position, 63

ref, 64
revert, 64
cadabra::ActionBase, 65
execute, 65
revert, 65
undoable, 65
cadabra::ActionEraseText, 66
ActionEraseText, 67
execute, 67
from_pos, 67
removed_text, 67
revert, 67
this_cell, 67
to_pos, 67
cadabra::ActionInsertText, 67
ActionInsertText, 68
execute, 68
insert_pos, 68
revert, 68
text, 68
this_cell, 68
cadabra::ActionPositionCursor, 69
ActionPositionCursor, 70
execute, 70
in, 69
needed_new_cell, 70
newref, 70
next, 69
pos, 70
Position, 69
previous, 69
ref, 70
revert, 70
cadabra::ActionRemoveCell, 70
∼ActionRemoveCell, 71
ActionRemoveCell, 71
execute, 71
reference_child_index, 71
reference_parent_cell, 71
removed_tree, 71
revert, 71
this_cell, 71
cadabra::ActionSetRunStatus, 72
ActionSetRunStatus, 72
execute, 72
new_running_, 73
revert, 72
this_cell, 73
undoable, 73
was_running_, 73
cadabra::ActionSplitCell, 73
∼ActionSplitCell, 74
ActionSplitCell, 74
execute, 74
newref, 74
revert, 74
this_cell, 74
cadabra::Algorithm, 75
∼Algorithm, 79
Algorithm, 79
apply, 79
apply_deep, 79
apply_generic, 79
apply_once, 80
apply_pre_order, 80
begin_index, 80
can_apply, 80
check_consistency, 80
check_degree_consistency, 80
check_index_consistency, 80
classify_add_index, 80
classify_indices, 80
classify_indices_up, 80
compare_, 80
contains, 80
determine_intersection, 81
discard_command_node, 84
dumpmap, 81
end_index, 81
equal_without_numbers, 81
fill_index_position_map, 81
fill_map, 81
find_arg_superset, 81
find_argument_lists, 81
find_modulo_parent_rel, 81
force_prod_wrap, 81
get_dummy, 81, 82
get_dummy_sw, 84
index_in_set, 82
index_map_t, 78
index_parity, 82
index_position_map_t, 78
index_sw, 84
interrupted, 84
intersection_number, 82
is_factorlike, 82
is_nonprod_factor_in_prod, 82
is_single_term, 82
is_termlike, 82
iterator, 78
kernel, 84
less_without_numbers, 82
locate_object_set, 82
locate_single_object, 82
max_numbered_name, 82
max_numbered_name_one, 82
node_integer, 82
node_one, 83
node_zero, 83
number_of_calls, 84
number_of_direct_indices, 83
number_of_indices, 83
number_of_modifications, 84
pm, 84
post_order_iterator, 79
print_classify_indices, 83

prod_unwrap_single_term, 83
prod_wrap_single_term, 83
propagate_zeroes, 83
pushup_multiplier, 83
range_t, 79
range_vector_t, 79
rename_replacement_dummies, 83
report_progress, 83
report_progress_stopwatch, 84
result_t, 79
separated_by_derivative, 83
set_progress_monitor, 83
sibling_iterator, 79
suppress_normal_output, 84
tr, 84
cadabra::AntiCommuting, 84
name, 85
sign, 85
cadabra::AntiSymmetric, 85
∼AntiSymmetric, 86
get_tab, 86
name, 86
size, 86
cadabra::CodeInput, 96
buffer, 98
CodeInput, 97
edit, 98
handle_button_press, 98
handle_erase, 98
handle_insert, 98
init, 98
set_font_size, 98
slice_cell, 98
update_buffer, 98
cadabra::CodeInput::exp_input_tv, 181
cell_got_focus, 182
CodeInput, 182
content_changed, 182
content_erase, 182
content_execute, 182
content_insert, 182
datacell, 182
exp_input_tv, 182
on_draw, 182
on_focus_in_event, 182
on_key_press_event, 182
on_show, 182
scale_, 182
shift_enter_pressed, 182
cadabra::Commuting, 109
name, 110
sign, 110
cadabra::CommutingAsProduct, 110
name, 111
cadabra::CommutingAsSum, 111
name, 111
cadabra::CommutingBehaviour, 112
equals, 112
sign, 112
cadabra::ComputeThread, 114
∼ComputeThread, 115
all_cells_nonrunning, 115
ComputeThread, 115
connection, 117
connection_is_open, 117
docthread, 117
execute_cell, 115
find_cell_by_id, 116
gui, 117
gui_thread_id, 117
init, 116
number_of_cells_executing, 116
on_close, 116
on_fail, 116
on_message, 116
on_open, 116
our_connection_hdl, 117
port, 117
restart_kernel, 116
restarting_kernel, 117
run, 116
running_cells, 117
server_pid, 117
server_stderr, 117
server_stdout, 117
set_master, 116
stop, 116
terminate, 116
try_connect, 116
try_spawn_server, 117
wsclient, 117
cadabra::Coordinate, 119
name, 119
cadabra::DAntiSymmetric, 119
∼DAntiSymmetric, 120
get_tab, 120
name, 120
size, 120
cadabra::DataCell, 120
cell_type, 122
CellType, 121
DataCell, 122
document, 121
error, 121
hidden, 122
id, 122
image_png, 121
latex, 121
latex_view, 121
max_serial_number, 122
output, 121
python, 121
running, 122
sensitive, 122
serial_mutex, 123
serial_number, 123

textbuf, 123
verbatim, 121
cadabra::DataCell::id_t, 205
created_by_client, 206
id, 206
id_t, 206
operator<, 206
cadabra::Depends, 126
dependencies, 126
dependencies_, 127
name, 126
parse, 126
unnamed_argument, 126
cadabra::DependsBase, 127
dependencies, 127
cadabra::DependsInherit, 128
dependencies, 128
name, 128
cadabra::Derivative, 129
~Derivative, 129
get_tab, 129
name, 129
size, 130
value, 130
cadabra::Diagonal, 130
name, 131
cadabra::DifferentialForm, 131
degree, 131
degree_, 132
name, 131
parse, 132
cadabra::DifferentialFormBase, 132
degree, 132
cadabra::DiracBar, 133
name, 133
cadabra::DisplayBase, 133
dispatch, 134
DisplayBase, 134
kernel, 134
needs_brackets, 134
output, 134
tree, 134
cadabra::DisplaySympy, 135
children_have_brackets, 136
depsyms, 138
dispatch, 136
DisplaySympy, 136
import, 136
needs_brackets, 136
print_arrowlike, 136
print_children, 137
print_closing_bracket, 137
print_commalike, 137
print_components, 137
print_equalitylike, 137
print_fraclike, 137
print_intlike, 137
print_matrix, 137
print_multiplier, 137
print_opening_bracket, 137
print_other, 137
print_parent_rel, 137
print_partial, 137
print_powlike, 137
print_productlike, 137
print_sumlike, 137
symmap, 138
cadabra::DisplayTerminal, 138
bracket_level, 140
children_have_brackets, 139
dispatch, 139
DisplayTerminal, 139
needs_brackets, 139
print_arrowlike, 139
print_children, 140
print_closing_bracket, 140
print_commalike, 140
print_commutator, 140
print_components, 140
print_equalitylike, 140
print_fraclike, 140
print_intlike, 140
print_multiplier, 140
print_opening_bracket, 140
print_other, 140
print_parent_rel, 140
print_powlike, 140
print_productlike, 140
print_sumlike, 140
symmap, 140
cadabra::DisplayTeX, 141
bracket_level, 144
children_have_brackets, 142
curly_bracket_operators, 144
dispatch, 142
DisplayTeX, 142
extra_brackets_for_symbols, 144
latex_linefeeds, 144
latex_spacing, 144
needs_brackets, 142
print_arrowlike, 142
print_children, 143
print_closing_bracket, 143
print_commalike, 143
print_commutator, 143
print_components, 143
print_conditional, 143
print_equalitylike, 143
print_fraclike, 143
print_ftableau, 143
print_indexbracket, 143
print_intlike, 143
print_multiplier, 143
print_opening_bracket, 143
print_other, 143
print_parent_rel, 143

print_powlike, 143
print_productlike, 143
print_relation, 143
print_star, 144
print_sumlike, 144
print_tableau, 144
print_wedgeproduct, 144
reads_as_operator, 144
symmap, 144
texify, 144
tight_plus, 144
tight_star, 144
utf8_output, 144
cadabra::Distributable, 145
~Distributable, 145
name, 145
cadabra::DocumentThread, 147
ActionAddCell, 150
ActionEraseText, 150
ActionInsertText, 150
ActionPositionCursor, 150
ActionRemoveCell, 150
ActionSetRunStatus, 150
ActionSplitCell, 150
ActionStack, 148
algorithm, 148
build_visual_representation, 149
compute, 150
disable_stacks, 150
doc, 150
DocumentThread, 149
gui, 150
help_t, 148
help_type_and_topic, 149
is_registered, 149
latex, 148
load_from_string, 149
new_document, 149
none, 148
pending_actions, 151
process_action_queue, 149
property, 148
queue_action, 149
redo_stack, 151
registered, 151
set_compute_thread, 150
set_user_details, 150
stack_mutex, 151
undo, 150
undo_stack, 151
cadabra::EpsilonTensor, 160
~EpsilonTensor, 161
krdelta, 161
metric, 161
name, 161
parse, 161
cadabra::Ex, 165
arg, 168
arg_size, 168
arg_to_num, 168
calc_hash, 168
changed_state, 168
equation_by_name, 168
equation_by_number, 168
equation_by_number_or_name, 168
equation_label, 168
equation_number, 168
equation_number_or_name, 168
erase_expression, 168
Ex, 167
flatten_and_erase, 168
history, 170
history_size, 168
is_rational, 169
l_applied, 167
l_checkpointed, 167
l_error, 167
l_no_action, 167
list_unwrap_single_element, 169
list_wrap_single_element, 169
move_index, 169
named_parent, 169
number_of_equations, 169
number_of_steps, 169
operator==, 169
pop_history, 169
print_entire_tree, 169
print_python, 169
print_recursive_treeform, 170
procedure_by_name, 170
push_history, 170
replace_index, 170
reset_state, 170
result_t, 167
selectors, 170
state, 170
state_, 170
to_rational, 170
update_state, 170
cadabra::Ex_comparator, 171
always, 173
can_move_adjacent, 173
can_swap, 173
can_swap_ilist_ilist, 173
can_swap_prod_obj, 174
can_swap_prod_prod, 174
can_swap_prod_sum, 174
can_swap_sum_obj, 174
can_swap_sum_sum, 174
clear, 174
compare, 174
equal_subtree, 174
Ex_comparator, 173
factor_locations, 175
factor_moving_signs, 175
index_value_map, 175

lhs_contains_dummies, 175
match_index_greater, 173
match_index_less, 173
match_subproduct, 174
match_subsum, 174
match_t, 173
never, 173
no_match_greater, 173
no_match_indexpos_greater, 173
no_match_indexpos_less, 173
no_match_less, 173
node_match, 173
not_at_top, 173
offset, 175
properties, 176
replacement_map, 176
replacement_map_t, 172
report, 175
satisfies_conditions, 175
should_swap, 175
subtree_match, 173
subtree_replacement_map, 176
subtree_replacement_map_t, 172
tab, 175
term_ratio, 176
useprops_t, 173
cadabra::Ex_is_equivalent, 176
Ex_is_equivalent, 176
operator(), 176
properties, 176
cadabra::Ex_is_less, 177
Ex_is_less, 177
mod_prel, 177
operator(), 177
properties, 177
cadabra::ExteriorDerivative, 187
degree, 187
name, 187
cadabra::FilledTableau, 197
∼FilledTableau, 197
dimension, 197
name, 197
parse, 197
cadabra::GUIBase, 203
add_cell, 203
get_cursor_position, 203
on_connect, 204
on_disconnect, 204
on_kernel_runstatus, 204
on_network_error, 204
position_cursor, 204
process_data, 204
remove_all_cells, 204
remove_cell, 204
update_cell, 205
cadabra::GammaMatrix, 201
latex, 201
metric, 202
name, 201
parse, 201
cadabra::GammaTraceless, 202
∼GammaTraceless, 202
name, 202
cadabra::ImageView, 208
∼ImageView, 208
image, 208
ImageView, 208
set_image_from_base64, 208
cadabra::ImaginaryI, 209
name, 209
cadabra::ImplicitIndex, 209
latex, 210
name, 210
parse, 210
set_names, 211
unnamed_argument, 210
cadabra::IndexInherit, 215
name, 216
cadabra::Indices, 217
collect_index_values, 218
equals, 218
fixed, 218
free, 218
independent, 218
Indices, 218
latex, 218
name, 219
parent_name, 219
parse, 219
position_t, 218
position_type, 219
set_name, 219
unnamed_argument, 219
values, 219
cadabra::Inherit
∼Inherit, 220
name, 220
cadabra::Inherit< T >, 219
cadabra::Integer, 220
∼Integer, 221
difference, 221
display, 221
from, 221
name, 221
parse, 221
to, 221
unnamed_argument, 221
cadabra::InverseMetric, 225
InverseMetric, 225
name, 225
parse, 225
signature, 226
validate, 226
cadabra::Kernel, 233
∼Kernel, 233
ex_from_string, 233

inject_property, 233
Kernel, 233
properties, 234
cadabra::KroneckerDelta, 235
~KroneckerDelta, 236
get_tab, 236
name, 236
size, 236
cadabra::LaTeXForm, 237
latex_, 238
latex_form, 238
name, 238
parse, 238
unnamed_argument, 238
cadabra::Matrix, 244
~Matrix, 245
name, 245
cadabra::Metric, 245
latex, 246
Metric, 246
name, 246
parse, 246
signature, 246
validate, 246
cadabra::NonCommuting, 249
name, 250
sign, 250
cadabra::NotebookCanvas, 251
~NotebookCanvas, 252
bottomline, 252
ebox, 252
NotebookCanvas, 252
refresh_all, 252
scroll, 252
visualcells, 252
cadabra::NotebookWindow, 255
~NotebookWindow, 257
actiongroup, 262
add_cell, 258
canvasses, 262
cdbapp, 262
cell_content_changed, 258
cell_content_erase, 258
cell_content_execute, 258
cell_content_insert, 258
cell_got_focus, 258
cell_toggle_visibility, 258
crash_window_hidden, 262
css_provider, 262
current_canvas, 262
current_cell, 262
dim_output_cells, 258
dispatcher, 262
engine, 262
follow_cell, 262
get_cursor_position, 258
idle_handler, 258
is_configured, 262
kernel_label, 262
kernel_spinner, 262
kernel_spinner_status, 262
kernel_string, 262
last_configure_width, 262
load_file, 258
mainbox, 262
modified, 262
name, 262
NotebookWindow, 257
on_configure_event, 258
on_connect, 258
on_crash_window_closed, 258
on_delete_event, 259
on_disconnect, 259
on_edit_cell_is_latex, 259
on_edit_cell_is_python, 259
on_edit_delete, 259
on_edit_insert_above, 259
on_edit_insert_below, 259
on_edit_split, 259
on_edit_undo, 259
on_file_close, 259
on_file_export_html, 259
on_file_export_html_segment, 259
on_file_export_latex, 259
on_file_export_python, 259
on_file_new, 259
on_file_open, 259
on_file_quit, 259
on_file_save, 259
on_file_save_as, 259
on_help, 259
on_help_about, 259
on_kernel_restart, 259
on_kernel_runstatus, 259
on_key_press_event, 259
on_network_error, 260
on_prefs_font_size, 260
on_run_cell, 260
on_run_runall, 260
on_run_runtocursor, 260
on_run_stop, 260
on_scroll, 260
on_scroll_size_allocate, 260
on_tex_error, 260
on_text_scaling_factor_changed, 260
on_view_close, 260
on_view_split, 260
on_vscroll_changed, 260
position_cursor, 260
prefs, 262
process_data, 260
process_todo_queue, 260
progressbar, 262
quit_safeguard, 261
read_only, 262
remove_all_cells, 261

remove_cell, 261
save, 261
scale, 263
scroll_current_cell_into_view, 261
set_name, 261
set_stop_sensitive, 261
set_title_prefix, 261
settings, 263
setup_css_provider, 261
status_label, 263
status_mutex, 263
status_string, 263
statusbarbox, 263
supermainbox, 263
title_prefix, 263
topbox, 263
uimanager, 263
update_cell, 261
update_title, 261
cadabra::NotebookWindow::Prefs, 274
font_step, 274
Prefs, 274
cadabra::NumericalFlat, 265
name, 265
cadabra::Parser, 267
advance, 269
current_bracket, 269
current_mode, 269
current_parent_rel, 269
erase, 269
finalise, 269
get_token, 269
is_closing_bracket, 269
is_link, 269
is_number, 269
is_opening_bracket, 269
m_backslashname, 268
m_childgroup, 268
m_findchildren, 268
m_initialgroup, 268
m_name, 268
m_property, 268
m_singlecharname, 268
m_skipwhite, 268
m_verbatim, 268
mode_t, 268
Parser, 269
parts, 269
remove_empty_nodes, 269
str, 269
string2tree, 269
tree, 269
cadabra::PartialDerivative, 270
~PartialDerivative, 270
get_tab, 270
name, 270
size, 271
cadabra::Properties, 281
clear, 282
get, 282
get_composite, 282, 283
has, 283
head, 283
insert_list_prop, 283
insert_prop, 283
master_insert, 283
pat_prop_pair_t, 282
pats, 283
pattern_map_t, 282
property_map_t, 282
props, 283
register_property, 283
registered_properties, 283
serial_number, 283
cadabra::Properties::registered_property_map_t, 291
~registered_property_map_t, 291
internal_property_map_t, 291
iterator, 291
store, 291
cadabra::PropertyInherit, 289
name, 289
cadabra::RiemannTensor, 295
name, 296
RiemannTensor, 296
validate, 296
cadabra::SatisfiesBianchi, 297
get_tab, 297
name, 297
size, 297
cadabra::SelfAntiCommuting, 298
name, 298
sign, 298
cadabra::SelfCommuting, 299
name, 299
sign, 299
cadabra::SelfCommutingBehaviour, 299
sign, 300
cadabra::SelfNonCommuting, 300
name, 300
sign, 300
cadabra::SortOrder, 318
equals, 318
name, 318
cadabra::Spinor, 319
~Spinor, 319
chirality, 320
dimension, 320
majorana, 320
name, 319
negative, 319
parse, 319
positive, 319
Spinor, 319
weyl, 320
cadabra::Symbol, 332
get, 332

name, 332
cadabra::Symmetric, 335
get_tab, 335
name, 335
size, 335
cadabra::Tableau, 340
~Tableau, 341
dimension, 341
name, 341
parse, 341
cadabra::TableauBase, 345
~TableauBase, 346
get_indexgroup, 346
get_tab, 346
is_simple_symmetry, 346
only_column_exchange, 346
size, 346
tab_t, 346
cadabra::TableauSymmetry, 346
~TableauSymmetry, 347
get_tab, 347
latex, 347
name, 347
only_col_, 348
only_column_exchange, 347
parse, 348
size, 348
tabs, 348
cadabra::TeXEngine, 352
~TeXEngine, 353
checkin, 353
checkout, 353
checkout_all, 353
convert_all, 353
convert_one, 353
convert_set, 353
erase_file, 353
font_size_, 354
handle_latex_errors, 353
horizontal_pixels_, 354
invalidate_all, 353
latex_packages, 354
millimeter_per_inch, 354
modify, 353
preamble_string, 354
requests, 354
scale_, 354
set_font_size, 354
set_geometry, 354
set_scale, 354
TeXEngine, 353
cadabra::TeXEngine::TeXException, 354
TeXException, 355
cadabra::TeXEngine::TeXRequest, 355
end_wrap, 356
height, 356
height_, 356
image, 356
image_, 356
latex_string, 356
needs_generating, 356
start_wrap, 356
TeXEngine, 356
TeXRequest, 356
width, 356
width_, 356
cadabra::TeXView, 356
~TeXView, 357
content, 358
convert, 357
datacell, 358
dim, 357
engine, 358
hbox, 358
image, 358
on_button_release_event, 357
on_show, 358
pixbuf, 358
show_hide_requested, 358
TeXView, 357
tex_error, 358
update_image, 358
vbox, 358
cadabra::TerminalStream, 351
kernel, 351
operator<<, 351
out_, 351
TerminalStream, 351
cadabra::Traceless, 360
~Traceless, 360
name, 360
cadabra::VisualCell, 370
document, 370
imagebox, 371
inbox, 371
outbox, 371
cadabra::Weight, 371
name, 371
parse, 371
unnamed_argument, 372
value, 372
value_, 372
cadabra::WeightBase, 372
value, 373
cadabra::WeightInherit, 374
additive, 374
combination_type, 375
multiplicative, 374
name, 375
parse, 375
power, 374
unnamed_argument, 375
value, 375
value_self, 375
cadabra::WeightInherit::WeightException, 373
WeightException, 373

cadabra::WeylTensor, 375
latex, 376
name, 376
validate, 376
WeylTensor, 376
cadabra::canonicalise, 94
apply, 94
can_apply, 94
canonicalise, 94
generating_set, 95
get_index_set_name, 95
only_one_on_derivative, 95
position_type, 95
remove_traceless_traces, 95
remove_vanishing_numericals, 95
reuse_generating_set, 95
cadabra::collect_components, 99
apply, 99
can_apply, 99
collect_components, 99
cadabra::collect_factors, 100
apply, 100
can_apply, 100
collect_factors, 100
factor_hash, 101
factor_hash_iterator_t, 100
factor_hash_t, 100
fill_hash_map, 101
cadabra::collect_terms, 101
apply, 102
can_apply, 102
collect_from_hash_map, 102
collect_terms, 102
fill_hash_map, 102
remove_zeroed_terms, 102
term_hash, 102
term_hash_iterator_t, 102
term_hash_t, 102
cadabra::combine, 108
apply, 109
can_apply, 109
combine, 109
iloc, 109
indexlocmap_t, 109
cadabra::complete, 112
apply, 113
can_apply, 113
complete, 113
goal, 113
cadabra::decompose_product, 123
apply, 124
asym_ranges, 125
can_apply, 124
decompose_product, 124
dim, 125
f1, 125
f2, 125
fill_asym_ranges, 125
ind1, 125
ind2, 125
indices_equivalent, 125
nt1, 125
nt2, 125
numtab_t, 124
numtabs_t, 124
project_onto_initial_symmetries, 125
sibtab_t, 124
sibtabs_t, 124
t1, 125
t1tab, 125
t2, 125
t2tab, 125
cadabra::distribute, 146
apply, 146
can_apply, 146
distribute, 146
cadabra::drop_keep_weight, 151
arg, 152
can_apply, 152
do_apply, 152
drop_keep_weight, 152
gmn, 152
label, 152
weight, 152
wgh, 152
cadabra::drop_weight, 153
apply, 153
drop_weight, 153
cadabra::einsteinify, 154
apply, 155
can_apply, 155
einsteinify, 155
metric, 155
cadabra::eliminate_converter, 155
apply, 156
can_apply, 156
eliminate_converter, 156
handle_one_index, 156
ind_dummy, 157
ind_free, 157
is_conversion_object, 156
preferred, 157
cadabra::eliminate_kronecker, 157
apply, 157
can_apply, 157
eliminate_kronecker, 157
cadabra::eliminate_metric, 158
eliminate_metric, 158
is_conversion_object, 158
cadabra::epsilon_to_delta, 159
apply, 159
can_apply, 159
epsilon_to_delta, 159
epsilons, 160
reduce, 160
repdelta, 160

signature, 160
cadabra::evaluate, 162
apply, 163
can_apply, 163
cleanup_components, 164
components, 165
dense_factor, 164
dependencies, 164
evaluate, 163
handle_components, 164
handle_derivative, 164
handle_epsilon, 164
handle_factor, 164
handle_prod, 164
handle_sum, 164
is_component, 164
merge_component_children, 164
merge_components, 164
only_rhs, 165
simplify_components, 165
wrap_scalar_in_components_node, 165
cadabra::exchange, 177
collect_identical_tensors, 178
get_node_gs, 178
possible_singlets, 178
cadabra::exchange::identical_tensors_t, 206
comm, 207
extra_sign, 207
gammatraceless, 207
number_of_indices, 207
seq_numbers_of_first_indices, 207
spino, 207
tab, 207
tensors, 207
traceless, 207
cadabra::exchange::tensor_type_t, 350
name, 351
number_of_indices, 351
cadabra::expand, 183
apply, 183
can_apply, 183
expand, 183
ii_first, 184
ii_last, 184
mx_first, 184
mx_last, 184
one_index, 184
cadabra::expand_delta, 184
apply, 184
can_apply, 184
expand_delta, 184
cadabra::expand_diracbar, 185
apply, 185
can_apply, 185
expand_diracbar, 185
cadabra::expand_power, 186
apply, 186
can_apply, 186
expand_power, 186
cadabra::factor_in, 188
apply, 188
calc_restricted_hash, 188
can_apply, 189
compare_prod_nonprod, 189
compare_restricted, 189
factnodes, 189
factor_in, 188
factors, 189
fill_hash_map, 189
term_hash, 189
term_hash_iterator_t, 188
term_hash_t, 188
cadabra::factor_out, 189
apply, 190
can_apply, 190
factor_out, 190
to_factor_out, 190
to_factor_out_t, 190
to_right, 190
cadabra::fierz, 191
apply, 191
can_apply, 191
dim, 192
fierz, 191
gam1, 192
gam2, 192
indprop, 192
prop1, 192
prop2, 192
prop3, 192
prop4, 192
spin1, 192
spin2, 192
spin3, 192
spin4, 192
spinor_list, 192
spinordim, 192
cadabra::flatten_product, 198
apply, 199
can_apply, 199
flatten_product, 199
is_diff, 199
make_consistent_only, 199
cadabra::flatten_sum, 199
apply, 200
can_apply, 200
flatten_sum, 200
make_consistent_only, 200
cadabra::index_iterator, 213
begin, 214
create, 214
end, 214
halt, 215
index_iterator, 214
is_index, 214
operator!=, 215

operator++, 215
operator+=, 215
operator=, 215
operator==, 215
properties, 215
roof, 215
walk, 215
cadabra::indexsort, 216
apply, 217
can_apply, 217
indexsort, 217
tb, 217
cadabra::indexsort::less_indexed_treenode, 238
it, 239
kernel, 239
less_indexed_treenode, 239
operator(), 239
tr, 239
cadabra::integrate_by_parts, 222
apply, 223
away_from, 223
can_apply, 223
derivative_acting_on_arg, 223
handle_term, 223
int_and_derivative_related, 223
integrate_by_parts, 223
wrap, 223
cadabra::join_gamma, 229
append_prod_, 230
apply, 230
can_apply, 230
expand, 231
gamma_bracket_, 231
gamma_name_, 231
gm1, 231
gm2, 231
join_gamma, 230
only_expand, 231
regroup_indices_, 230
use_generalised_delta_, 231
cadabra::keep_terms, 231
apply, 232
can_apply, 232
keep_terms, 232
terms_, 232
cadabra::keep_weight, 232
apply, 233
keep_weight, 233
cadabra::keyval_t, 234
begin, 235
const_iterator, 235
end, 235
erase, 235
find, 235
iterator, 235
keyvals, 235
kvlist_t, 235
kvpair_t, 235
push_back, 235
value_type, 235
cadabra::labelled_property, 236
label, 237
parse, 237
cadabra::list_property, 239
cadabra::lr_tensor, 242
apply, 242
can_apply, 242
do_filledtableau, 242
do_tableau, 243
lr_tensor, 242
tab1, 243
tab2, 243
cadabra::map_sympy, 243
apply, 244
can_apply, 244
head_, 244
index_factors, 244
left, 244
map_sympy, 244
cadabra::nset_it_less, 264
operator(), 264
cadabra::order, 265
anticomm, 266
apply, 266
can_apply, 266
objects, 266
order, 266
cadabra::pattern, 271
children_wildcard, 271
match, 271
obj, 271
pattern, 271
cadabra::product_rule, 278
apply, 279
can_apply, 279
number_of_indices, 279
prodnode, 279
product_rule, 279
cadabra::property, 286
~property, 287
equals, 287
exact_match, 287
hidden, 287, 288
hidden_, 289
id_match, 287
latex, 288
match_t, 287
name, 288
no_match, 287
parse, 288
parse_one_argument, 288
parse_to_keyvals, 288
property, 287
unnamed_argument, 288
validate, 288
cadabra::reduce_delta, 290

apply, 290
can_apply, 290
one_step_, 290
reduce_delta, 290
cadabra::rename_dummies, 292
apply, 293
can_apply, 293
dset1, 293
dset2, 293
rename_dummies, 293
cadabra::replace_match, 293
apply, 294
can_apply, 294
replace_match, 294
cadabra::rewrite_indices, 294
apply, 295
can_apply, 295
converters, 295
is_derivative_argument, 295
preferred, 295
rewrite_indices, 295
cadabra::sort_product, 315
apply, 315
can_apply, 315
ignore_numbers_, 316
sort_product, 315
cadabra::sort_spinors, 316
apply, 316
can_apply, 316
gammamat, 317
one, 317
sort_spinors, 316
two, 317
cadabra::sort_sum, 317
apply, 317
can_apply, 317
should_swap, 318
sort_sum, 317
cadabra::split_gamma, 321
apply, 321
can_apply, 321
on_back, 322
split_gamma, 321
cadabra::split_index, 322
apply, 323
can_apply, 323
full_class, 323
num1, 323
num2, 323
part1_class, 323
part1_coord, 323
part1_coord_node, 323
part1_is_number, 323
part2_class, 323
part2_coord, 323
part2_coord_node, 323
part2_is_number, 323
split_index, 323
cadabra::str_node, 325
b_curly, 326
b_invalid, 326
b_no, 326
b_none, 326
b_pointy, 326
b_round, 326
b_square, 326
bracket_t, 326
compare_name_brack_par, 327
compare_name_inverse_par, 327
compare_names_only, 327
fl, 328
flip_parent_rel, 327
is_autodeclare_wildcard, 327
is_command, 328
is_identity, 328
is_index, 328
is_indexplus_wildcard, 328
is_indexstar_wildcard, 328
is_inert_command, 328
is_integer, 328
is_name_wildcard, 328
is_numbered_symbol, 328
is_object_wildcard, 328
is_quoted_string, 328
is_range_wildcard, 328
is_rational, 328
is_siblings_wildcard, 328
is_unsimplified_integer, 328
is_unsimplified_rational, 328
is_zero, 328
multiplier, 328
name, 328
name_only, 328
operator<, 328
operator==, 328
p_components, 327
p_exponent, 327
p_none, 327
p_property, 327
p_sub, 327
p_super, 327
parent_rel_t, 326
str_node, 327
cadabra::str_node::flag_t, 198
bracket, 198
keep_after_eval, 198
line_per_node, 198
parent_rel, 198
cadabra::substitute, 329
apply, 329
args, 330
can_apply, 329
comparator, 330
conditions, 330
lhs_contains_dummies, 330
rhs_contains_dummies, 330

sort_product_, 330
substitute, 329
use_rule, 330
cadabra::sym, 330
apply, 331
argloc_2_treeloc, 331
can_apply, 331
doit, 331
objects, 331
raw_ints, 331
sign, 331
sym, 331
cadabra::symbols, 54
greek, 55
cadabra::tab_basics, 339
find_obj, 340
num_to_it, 340
tab_basics, 340
tabs_to_singlet_rules, 340
tabs_to_tree, 340
tree_to_numerical_tab, 340
uinttab_t, 340
uinttabs_t, 340
cadabra::take_match, 349
apply, 350
can_apply, 350
rules, 350
take_match, 350
cadabra::tree_equal_obj, 360
operator(), 361
properties, 361
tree_equal_obj, 361
cadabra::tree_exact_equal_mod_prel_obj, 361
operator(), 362
properties, 362
tree_exact_equal_mod_prel_obj, 362
cadabra::tree_exact_equal_obj, 362
operator(), 362
properties, 362
tree_exact_equal_obj, 362
cadabra::tree_exact_less_for_indexmap_obj, 363
operator(), 363
cadabra::tree_exact_less_mod_prel_obj, 363
operator(), 364
properties, 364
tree_exact_less_mod_prel_obj, 364
cadabra::tree_exact_less_no_wildcards_mod_prel_obj, 364
operator(), 364
properties, 364
tree_exact_less_no_wildcards_mod_prel_obj, 364
cadabra::tree_exact_less_no_wildcards_obj, 365
operator(), 365
properties, 365
tree_exact_less_no_wildcards_obj, 365
cadabra::tree_exact_less_obj, 365
operator(), 366
properties, 366
tree_exact_less_obj, 366
cadabra::tree_less_modprel_obj, 366
operator(), 367
properties, 367
tree_less_modprel_obj, 367
cadabra::tree_less_obj, 367
operator(), 367
properties, 367
tree_less_obj, 367
cadabra::unwrap, 368
apply, 368
can_apply, 368
unwrap, 368
wrappers, 369
cadabra::vary, 369
apply, 369
args, 370
can_apply, 369
vary, 369
cadabra::young_project, 377
apply, 378
asym_ranges, 378
can_apply, 378
name_tab_t, 378
nametab, 378
nth_index_node, 378
pos_tab_t, 378
remove_traces, 378
sym, 378
tab, 378
young_project, 378
cadabra::young_project_product, 379
apply, 379
can_apply, 379
young_project_product, 379
cadabra::young_project_tensor, 379
apply, 380
can_apply, 380
modulo_monoterm, 380
sym, 380
tb, 380
young_project_tensor, 380
cadabra_osx, 55
cadabra_osx::NotebookWindow, 253
add_cell, 254
canvasses, 254
controller, 254
current_canvas, 254
NotebookWindow, 254
on_connect, 254
on_disconnect, 254
on_network_error, 254
position_cursor, 254
process_data, 254
remove_all_cells, 254
remove_cell, 254
update_cell, 254
CadabraException, 93

CadabraException, 93
calc_hash
cadabra::Ex, 168
calc_restricted_hash
cadabra::factor_in, 188
call_count
ProgressMonitor::Total, 359
call_mutex
snoop::SnoopImpl, 314
call_post_process
Python Core, 32
call_stack
ProgressMonitor, 280
call_totals
ProgressMonitor, 280
can_apply
cadabra::Algorithm, 80
cadabra::canonicalise, 94
cadabra::collect_components, 99
cadabra::collect_factors, 100
cadabra::collect_terms, 102
cadabra::combine, 109
cadabra::complete, 113
cadabra::decompose_product, 124
cadabra::distribute, 146
cadabra::drop_keep_weight, 152
cadabra::einsteinify, 155
cadabra::eliminate_converter, 156
cadabra::eliminate_kronecker, 157
cadabra::epsilon_to_delta, 159
cadabra::evaluate, 163
cadabra::expand, 183
cadabra::expand_delta, 184
cadabra::expand_diracbar, 185
cadabra::expand_power, 186
cadabra::factor_in, 189
cadabra::factor_out, 190
cadabra::fierz, 191
cadabra::flatten_product, 199
cadabra::flatten_sum, 200
cadabra::indexsort, 217
cadabra::integrate_by_parts, 223
cadabra::join_gamma, 230
cadabra::keep_terms, 232
cadabra::lr_tensor, 242
cadabra::map_sympy, 244
cadabra::order, 266
cadabra::product_rule, 279
cadabra::reduce_delta, 290
cadabra::rename_dummies, 293
cadabra::replace_match, 294
cadabra::rewrite_indices, 295
cadabra::sort_product, 315
cadabra::sort_spinors, 316
cadabra::sort_sum, 317
cadabra::split_gamma, 321
cadabra::split_index, 323
cadabra::substitute, 329
cadabra::sym, 331
cadabra::take_match, 350
cadabra::unwrap, 368
cadabra::vary, 369
cadabra::young_project, 378
cadabra::young_project_product, 379
cadabra::young_project_tensor, 380
component, 114
rename_dummies, 292
split, 321
can_move_adjacent
cadabra::Ex_comparator, 173
can_swap
cadabra::Ex_comparator, 173
can_swap_ilist_ilist
cadabra::Ex_comparator, 173
can_swap_prod_obj
cadabra::Ex_comparator, 174
can_swap_prod_prod
cadabra::Ex_comparator, 174
can_swap_prod_sum
cadabra::Ex_comparator, 174
can_swap_sum_obj
cadabra::Ex_comparator, 174
can_swap_sum_sum
cadabra::Ex_comparator, 174
canonicalise
cadabra::canonicalise, 94
yngtab::filled_tableau, 195
canvasses
cadabra::NotebookWindow, 262
cadabra_osx::NotebookWindow, 254
catchErr
Server, 306
catchOut
Server, 306
CatchOutput
Server::CatchOutput, 96
cdb2python
cadabra, 49
cdbapp
cadabra::NotebookWindow, 262
cell_content_changed
cadabra::NotebookWindow, 258
cell_content_erase
cadabra::NotebookWindow, 258
cell_content_execute
cadabra::NotebookWindow, 258
cell_content_insert
cadabra::NotebookWindow, 258
cell_got_focus
cadabra::CodeInput::exp_input_tv, 182
cadabra::NotebookWindow, 258
cell_id
Server::Block, 91
cell_toggle_visibility
cadabra::NotebookWindow, 258
cell_type

cadabra::DataCell, 122
CellType
cadabra::DataCell, 121
cells_ran
Server, 306
changed_state
cadabra::Ex, 168
check_consistency
cadabra::Algorithm, 80
check_degree_consistency
cadabra::Algorithm, 80
check_index_consistency
cadabra, 49
cadabra::Algorithm, 80
checkin
cadabra::TeXEngine, 353
checkout
cadabra::TeXEngine, 353
checkout_all
cadabra::TeXEngine, 353
checkpoint_
Stopwatch, 324
child
cadabra::ActionAddCell, 63
child_num
cadabra::ActionAddCell, 64
children_have_brackets
cadabra::DisplaySympy, 136
cadabra::DisplayTerminal, 139
cadabra::DisplayTeX, 142
children_wildcard
cadabra::pattern, 271
chirality
cadabra::Spinor, 320
classify_add_index
cadabra::Algorithm, 80
classify_indices
cadabra::Algorithm, 80
classify_indices_up
cadabra::Algorithm, 80
Cleanup, 28
cleanup_components
cadabra, 49
cadabra::evaluate, 164
cleanup_derivative
cadabra, 50
cleanup_diagonal
cadabra, 50
cleanup_dispatch
cadabra, 50
cleanup_dispatch_deep
cadabra, 50
cleanup_exterior_derivative
cadabra, 50
cleanup_frac
cadabra, 50
cleanup_indexbracket
cadabra, 50
cleanup_kronecker
cadabra, 50
cleanup_numericalflat
cadabra, 50
cleanup_partialderivative
cadabra, 50
cleanup_productlike
cadabra, 50
cleanup_rational
cadabra, 50
cleanup_sub
cadabra, 51
cleanup_sumlike
cadabra, 51
cleanup_updown
cadabra, 51
clear
cadabra::Ex_comparator, 174
cadabra::Properties, 282
combin::combinations, 104
combin::combinations_base, 106
combin::symm_helper, 333
combin::symm_val_helper, 334
combin::symmetriser, 337
Server::CatchOutput, 96
yngtab::filled_tableau, 195
yngtab::tableau, 342
yngtab::tableau_base, 344
clear_results
combin::combinations, 104
combin::combinations_base, 106
client
ComputeThread.cc, 385
test_client.cc, 394
test_talk_to_server.cc, 395
Client-Server, 37
client_log_id
snoop::Snoop::LogEntry, 241
client_server/Actions.cc, 383
client_server/Actions.hh, 383
client_server/CMakeLists.txt, 426
cmake_minimum_required, 426
client_server/ComputeThread.cc, 385
client_server/ComputeThread.hh, 385
client_server/DataCell.cc, 386
client_server/DataCell.hh, 386
client_server/DocumentThread.cc, 387
client_server/DocumentThread.hh, 388
client_server/GUIBase.hh, 388
client_server/Server.cc, 390
client_server/Server.hh, 390
client_server/Snoop.cc, 391
client_server/Snoop.hh, 392
client_server/SnoopPrivate.hh, 393
client_server/cadabra-server.cc, 384
client_server/cadabra2cadabra.cc, 384
client_server/cadabra2html.cc, 384
client_server/popen2.cc, 388

client_server/popen2.hh, 389
client_server/regexp_tester.cc, 389
client_server/test_client.cc, 393
client_server/test_talk_to_server.cc, 394
close
    exec_stream_t, 180
close_brackets
    preprocessor, 277
close_in
    exec_stream_t, 180
cmake_minimum_required
    client_server/CMakeLists.txt, 426
    core/CMakeLists.txt, 426
    frontend/common/CMakeLists.txt, 426
CodeInput
    cadabra::CodeInput, 97
    cadabra::CodeInput::exp_input_tv, 182
CodeInput.cc
    trim, 488
collect
    combin::symmetriser, 337
    Server::CatchOutput, 96
collect_components
    cadabra::collect_components, 99
collect_factors
    cadabra::collect_factors, 100
collect_from_hash_map
    cadabra::collect_terms, 102
collect_identical_tensors
    cadabra::exchange, 178
collect_index_values
    cadabra::Indices, 218
collect_terms
    cadabra::collect_terms, 102
column_number
    yngtab::filled_tableau::in_column_iterator, 213
    yngtab::filled_tableau::iterator, 228
column_size
    yngtab::tableau_base, 344
combin, 55
    determine_intersection_ranges, 56
    fact, 56
    factorial, 56
    hash, 56
    operator<<, 56
    operator==, 56
    ordersign, 56
    range_t, 56
    range_vector_t, 56
    vector_prod, 56
    vector_prod_fact, 56
    vector_sum, 56
    weights_t, 56
combin::combinations
    ∼combinations, 104
    clear, 104
    clear_results, 104
    combinations, 104
    const_iterator, 104
    multiplier, 104
    operator[], 104
    ordersign, 104
    permuted_sets_t, 104
    size, 104
    storage, 104
    vector_generated, 104
combin::combinations< T >, 103
combin::combinations_base
    ∼combinations_base, 106
    block_length, 107
    clear, 106
    clear_results, 106
    combinations_base, 106
    current_weight, 107
    end_, 107
    entry_accepted, 106
    final_weight_constraints_check, 107
    input_asym, 107
    is_allowed_by_weight_constraints, 107
    max_weights, 107
    multiple_pick, 107
    multiplier, 107
    nextstep, 107
    original, 108
    permute, 107
    restore_weights, 107
    set_unit_sublengths, 107
    start_, 108
    sub_problem_blocksize, 108
    sublengths, 108
    sum_of_sublengths, 107
    temparr, 108
    total_permutations, 107
    update_weights, 107
    vector_generated, 107
    vector_generated_called_, 108
    weight_cond, 106
    weight_conditions, 108
    weight_equals, 106
    weight_greater, 106
    weight_less, 106
    weights, 108
combin::combinations_base< T >, 105
combin::symm_helper
    clear, 333
    current_multiplicity, 333
    first_one, 333
    owner_, 333
    symm_helper, 333
    vector_generated, 333
combin::symm_helper< T >, 332
combin::symm_val_helper
    clear, 334
    current_multiplicity, 335
    first_one, 335
    owner_, 335

symm_val_helper, 334
vector_generated, 334
combin::symm_val_helper< T >, 334
combin::symmetriser
apply_symmetry, 337
block_length, 338
clear, 337
collect, 337
current_, 338
input_asym, 338
multiplicity, 338
operator[], 337
original, 338
originals, 338
permutation_sign, 338
permute_blocks, 338
remove_multiplicity_zero, 337
set_multiplicity, 337
sh_, 338
signature, 337
size, 337
sublengths, 338
sublengths_scattered, 338
svh_, 338
symm_helper< T >, 338
symm_val_helper< T >, 338
symmetriser, 337
value_permute, 338
values_to_locations, 337
combin::symmetriser< T >, 336
combination_type
cadabra::WeightInherit, 375
combinations
combin::combinations, 104
combinations_base
combin::combinations_base, 106
combine
cadabra::combine, 109
comm
cadabra::exchange::identical_tensors_t, 207
comparator
cadabra::substitute, 330
compare
cadabra::Ex_comparator, 174
Compare.cc
DEBUG, 429
operator<, 429
operator<<, 429
Compare.hh
operator<, 430
operator<<, 430
compare_
cadabra::Algorithm, 80
compare_name_brack_par
cadabra::str_node, 327
compare_name_inverse_par
cadabra::str_node, 327
compare_names_only
cadabra::str_node, 327
compare_prod_nonprod
cadabra::factor_in, 189
compare_restricted
cadabra::factor_in, 189
compare_without_multiplicity
yngtab::filled_tableau, 195
complete
cadabra::complete, 113
component, 113
apply, 114
can_apply, 114
component, 114
components
cadabra::evaluate, 165
compose
exec_stream_t::error_t, 162
compute
Cadabra, 92
cadabra::DocumentThread, 150
NotebookController.mm, 495
compute_thread
Cadabra, 92
NotebookController.mm, 495
ComputeThread
cadabra::ComputeThread, 115
ComputeThread.cc
client, 385
SlotSpawnChildSetup, 385
ComputeThread.hh
message_ptr, 386
scoped_lock, 386
WSClient, 386
conditions
cadabra::substitute, 330
connected
snoop::Snoop::AppEntry, 88
Connection
Server::Connection, 118
connection
cadabra::ComputeThread, 117
snoop::SnoopImpl, 314
connection_attempt_failed
snoop::SnoopImpl, 314
connection_cv
snoop::SnoopImpl, 314
connection_is_open
cadabra::ComputeThread, 117
snoop::SnoopImpl, 314
connection_mutex
snoop::SnoopImpl, 314
ConnectionMap
Server, 304
connections
Server, 306
ConsistencyException, 118
ConsistencyException, 118
ConsistencyExceptionType

PythonCdb.cc, 475
const_iterator
cadabra::keyval_t, 235
combin::combinations, 104
construct
iterable_converter, 226
construct_Ex_from_int
PythonCdb.cc, 473
construct_Ex_from_int_2
PythonCdb.cc, 473
construct_Ex_from_string
PythonCdb.cc, 473
construct_Ex_from_string_2
PythonCdb.cc, 474
contains
cadabra::Algorithm, 80
content
cadabra::TeXView, 358
content_changed
cadabra::CodeInput::exp_input_tv, 182
content_erase
cadabra::CodeInput::exp_input_tv, 182
content_execute
cadabra::CodeInput::exp_input_tv, 182
content_insert
cadabra::CodeInput::exp_input_tv, 182
controller
cadabra_osx::NotebookWindow, 254
convert
cadabra::TeXView, 357
convert_all
cadabra::TeXEngine, 353
convert_line
cadabra, 51
convert_one
cadabra::TeXEngine, 353
convert_set
cadabra::TeXEngine, 353
converters
cadabra::rewrite_indices, 295
convertible
iterable_converter, 226
copy_shape
yngtab::filled_tableau, 195
Core, 25
do_list, 26
do_subtree, 26
find_in_list, 27
find_in_subtree, 27
is_in, 27
make_list, 27
multiply, 27
core/Algorithm.cc, 395
core/Algorithm.hh, 395
core/Bridge.cc, 422
core/Bridge.hh, 422
core/CMakeLists.txt, 425
cmake_minimum_required, 426
endif, 426
install, 426
print, 426
core/CdbPython.cc, 423
core/CdbPython.hh, 423
core/Cleanup.cc, 424
core/Cleanup.hh, 425
core/Combinatorics.cc, 426
core/Combinatorics.hh, 427
core/Compare.cc, 428
core/Compare.hh, 429
core/DisplayBase.cc, 430
core/DisplayBase.hh, 430
core/DisplaySympy.cc, 431
core/DisplaySympy.hh, 431
core/DisplayTeX.cc, 432
core/DisplayTeX.hh, 433
core/DisplayTerminal.cc, 431
core/DisplayTerminal.hh, 432
core/Dummies.hh, 434
core/Exceptions.cc, 434
core/Exceptions.hh, 434
core/Exchange.cc, 435
core/Exchange.hh, 435
core/Functional.cc, 436
core/Functional.hh, 436
core/IndexIterator.cc, 437
core/IndexIterator.hh, 437
core/Kernel.cc, 438
core/Kernel.hh, 438
core/Parser.cc, 438
core/Parser.hh, 438
core/Permutations.hh, 439
core/PreClean.cc, 440
core/PreClean.hh, 440
core/PreProcessor.cc, 441
core/PreProcessor.hh, 441
core/ProgressMonitor.cc, 442
core/ProgressMonitor.hh, 442
core/Props.cc, 468
core/Props.hh, 468
core/PythonCdb.cc, 471
core/PythonCdb.hh, 476
core/PythonException.cc, 477
core/PythonException.hh, 477
core/ServerWrapper.cc, 478
core/ServerWrapper.hh, 478
core/Stopwatch.cc, 478
core/Stopwatch.hh, 478
core/Storage.cc, 479
core/Storage.hh, 480
core/Symbols.cc, 481
core/Symbols.hh, 481
core/SympyCdb.cc, 481
core/SympyCdb.hh, 481
core/TerminalStream.cc, 482
core/TerminalStream.hh, 482
core/YoungTab.cc, 483

core/YoungTab.hh, 484
core/algorithms/canonicalise.cc, 396
core/algorithms/canonicalise.hh, 396
core/algorithms/collect_components.cc, 396
core/algorithms/collect_components.hh, 397
core/algorithms/collect_factors.cc, 397
core/algorithms/collect_factors.hh, 397
core/algorithms/collect_terms.cc, 397
core/algorithms/collect_terms.hh, 398
core/algorithms/combine.cc, 398
core/algorithms/combine.hh, 398
core/algorithms/complete.cc, 398
core/algorithms/complete.hh, 399
core/algorithms/component.hh, 399
core/algorithms/decompose_product.cc, 399
core/algorithms/decompose_product.hh, 399
core/algorithms/distribute.cc, 400
core/algorithms/distribute.hh, 400
core/algorithms/drop_weight.cc, 400
core/algorithms/drop_weight.hh, 400
core/algorithms/einsteinify.cc, 401
core/algorithms/einsteinify.hh, 401
core/algorithms/eliminate_kronecker.cc, 401
core/algorithms/eliminate_kronecker.hh, 401
core/algorithms/eliminate_metric.cc, 402
core/algorithms/eliminate_metric.hh, 402
core/algorithms/epsilon_to_delta.cc, 402
core/algorithms/epsilon_to_delta.hh, 402
core/algorithms/evaluate.cc, 403
core/algorithms/evaluate.hh, 403
core/algorithms/expand.cc, 403
core/algorithms/expand.hh, 404
core/algorithms/expand_delta.cc, 404
core/algorithms/expand_delta.hh, 404
core/algorithms/expand_diracbar.cc, 404
core/algorithms/expand_diracbar.hh, 405
core/algorithms/expand_power.cc, 405
core/algorithms/expand_power.hh, 405
core/algorithms/factor_in.cc, 405
core/algorithms/factor_in.hh, 406
core/algorithms/factor_out.cc, 406
core/algorithms/factor_out.hh, 406
core/algorithms/fierz.cc, 406
core/algorithms/fierz.hh, 407
core/algorithms/flatten_product.cc, 407
core/algorithms/flatten_product.hh, 407
core/algorithms/flatten_sum.cc, 407
core/algorithms/flatten_sum.hh, 408
core/algorithms/indexsort.cc, 408
core/algorithms/indexsort.hh, 408
core/algorithms/integrate_by_parts.cc, 408
core/algorithms/integrate_by_parts.hh, 409
core/algorithms/join_gamma.cc, 409
core/algorithms/join_gamma.hh, 409
core/algorithms/keep_terms.cc, 409
core/algorithms/keep_terms.hh, 410
core/algorithms/lr_tensor.cc, 410
core/algorithms/lr_tensor.hh, 410
core/algorithms/map_sympy.cc, 410
core/algorithms/map_sympy.hh, 411
core/algorithms/order.cc, 411
core/algorithms/order.hh, 411
core/algorithms/product_rule.cc, 412
core/algorithms/product_rule.hh, 412
core/algorithms/reduce_delta.cc, 412
core/algorithms/reduce_delta.hh, 412
core/algorithms/rename_dummies.cc, 413
core/algorithms/rename_dummies.hh, 413
core/algorithms/replace_match.cc, 413
core/algorithms/replace_match.hh, 413
core/algorithms/rewrite_indices.cc, 414
core/algorithms/rewrite_indices.hh, 414
core/algorithms/sort_product.cc, 414
core/algorithms/sort_product.hh, 414
core/algorithms/sort_spinors.cc, 415
core/algorithms/sort_spinors.hh, 415
core/algorithms/sort_sum.cc, 415
core/algorithms/sort_sum.hh, 415
core/algorithms/split.cc, 416
core/algorithms/split.hh, 416
core/algorithms/split_gamma.cc, 416
core/algorithms/split_gamma.hh, 416
core/algorithms/split_index.cc, 416
core/algorithms/split_index.hh, 417
core/algorithms/substitute.cc, 417
core/algorithms/substitute.hh, 417
core/algorithms/sym.cc, 417
core/algorithms/sym.hh, 418
core/algorithms/tab_basics.cc, 418
core/algorithms/tab_basics.hh, 418
core/algorithms/take_match.cc, 418
core/algorithms/take_match.hh, 419
core/algorithms/unwrap.cc, 419
core/algorithms/unwrap.hh, 419
core/algorithms/vary.cc, 420
core/algorithms/vary.hh, 420
core/algorithms/young_project.cc, 420
core/algorithms/young_project.hh, 420
core/algorithms/young_project_product.cc, 421
core/algorithms/young_project_product.hh, 421
core/algorithms/young_project_tensor.cc, 421
core/algorithms/young_project_tensor.hh, 421
core/cadabra2, 422
core/cadabra2_defaults.py, 422
core/cadabra2python.cc, 423
core/echokernel.py, 434
core/passing.cc, 439
core/properties/Accent.cc, 443
core/properties/Accent.hh, 443
core/properties/AntiCommuting.cc, 443
core/properties/AntiCommuting.hh, 443
core/properties/AntiSymmetric.cc, 444
core/properties/AntiSymmetric.hh, 444
core/properties/Commuting.cc, 444
core/properties/Commuting.hh, 444
core/properties/CommutingAsProduct.cc, 445

core/properties/CommutingAsProduct.hh, 445
core/properties/CommutingAsSum.cc, 445
core/properties/CommutingAsSum.hh, 445
core/properties/CommutingBehaviour.cc, 446
core/properties/CommutingBehaviour.hh, 446
core/properties/Coordinate.cc, 446
core/properties/Coordinate.hh, 446
core/properties/DAntiSymmetric.cc, 446
core/properties/DAntiSymmetric.hh, 447
core/properties/Depends.cc, 447
core/properties/Depends.hh, 447
core/properties/DependsBase.hh, 447
core/properties/DependsInherit.cc, 448
core/properties/DependsInherit.hh, 448
core/properties/Derivative.cc, 448
core/properties/Derivative.hh, 448
core/properties/Diagonal.cc, 449
core/properties/Diagonal.hh, 449
core/properties/DifferentialForm.cc, 449
core/properties/DifferentialForm.hh, 449
core/properties/DifferentialFormBase.hh, 450
core/properties/DiracBar.cc, 450
core/properties/DiracBar.hh, 450
core/properties/Distributable.cc, 451
core/properties/Distributable.hh, 451
core/properties/EpsilonTensor.cc, 451
core/properties/EpsilonTensor.hh, 451
core/properties/ExteriorDerivative.cc, 451
core/properties/ExteriorDerivative.hh, 452
core/properties/FilledTableau.cc, 452
core/properties/FilledTableau.hh, 452
core/properties/GammaMatrix.cc, 452
core/properties/GammaMatrix.hh, 453
core/properties/GammaTraceless.cc, 453
core/properties/GammaTraceless.hh, 453
core/properties/ImaginaryI.cc, 453
core/properties/ImaginaryI.hh, 454
core/properties/ImplicitIndex.cc, 454
core/properties/ImplicitIndex.hh, 454
core/properties/IndexInherit.hh, 454
core/properties/Indices.cc, 455
core/properties/Indices.hh, 455
core/properties/Integer.cc, 455
core/properties/Integer.hh, 455
core/properties/Integral.hh, 456
core/properties/InverseMetric.cc, 456
core/properties/InverseMetric.hh, 456
core/properties/KroneckerDelta.cc, 456
core/properties/KroneckerDelta.hh, 457
core/properties/LaTeXForm.cc, 457
core/properties/LaTeXForm.hh, 457
core/properties/Matrix.cc, 457
core/properties/Matrix.hh, 457
core/properties/Metric.cc, 458
core/properties/Metric.hh, 458
core/properties/NonCommuting.cc, 458
core/properties/NonCommuting.hh, 458
core/properties/NumericalFlat.cc, 459
core/properties/NumericalFlat.hh, 459
core/properties/PartialDerivative.cc, 459
core/properties/PartialDerivative.hh, 459
core/properties/RiemannTensor.cc, 460
core/properties/RiemannTensor.hh, 460
core/properties/SatisfiesBianchi.cc, 460
core/properties/SatisfiesBianchi.hh, 460
core/properties/SelfAntiCommuting.cc, 461
core/properties/SelfAntiCommuting.hh, 461
core/properties/SelfCommuting.cc, 461
core/properties/SelfCommuting.hh, 461
core/properties/SelfCommutingBehaviour.hh, 462
core/properties/SelfNonCommuting.cc, 462
core/properties/SelfNonCommuting.hh, 462
core/properties/SortOrder.cc, 462
core/properties/SortOrder.hh, 462
core/properties/Spinor.cc, 463
core/properties/Spinor.hh, 463
core/properties/Symbol.cc, 463
core/properties/Symbol.hh, 463
core/properties/Symmetric.cc, 464
core/properties/Symmetric.hh, 464
core/properties/Tableau.cc, 464
core/properties/Tableau.hh, 464
core/properties/TableauBase.cc, 465
core/properties/TableauBase.hh, 465
core/properties/TableauSymmetry.cc, 465
core/properties/TableauSymmetry.hh, 465
core/properties/Traceless.cc, 466
core/properties/Traceless.hh, 466
core/properties/Weight.cc, 466
core/properties/Weight.hh, 466
core/properties/WeightBase.hh, 466
core/properties/WeightInherit.cc, 467
core/properties/WeightInherit.hh, 467
core/properties/WeylTensor.cc, 467
core/properties/WeylTensor.hh, 467
core/ptest.cc, 469
core/test_permutations.cc, 482
core/test_preprocessor.cc, 483
crash_window_hidden
    cadabra::NotebookWindow, 262
create
    Cadabra, 92
    cadabra::index_iterator, 214
create_empty_scope
    PythonCdb.cc, 474
    PythonCdb.hh, 477
create_millis
    snoop::Snoop::AppEntry, 88
    snoop::Snoop::LogEntry, 241
create_scope
    Python Core, 32
create_scope_from_global
    PythonCdb.cc, 474
    PythonCdb.hh, 477
create_tables
    snoop::SnoopImpl, 312

createExceptionClass
PythonCdb.cc, 474
created_by_client
cadabra::DataCell::id_t, 206
css_provider
cadabra::NotebookWindow, 262
cur
preprocessor, 277
cur_pos
preprocessor, 278
cur_str
preprocessor, 278
curly_bracket_operators
cadabra::DisplayTeX, 144
current_
combin::symmetriser, 338
current_bracket
cadabra::Parser, 269
current_bracket_
preprocessor, 276
current_canvas
cadabra::NotebookWindow, 262
cadabra_osx::NotebookWindow, 254
current_cell
cadabra::NotebookWindow, 262
current_hdl
Server, 306
current_id
Server, 306
current_mode
cadabra::Parser, 269
current_multiplicity
combin::symm_helper, 333
combin::symm_val_helper, 335
current_parent_rel
cadabra::Parser, 269
current_weight
combin::combinations_base, 107

DEBUG
Compare.cc, 429
map_sympy.cc, 411
DTree
cadabra, 49
DataCell
cadabra::DataCell, 122
datacell
cadabra::CodeInput::exp_input_tv, 182
cadabra::TeXView, 358
db
snoop::SnoopImpl, 314
decompose_product
cadabra::decompose_product, 124
def_algo_1
PythonCdb.cc, 474
def_prop
PythonCdb.cc, 474
default_is_product_
preprocessor, 276
degree
cadabra::DifferentialForm, 131
cadabra::DifferentialFormBase, 132
cadabra::ExteriorDerivative, 187
degree_
cadabra::DifferentialForm, 132
dense_factor
cadabra::evaluate, 164
dependencies
cadabra::Depends, 126
cadabra::DependsBase, 127
cadabra::DependsInherit, 128
cadabra::evaluate, 164
dependencies_
cadabra::Depends, 127
depsyms
cadabra::DisplaySympy, 138
derivative_acting_on_arg
cadabra::integrate_by_parts, 223
determine_intersection
cadabra::Algorithm, 81
determine_intersection_ranges
combin, 56
difference
cadabra::Integer, 221
difference_type
yngtab::filled_tableau::iterator_base, 229
diffsec
Stopwatch, 325
diffusec
Stopwatch, 325
dim
cadabra::TeXView, 357
cadabra::decompose_product, 125
cadabra::fierz, 192
dim_output_cells
cadabra::NotebookWindow, 258
dimension
cadabra::FilledTableau, 197
cadabra::Spinor, 320
cadabra::Tableau, 341
yngtab::tableau_base, 344
disable_stacks
cadabra::DocumentThread, 150
discard_command_node
cadabra::Algorithm, 84
dispatch
cadabra::DisplayBase, 134
cadabra::DisplaySympy, 136
cadabra::DisplayTerminal, 139
cadabra::DisplayTeX, 142
dispatch_base
PythonCdb.cc, 474
dispatch_ex
PythonCdb.cc, 474
dispatch_message
Server, 304
dispatch_string

PythonCdb.cc, 474
dispatcher
cadabra::NotebookWindow, 262
dispatcher_t
cadabra, 49
display
cadabra2_defaults, 54
cadabra::Integer, 221
Integral, 222
DisplayBase
cadabra::DisplayBase, 134
DisplaySympy
cadabra::DisplaySympy, 136
DisplaySympy.hh
kunichar, 431
DisplayTeX.cc
k_unichar_to_utf8, 433
nbsp, 433
unichar, 433
zwnbsp, 433
DisplayTeX.hh
kunichar, 434
DisplayTerminal
cadabra::DisplayTerminal, 139
DisplayTerminal.hh
kunichar, 432
DisplayTeX
cadabra::DisplayTeX, 142
displayhook
cadabra2_defaults, 54
distribute
cadabra::distribute, 146
do_apply
cadabra::drop_keep_weight, 152
do_execute
echokernel::EchoKernel, 154
do_filledtableau
cadabra::lr_tensor, 242
do_list
Core, 26
do_subtree
Core, 26
do_tableau
cadabra::lr_tensor, 243
doc
cadabra::DocumentThread, 150
doc/main.md, 485
doc/modules.dox, 485
docthread
cadabra::ComputeThread, 117
document
cadabra::DataCell, 121
cadabra::VisualCell, 370
DocumentThread
cadabra::DocumentThread, 149
doit
cadabra::sym, 331
drop_keep_weight
cadabra::drop_keep_weight, 152
drop_weight
cadabra::drop_weight, 153
dset1
cadabra::rename_dummies, 293
dset2
cadabra::rename_dummies, 293
dumpmap
cadabra::Algorithm, 81

eat_initial_whitespace_
preprocessor, 278
ebox
cadabra::NotebookCanvas, 252
echokernel, 57
kernel_class, 57
echokernel.EchoKernel, 153
echokernel::EchoKernel
banner, 154
do_execute, 154
implementation, 154
implementation_version, 154
language, 154
language_info, 154
language_version, 154
edit
cadabra::CodeInput, 98
einsteinify
cadabra::einsteinify, 155
eliminate_converter
cadabra::eliminate_converter, 156
eliminate_kronecker
cadabra::eliminate_kronecker, 157
eliminate_metric
cadabra::eliminate_metric, 158
email
snoop, 57
end
cadabra::index_iterator, 214
cadabra::keyval_t, 235
yngtab::filled_tableau, 195
end_
combin::combinations_base, 107
end_column
yngtab::filled_tableau, 195
end_index
cadabra::Algorithm, 81
end_wrap
cadabra::TeXEngine::TeXRequest, 356
endif
core/CMakeLists.txt, 426
engine
cadabra::NotebookWindow, 262
cadabra::TeXView, 358
entry_accepted
combin::combinations_base, 106
epsilon_to_delta
cadabra::epsilon_to_delta, 159
epsilons

cadabra::epsilon_to_delta, 160
equal_subtree
cadabra::Ex_comparator, 174
equal_without_numbers
cadabra::Algorithm, 81
equals
cadabra::CommutingBehaviour, 112
cadabra::Indices, 218
cadabra::SortOrder, 318
cadabra::property, 287
equation_by_name
cadabra::Ex, 168
equation_by_number
cadabra::Ex, 168
equation_by_number_or_name
cadabra::Ex, 168
equation_label
cadabra::Ex, 168
equation_number
cadabra::Ex, 168
equation_number_or_name
cadabra::Ex, 168
erase
cadabra::Parser, 269
cadabra::keyval_t, 235
preprocessor, 277
preprocessor::accu_t, 62
erase_expression
cadabra::Ex, 168
erase_file
cadabra::TeXEngine, 353
err
exec_stream_t, 180
error
cadabra::DataCell, 121
Server::Block, 91
snoop, 57
error_code_t
exec_stream_t, 179
error_t
exec_stream_t::error_t, 162
escape_quotes
cadabra, 51
evaluate
cadabra::evaluate, 163
Ex
cadabra::Ex, 167
Ex_comparator
cadabra::Ex_comparator, 173
ex_from_string
cadabra::Kernel, 233
Ex_getitem
PythonCdb.cc, 474
Ex_getslice
PythonCdb.cc, 474
Ex_is_equivalent
cadabra::Ex_is_equivalent, 176
Ex_is_less
cadabra::Ex_is_less, 177
Ex_latex_
Python Core, 32
Ex_len
PythonCdb.cc, 474
Ex_repr_
PythonCdb.cc, 474
PythonCdb.hh, 477
Ex_setitem
PythonCdb.cc, 474
Ex_str_
Python Core, 32
Ex_to_Sympy
Scalar algebra, 34
exact_match
cadabra::property, 287
exceptions
exec_stream_t, 180
Exchange.cc
operator<, 435
exec-stream.cc
HELPERS_CPP, 486
HELPERS_H, 486
IMPL_CPP, 486
exec_stream_t, 178
∼exec_stream_t, 180
close, 180
close_in, 180
err, 180
error_code_t, 179
exceptions, 180
exec_stream_t, 180
exit_code, 180
impl_t, 181
in, 180
kill, 180
m_impl, 181
new_impl, 180
operator=, 180
out, 180
s_all, 179
s_child, 179
s_err, 179
s_in, 179
s_out, 179
set_binary_mode, 180
set_buffer_limit, 180
set_text_mode, 180
set_wait_timeout, 180
start, 180, 181
stream_kind_t, 179
timeout_t, 179
exec_stream_t::error_t, 161
∼error_t, 162
compose, 162
error_t, 162
m_msg, 162
what, 162

exec_stream_t::next_arg_impl_t
m_arg, 248
m_args_end, 248
m_args_i, 248
next, 248
next_arg_impl_t, 248
exec_stream_t::next_arg_impl_t< iterator >, 248
exec_stream_t::next_arg_t, 249
~next_arg_t, 249
next, 249
execute
cadabra::ActionAddCell, 64
cadabra::ActionBase, 65
cadabra::ActionEraseText, 67
cadabra::ActionInsertText, 68
cadabra::ActionPositionCursor, 70
cadabra::ActionRemoveCell, 71
cadabra::ActionSetRunStatus, 72
cadabra::ActionSplitCell, 74
execute_cell
cadabra::ComputeThread, 115
exit_code
exec_stream_t, 180
exp_input_tv
cadabra::CodeInput::exp_input_tv, 182
expand
cadabra::expand, 183
cadabra::join_gamma, 231
expand_delta
cadabra::expand_delta, 184
expand_diracbar
cadabra::expand_diracbar, 185
expand_power
cadabra::expand_power, 186
export_as_HTML
cadabra, 51
export_as_LaTeX
cadabra, 51
export_as_python
cadabra, 51
Expression comparison, 29
subtree_compare, 29
extra_brackets_for_symbols
cadabra::DisplayTeX, 144
extra_sign
cadabra::exchange::identical_tensors_t, 207

f1
cadabra::decompose_product, 125
f2
cadabra::decompose_product, 125
fact
combin, 56
factnodes
cadabra::factor_in, 189
factor_hash
cadabra::collect_factors, 101
factor_hash_iterator_t
cadabra::collect_factors, 100
factor_hash_t
cadabra::collect_factors, 100
factor_in
cadabra::factor_in, 188
factor_locations
cadabra::Ex_comparator, 175
factor_moving_signs
cadabra::Ex_comparator, 175
factor_out
cadabra::factor_out, 190
factorial
combin, 56
factors
cadabra::factor_in, 189
fatal
snoop, 57
fetch_from_python
PythonCdb.cc, 474
PythonCdb.hh, 477
fierz
cadabra::fierz, 191
fill_asym_ranges
cadabra::decompose_product, 125
fill_hash_map
cadabra::collect_factors, 101
cadabra::collect_terms, 102
cadabra::factor_in, 189
fill_index_position_map
cadabra::Algorithm, 81
fill_map
cadabra::Algorithm, 81
filled_tableau< T >
yngtab::filled_tableau::in_column_iterator, 213
yngtab::filled_tableau::iterator, 228
final_weight_constraints_check
combin::combinations_base, 107
finalise
cadabra::Parser, 269
find
cadabra::keyval_t, 235
Images, 207
Perm, 272
yngtab::filled_tableau, 195
find_arg_superset
cadabra::Algorithm, 81
find_argument_lists
cadabra::Algorithm, 81
find_cell_by_id
cadabra::ComputeThread, 116
find_in_list
Core, 27
find_in_subtree
Core, 27
find_modulo_parent_rel
cadabra::Algorithm, 81
find_obj
cadabra::tab_basics, 340
first_one

combin::symm_helper, 333
combin::symm_val_helper, 335
fixed
cadabra::Indices, 218
fl
cadabra::str_node, 328
flatten_and_erase
cadabra::Ex, 168
flatten_product
cadabra::flatten_product, 199
flatten_sum
cadabra::flatten_sum, 200
flip_parent_rel
cadabra::str_node, 327
flip_sign
cadabra, 51
flush
snoop, 57
follow_cell
cadabra::NotebookWindow, 262
font_size_
cadabra::TeXEngine, 354
font_step
cadabra::NotebookWindow::Prefs, 274
for_obj
Property, 285
force_prod_wrap
cadabra::Algorithm, 81
free
cadabra::Indices, 218
from
cadabra::Integer, 221
from_json
snoop::Snoop::AppEntry, 88
snoop::Snoop::LogEntry, 241
from_pos
cadabra::ActionEraseText, 67
from_python
iterable_converter, 226
Front-end, 38
frontend/common/CMakeLists.txt, 426
cmake_minimum_required, 426
frontend/common/TeXEngine.cc, 486
frontend/common/TeXEngine.hh, 487
frontend/common/exec-stream.cc, 485
frontend/common/exec-stream.h, 486
frontend/common/test_tex.cc, 486
frontend/gtkmm/CMakeLists.txt, 426
frontend/gtkmm/Cadabra.cc, 487
frontend/gtkmm/Cadabra.hh, 488
frontend/gtkmm/CodeInput.cc, 488
frontend/gtkmm/CodeInput.hh, 488
frontend/gtkmm/ImageView.cc, 489
frontend/gtkmm/ImageView.hh, 489
frontend/gtkmm/NotebookCanvas.cc, 490
frontend/gtkmm/NotebookCanvas.hh, 490
frontend/gtkmm/NotebookWindow.cc, 491
frontend/gtkmm/NotebookWindow.hh, 491
frontend/gtkmm/Preferences.hh, 492
frontend/gtkmm/TeXView.cc, 492
frontend/gtkmm/TeXView.hh, 492
frontend/gtkmm/VisualCell.hh, 493
frontend/gtkmm/main.cc, 489
frontend/osx/Cadabra/Cadabra/AppDelegate.h, 493
frontend/osx/Cadabra/Cadabra/AppDelegate.mm, 493
frontend/osx/Cadabra/Cadabra/NotebookCanvas.hh, 490
frontend/osx/Cadabra/Cadabra/NotebookCanvas.mm, 494
frontend/osx/Cadabra/Cadabra/NotebookWindow.hh, 492
frontend/osx/Cadabra/Cadabra/NotebookWindow.mm, 494
frontend/osx/Cadabra/Cadabra/main.m, 494
frontend/osx/Cadabra/NotebookController.h, 494
frontend/osx/Cadabra/NotebookController.mm, 494
full_class
cadabra::split_index, 323
fun
passing.cc, 439
ptest.cc, 469

gam1
cadabra::fierz, 192
gam2
cadabra::fierz, 192
gamma_bracket_
cadabra::join_gamma, 231
gamma_name_
cadabra::join_gamma, 231
gammamat
cadabra::sort_spinors, 317
gammatraceless
cadabra::exchange::identical_tensors_t, 207
Garnir_set
yngtab::filled_tableau, 195
generating_set
cadabra::canonicalise, 95
get
cadabra::Properties, 282
cadabra::Symbol, 332
get_app_registrations
snoop::SnoopImpl, 312
get_back_insert_iterator
yngtab::tableaux, 349
get_composite
cadabra::Properties, 282, 283
get_cursor_position
cadabra::GUIBase, 203
cadabra::NotebookWindow, 258
get_dummy
cadabra::Algorithm, 81, 82
get_dummy_sw
cadabra::Algorithm, 84
get_index_set_name
cadabra::canonicalise, 95
get_indexgroup

cadabra::TableauBase, 346
get_kernel_from_scope
Python Core, 33
get_node_gs
cadabra::exchange, 178
get_tab
cadabra::AntiSymmetric, 86
cadabra::DAntiSymmetric, 120
cadabra::Derivative, 129
cadabra::KroneckerDelta, 236
cadabra::PartialDerivative, 270
cadabra::SatisfiesBianchi, 297
cadabra::Symmetric, 335
cadabra::TableauBase, 346
cadabra::TableauSymmetry, 347
get_token
cadabra::Parser, 269
get_token_
preprocessor, 277
get_user_uuid
snoop::Snoop, 309
snoop::SnoopImpl, 312
gm1
cadabra::join_gamma, 231
gm2
cadabra::join_gamma, 231
gmn
cadabra::drop_keep_weight, 152
goal
cadabra::complete, 113
greek
cadabra::symbols, 55
group
ProgressMonitor, 280
ServerWrapper, 307
Gtk+, 39
gui
cadabra::ComputeThread, 117
cadabra::DocumentThread, 150
gui_thread_id
cadabra::ComputeThread, 117

HELPERS_CPP
exec-stream.cc, 486
HELPERS_H
exec-stream.cc, 486
HTML_recurse
cadabra, 51
half
cadabra, 51
halt
cadabra::index_iterator, 215
handle_button_press
cadabra::CodeInput, 98
handle_components
cadabra::evaluate, 164
handle_derivative
cadabra::evaluate, 164
handle_epsilon
cadabra::evaluate, 164
handle_erase
cadabra::CodeInput, 98
handle_factor
cadabra::evaluate, 164
handle_insert
cadabra::CodeInput, 98
handle_latex_errors
cadabra::TeXEngine, 353
handle_one_index
cadabra::eliminate_converter, 156
handle_prod
cadabra::evaluate, 164
handle_sum
cadabra::evaluate, 164
handle_term
cadabra::integrate_by_parts, 223
handles
cadabra2_defaults::Server, 301
Server, 304
has
cadabra::Properties, 283
has_nullifying_trace
yngtab::filled_tableau, 195
hash
combin, 56
hashval_t
cadabra, 49
have_matplotlib
cadabra2_defaults, 54
hbox
cadabra::TeXView, 358
hdl
Server::Block, 91
Server::Connection, 118
head
cadabra::Properties, 283
head_
cadabra::map_sympy, 244
head_is_generated
preprocessor::accu_t, 62
height
cadabra::TeXEngine::TeXRequest, 356
height_
cadabra::TeXEngine::TeXRequest, 356
help_t
cadabra::DocumentThread, 148
help_type_and_topic
cadabra::DocumentThread, 149
hidden
cadabra::DataCell, 122
cadabra::property, 287, 288
hidden_
cadabra::property, 289
history
cadabra::Ex, 170
history_size
cadabra::Ex, 168

hook_length
yngtab::tableau_base, 344
hook_length_prod
yngtab::tableau_base, 344
horizontal_pixels_
cadabra::TeXEngine, 354

IMPL_CPP
exec-stream.cc, 486
id
cadabra::DataCell, 122
cadabra::DataCell::id_t, 206
snoop::Snoop::AppEntry, 88
snoop::Snoop::LogEntry, 241
id_for_uuid_statement
snoop::SnoopImpl, 314
id_match
cadabra::property, 287
id_t
cadabra::DataCell::id_t, 206
idle_handler
cadabra::NotebookWindow, 258
ignore_numbers_
cadabra::sort_product, 316
ii_first
cadabra::expand, 184
ii_last
cadabra::expand, 184
iloc
cadabra::combine, 109
image
cadabra::ImageView, 208
cadabra::TeXEngine::TeXRequest, 356
cadabra::TeXView, 358
image_
cadabra::TeXEngine::TeXRequest, 356
image_png
cadabra::DataCell, 121
ImageView
cadabra::ImageView, 208
imagebox
cadabra::VisualCell, 371
Images, 207
find, 207
images, 208
images
Images, 208
impl
snoop::Snoop, 310
impl_t
exec_stream_t, 181
implementation
echokernel::EchoKernel, 154
implementation_version
echokernel::EchoKernel, 154
import
cadabra::DisplaySympy, 136
in
cadabra::ActionPositionCursor, 69
exec_stream_t, 180
in_column_iterator
yngtab::filled_tableau::in_column_iterator, 212
inbox
cadabra::VisualCell, 371
ind1
cadabra::decompose_product, 125
ind2
cadabra::decompose_product, 125
ind_dummy
cadabra::eliminate_converter, 157
ind_free
cadabra::eliminate_converter, 157
independent
cadabra::Indices, 218
index_factors
cadabra::map_sympy, 244
index_in_set
cadabra::Algorithm, 82
index_iterator
cadabra::index_iterator, 214
index_map_t
cadabra::Algorithm, 78
index_parity
cadabra::Algorithm, 82
index_position_map_t
cadabra::Algorithm, 78
index_sw
cadabra::Algorithm, 84
index_value_map
cadabra::Ex_comparator, 175
indexlocmap_t
cadabra::combine, 109
indexsort
cadabra::indexsort, 217
Indices
cadabra::Indices, 218
indices_equivalent
cadabra::decompose_product, 125
indprop
cadabra::fierz, 192
info
snoop, 57
init
cadabra::CodeInput, 98
cadabra::ComputeThread, 116
Server, 304
snoop::Snoop, 309
snoop::SnoopImpl, 312
init_ipython
PythonCdb.cc, 474
inject_defaults
Python Core, 33
inject_property
cadabra::Kernel, 233
input
Server::Block, 91
input_asym

combin::combinations_base, 107
combin::symmetriser, 338
insert_list_prop
cadabra::Properties, 283
insert_pos
cadabra::ActionInsertText, 68
insert_prop
cadabra::Properties, 283
insert_statement
snoop::SnoopImpl, 314
install
core/CMakeLists.txt, 426
int_and_derivative_related
cadabra::integrate_by_parts, 223
Integral, 222
~Integral, 222
display, 222
name, 222
parse, 222
integrate_by_parts
cadabra::integrate_by_parts, 223
internal_property_map_t
cadabra::Properties::registered_property_map_t, 291
InternalError, 224
InternalError, 224
py_what, 224
InternalErrorType
PythonCdb.cc, 475
interrupted
cadabra::Algorithm, 84
InterruptionException, 224
InterruptionException, 225
intersection_number
cadabra::Algorithm, 82
invalidate_all
cadabra::TeXEngine, 353
InverseMetric
cadabra::InverseMetric, 225
invert_matrix
Scalar algebra, 34
ip_address
snoop::Snoop::AppEntry, 88
is_allowed_by_weight_constraints
combin::combinations_base, 107
is_already_bracketed_
preprocessor, 277
is_autodeclare_wildcard
cadabra::str_node, 327
is_bracket_
preprocessor, 277
is_closing_bracket
cadabra::Parser, 269
is_closing_bracket_
preprocessor, 277
is_command
cadabra::str_node, 328
is_component
cadabra::evaluate, 164
is_configured
cadabra::NotebookWindow, 262
is_conversion_object
cadabra::eliminate_converter, 156
cadabra::eliminate_metric, 158
is_derivative_argument
cadabra::rewrite_indices, 295
is_diff
cadabra::flatten_product, 199
is_digits_
preprocessor, 277
is_factorlike
cadabra::Algorithm, 82
is_identity
cadabra::str_node, 328
is_in
Core, 27
is_index
cadabra::index_iterator, 214
cadabra::str_node, 328
preprocessor::accu_t, 62
is_indexplus_wildcard
cadabra::str_node, 328
is_indexstar_wildcard
cadabra::str_node, 328
is_inert_command
cadabra::str_node, 328
is_infix_operator_
preprocessor, 277
is_integer
cadabra::str_node, 328
is_link
cadabra::Parser, 269
is_link_
preprocessor, 277
is_name_wildcard
cadabra::str_node, 328
is_nonprod_factor_in_prod
cadabra::Algorithm, 82
is_number
cadabra::Parser, 269
is_numbered_symbol
cadabra::str_node, 328
is_object_wildcard
cadabra::str_node, 328
is_opening_bracket
cadabra::Parser, 269
is_opening_bracket_
preprocessor, 277
is_quoted_string
cadabra::str_node, 328
is_range_wildcard
cadabra::str_node, 328
is_rational
cadabra::Ex, 169
cadabra::str_node, 328
is_registered

cadabra::DocumentThread, 149
is_siblings_wildcard
cadabra::str_node, 328
is_simple_symmetry
cadabra::TableauBase, 346
is_single_term
cadabra::Algorithm, 82
is_termlike
cadabra::Algorithm, 82
is_unsimplified_integer
cadabra::str_node, 328
is_unsimplified_rational
cadabra::str_node, 328
is_zero
cadabra::str_node, 328
it
cadabra::indexsort::less_indexed_treenode, 239
iterable_converter, 226
construct, 226
convertible, 226
from_python, 226
iterator
cadabra::Algorithm, 78
cadabra::Properties::registered_property_map_t, 291
cadabra::keyval_t, 235
yngtab::filled_tableau::iterator, 228
iterator_category
yngtab::filled_tableau::iterator_base, 229

JSON_deserialise
cadabra, 51
JSON_in_recurse
cadabra, 51
JSON_recurse
cadabra, 52
JSON_serialise
cadabra, 52
job
Server, 306
join_gamma
cadabra::join_gamma, 230

k
passing.cc, 439
k_unichar_to_utf8
DisplayTeX.cc, 433
keep_after_eval
cadabra::str_node::flag_t, 198
keep_terms
cadabra::keep_terms, 232
keep_weight
cadabra::keep_weight, 233
keeptrack_tab_t
yngtab, 59
Kernel
cadabra::Kernel, 233
kernel
cadabra::Algorithm, 84
cadabra::DisplayBase, 134
cadabra::TerminalStream, 351
cadabra::indexsort::less_indexed_treenode, 239
kernel_class
echokernel, 57
kernel_label
cadabra::NotebookWindow, 262
kernel_spinner
cadabra::NotebookWindow, 262
kernel_spinner_status
cadabra::NotebookWindow, 262
kernel_string
cadabra::NotebookWindow, 262
keyvals
cadabra::keyval_t, 235
kill
exec_stream_t, 180
krdelta
cadabra::EpsilonTensor, 161
kunichar
DisplaySympy.hh, 431
DisplayTeX.hh, 434
DisplayTerminal.hh, 432
kvlist_t
cadabra::keyval_t, 235
kvpair_t
cadabra::keyval_t, 235

l_applied
cadabra::Ex, 167
l_checkpointed
cadabra::Ex, 167
l_error
cadabra::Ex, 167
l_no_action
cadabra::Ex, 167
LOC
Snoop.hh, 393
LR_add_box
yngtab, 59
LR_tensor
yngtab, 59
LaTeX_recurse
cadabra, 52
label
cadabra::drop_keep_weight, 152
cadabra::labelled_property, 237
language
echokernel::EchoKernel, 154
language_info
echokernel::EchoKernel, 154
language_version
echokernel::EchoKernel, 154
last_configure_width
cadabra::NotebookWindow, 262
latex
cadabra::DataCell, 121
cadabra::DocumentThread, 148
cadabra::GammaMatrix, 201

cadabra::ImplicitIndex, 210
cadabra::Indices, 218
cadabra::Metric, 246
cadabra::TableauSymmetry, 347
cadabra::WeylTensor, 376
cadabra::property, 288
latex_
cadabra::LaTeXForm, 238
Property, 285
latex_form
cadabra::LaTeXForm, 238
latex_linefeeds
cadabra::DisplayTeX, 144
latex_packages
cadabra::TeXEngine, 354
latex_spacing
cadabra::DisplayTeX, 144
latex_string
cadabra::TeXEngine::TeXRequest, 356
latex_to_html
cadabra, 52
latex_view
cadabra::DataCell, 121
left
cadabra::map_sympy, 244
legal_box
yngtab, 59
less_indexed_treenode
cadabra::indexsort::less_indexed_treenode, 239
less_without_numbers
cadabra::Algorithm, 82
lhs
PythonCdb.cc, 475
lhs_contains_dummies
cadabra::Ex_comparator, 175
cadabra::substitute, 330
line_per_node
cadabra::str_node::flag_t, 198
list_properties
PythonCdb.cc, 475
list_size
cadabra, 52
list_unwrap_single_element
cadabra::Ex, 169
list_wrap_single_element
cadabra::Ex, 169
load_file
cadabra::NotebookWindow, 258
load_from_string
cadabra::DocumentThread, 149
loc_file
snoop::Snoop::LogEntry, 241
loc_line
snoop::Snoop::LogEntry, 241
loc_method
snoop::Snoop::LogEntry, 241
locate_object_set
cadabra::Algorithm, 82
locate_single_object
cadabra::Algorithm, 82
log
snoop, 57
log_id
snoop::Snoop::LogEntry, 241
LogEntry
snoop::Snoop::LogEntry, 241
lr_tensor
cadabra::lr_tensor, 242

m_arg
exec_stream_t::next_arg_impl_t, 248
m_args_end
exec_stream_t::next_arg_impl_t, 248
m_args_i
exec_stream_t::next_arg_impl_t, 248
m_backslashname
cadabra::Parser, 268
m_childgroup
cadabra::Parser, 268
m_findchildren
cadabra::Parser, 268
m_impl
exec_stream_t, 181
m_initialgroup
cadabra::Parser, 268
m_msg
exec_stream_t::error_t, 162
m_name
cadabra::Parser, 268
m_property
cadabra::Parser, 268
m_singlecharname
cadabra::Parser, 268
m_skipwhite
cadabra::Parser, 268
m_verbatim
cadabra::Parser, 268
machine_id
snoop::Snoop::AppEntry, 88
main
cadabra-server.cc, 384
cadabra2cadabra.cc, 384
cadabra2html.cc, 385
cadabra2python.cc, 423
main.cc, 490
main.m, 494
passing.cc, 439
regexp_tester.cc, 390
test_client.cc, 394
test_permutations.cc, 483
test_preprocessor.cc, 483
test_talk_to_server.cc, 395
test_tex.cc, 486
main.cc
main, 490
main.m
main, 494

main_module
Server, 306
main_namespace
Server, 306
mainbox
cadabra::NotebookWindow, 262
PreferencesWindow, 274
majorana
cadabra::Spinor, 320
make_Ex_from_int
PythonCdb.cc, 475
make_Ex_from_string
PythonCdb.cc, 475
make_consistent_only
cadabra::flatten_product, 199
cadabra::flatten_sum, 200
make_list
Core, 27
make_x_and_inject_y
ptest.cc, 469
map_sympy
cadabra::map_sympy, 244
Scalar algebra, 34
map_sympy.cc
DEBUG, 411
map_sympy_wrapper
PythonCdb.cc, 475
master_insert
cadabra::Properties, 283
match
cadabra::pattern, 271
match_index_greater
cadabra::Ex_comparator, 173
match_index_less
cadabra::Ex_comparator, 173
match_subproduct
cadabra::Ex_comparator, 174
match_subsum
cadabra::Ex_comparator, 174
match_t
cadabra::Ex_comparator, 173
cadabra::property, 287
max_numbered_name
cadabra::Algorithm, 82
max_numbered_name_one
cadabra::Algorithm, 82
max_serial_number
cadabra::DataCell, 122
max_weights
combin::combinations_base, 107
mclose
cadabra2_defaults, 54
merge_component_children
cadabra::evaluate, 164
merge_components
cadabra::evaluate, 164
message
snoop::Snoop::LogEntry, 241
message_ptr
ComputeThread.hh, 386
SnoopPrivate.hh, 393
test_talk_to_server.cc, 395
Metric
cadabra::Metric, 246
metric
cadabra::EpsilonTensor, 161
cadabra::GammaMatrix, 202
cadabra::einsteinify, 155
millimeter_per_inch
cadabra::TeXEngine, 354
mod_prel
cadabra::Ex_is_less, 177
mode_t
cadabra::Parser, 268
modified
cadabra::NotebookWindow, 262
modify
cadabra::TeXEngine, 353
modulo_monoterm
cadabra::young_project_tensor, 380
mopen
cadabra2_defaults, 54
move_index
cadabra::Ex, 169
multiple_pick
combin::combinations_base, 107
multiplicative
cadabra::WeightInherit, 374
multiplicity
combin::symmetriser, 338
yngtab::tableau_base, 345
multiplier
cadabra::str_node, 328
combin::combinations, 104
combin::combinations_base, 107
multiplier_t
cadabra, 49
multiply
Core, 27
mx_first
cadabra::expand, 184
mx_last
cadabra::expand, 184
MyClient, 247
after_tree_change, 247
before_tree_change, 247
MyClient, 247
on_connect, 247
on_disconnect, 247
on_network_error, 247
on_progress, 247

name
cadabra::Accent, 61
cadabra::AntiCommuting, 85
cadabra::AntiSymmetric, 86
cadabra::Commuting, 110

cadabra::CommutingAsProduct, 111
cadabra::CommutingAsSum, 111
cadabra::Coordinate, 119
cadabra::DAntiSymmetric, 120
cadabra::Depends, 126
cadabra::DependsInherit, 128
cadabra::Derivative, 129
cadabra::Diagonal, 131
cadabra::DifferentialForm, 131
cadabra::DiracBar, 133
cadabra::Distributable, 145
cadabra::EpsilonTensor, 161
cadabra::ExteriorDerivative, 187
cadabra::FilledTableau, 197
cadabra::GammaMatrix, 201
cadabra::GammaTraceless, 202
cadabra::ImaginaryI, 209
cadabra::ImplicitIndex, 210
cadabra::IndexInherit, 216
cadabra::Indices, 219
cadabra::Inherit, 220
cadabra::Integer, 221
cadabra::InverseMetric, 225
cadabra::KroneckerDelta, 236
cadabra::LaTeXForm, 238
cadabra::Matrix, 245
cadabra::Metric, 246
cadabra::NonCommuting, 250
cadabra::NotebookWindow, 262
cadabra::NumericalFlat, 265
cadabra::PartialDerivative, 270
cadabra::PropertyInherit, 289
cadabra::RiemannTensor, 296
cadabra::SatisfiesBianchi, 297
cadabra::SelfAntiCommuting, 298
cadabra::SelfCommuting, 299
cadabra::SelfNonCommuting, 300
cadabra::SortOrder, 318
cadabra::Spinor, 319
cadabra::Symbol, 332
cadabra::Symmetric, 335
cadabra::Tableau, 341
cadabra::TableauSymmetry, 347
cadabra::Traceless, 360
cadabra::Weight, 371
cadabra::WeightInherit, 375
cadabra::WeylTensor, 376
cadabra::exchange::tensor_type_t, 351
cadabra::property, 288
cadabra::str_node, 328
Integral, 222
ProgressMonitor::Block, 90
ProgressMonitor::Total, 359

name_only
cadabra::str_node, 328

name_set
cadabra, 53

name_tab_t
cadabra::young_project, 378

named_parent
cadabra::Ex, 169

nametab
cadabra::young_project, 378

nbsp
DisplayTeX.cc, 433

needed_new_cell
cadabra::ActionPositionCursor, 70

needs_brackets
cadabra::DisplayBase, 134
cadabra::DisplaySympy, 136
cadabra::DisplayTerminal, 139
cadabra::DisplayTeX, 142

needs_generating
cadabra::TeXEngine::TeXRequest, 356

negative
cadabra::Spinor, 319

never
cadabra::Ex_comparator, 173

new_document
cadabra::DocumentThread, 149

new_impl
exec_stream_t, 180

new_running_
cadabra::ActionSetRunStatus, 73

newcell
cadabra::ActionAddCell, 64

newref
cadabra::ActionAddCell, 64
cadabra::ActionPositionCursor, 70
cadabra::ActionSplitCell, 74

next
cadabra::ActionPositionCursor, 69
exec_stream_t::next_arg_impl_t, 248
exec_stream_t::next_arg_t, 249

next_arg_impl_t
exec_stream_t::next_arg_impl_t, 248

next_is_product_
preprocessor, 278

nextstep
combin::combinations_base, 107

no_match
cadabra::property, 287

no_match_greater
cadabra::Ex_comparator, 173

no_match_indexpos_greater
cadabra::Ex_comparator, 173

no_match_indexpos_less
cadabra::Ex_comparator, 173

no_match_less
cadabra::Ex_comparator, 173

node_integer
cadabra::Algorithm, 82

node_match
cadabra::Ex_comparator, 173

node_one
cadabra::Algorithm, 83

node_zero
cadabra::Algorithm, 83
NonScalarException, 250
NonScalarException, 251
py_what, 251
NonScalarExceptionType
PythonCdb.cc, 475
none
cadabra::DocumentThread, 148
nonstandard_loc
yngtab::filled_tableau, 195
not_at_top
cadabra::Ex_comparator, 173
NotYetImplemented, 263
NotYetImplemented, 264
py_what, 264
NotYetImplementedType
PythonCdb.cc, 475
NotebookCanvas
cadabra::NotebookCanvas, 252
NotebookController, 252
scrollbox, 253
status_label, 253
notebookController
AppDelegate(), 87
NotebookController(), 253
NotebookController.mm
compute, 495
compute_thread, 495
NotebookWindow
cadabra::NotebookWindow, 257
cadabra_osx::NotebookWindow, 254
nset_t
cadabra, 49
nt1
cadabra::decompose_product, 125
nt2
cadabra::decompose_product, 125
nth_index_node
cadabra::young_project, 378
num1
cadabra::split_index, 323
num2
cadabra::split_index, 323
num_to_it
cadabra::tab_basics, 340
number_of_calls
cadabra::Algorithm, 84
number_of_cells_executing
cadabra::ComputeThread, 116
number_of_direct_indices
cadabra::Algorithm, 83
number_of_equations
cadabra::Ex, 169
number_of_indices
cadabra::Algorithm, 83
cadabra::exchange::identical_tensors_t, 207
cadabra::exchange::tensor_type_t, 351
cadabra::product_rule, 279
number_of_modifications
cadabra::Algorithm, 84
number_of_rows
yngtab::filled_tableau, 195
yngtab::tableau, 342
yngtab::tableau_base, 344
number_of_steps
cadabra::Ex, 169
numtab_t
cadabra::decompose_product, 124
numtabs_t
cadabra::decompose_product, 124

OS-X, 40
obj
cadabra::pattern, 271
objects
cadabra::order, 266
cadabra::sym, 331
obtain_uuid
snoop::SnoopImpl, 312
offset
cadabra::Ex_comparator, 175
on_activate
Cadabra, 92
on_back
cadabra::split_gamma, 322
on_block_error
Server, 304
on_block_finished
Server, 304
on_button_release_event
cadabra::TeXView, 357
on_client_close
snoop::SnoopImpl, 313
on_client_fail
snoop::SnoopImpl, 313
on_client_message
snoop::SnoopImpl, 313
on_client_open
snoop::SnoopImpl, 313
on_close
cadabra::ComputeThread, 116
Server, 304
on_configure_event
cadabra::NotebookWindow, 258
on_connect
cadabra::GUIBase, 204
cadabra::NotebookWindow, 258
cadabra_osx::NotebookWindow, 254
MyClient, 247
on_crash_window_closed
cadabra::NotebookWindow, 258
on_delete_event
cadabra::NotebookWindow, 259
on_disconnect
cadabra::GUIBase, 204
cadabra::NotebookWindow, 259

cadabra_osx::NotebookWindow, 254
MyClient, 247
on_draw
cadabra::CodeInput::exp_input_tv, 182
on_edit_cell_is_latex
cadabra::NotebookWindow, 259
on_edit_cell_is_python
cadabra::NotebookWindow, 259
on_edit_delete
cadabra::NotebookWindow, 259
on_edit_insert_above
cadabra::NotebookWindow, 259
on_edit_insert_below
cadabra::NotebookWindow, 259
on_edit_split
cadabra::NotebookWindow, 259
on_edit_undo
cadabra::NotebookWindow, 259
on_fail
cadabra::ComputeThread, 116
on_file_close
cadabra::NotebookWindow, 259
on_file_export_html
cadabra::NotebookWindow, 259
on_file_export_html_segment
cadabra::NotebookWindow, 259
on_file_export_latex
cadabra::NotebookWindow, 259
on_file_export_python
cadabra::NotebookWindow, 259
on_file_new
cadabra::NotebookWindow, 259
on_file_open
cadabra::NotebookWindow, 259
on_file_quit
cadabra::NotebookWindow, 259
on_file_save
cadabra::NotebookWindow, 259
on_file_save_as
cadabra::NotebookWindow, 259
on_focus_in_event
cadabra::CodeInput::exp_input_tv, 182
on_help
cadabra::NotebookWindow, 259
on_help_about
cadabra::NotebookWindow, 259
on_kernel_fault
Server, 305
on_kernel_restart
cadabra::NotebookWindow, 259
on_kernel_runstatus
cadabra::GUIBase, 204
cadabra::NotebookWindow, 259
on_key_press_event
cadabra::CodeInput::exp_input_tv, 182
cadabra::NotebookWindow, 259
on_message
cadabra::ComputeThread, 116
Server, 305
test_talk_to_server.cc, 395
on_network_error
cadabra::GUIBase, 204
cadabra::NotebookWindow, 260
cadabra_osx::NotebookWindow, 254
MyClient, 247
on_open
Cadabra, 92
cadabra::ComputeThread, 116
Server, 305
test_talk_to_server.cc, 395
on_prefs_font_size
cadabra::NotebookWindow, 260
on_progress
MyClient, 247
on_run_cell
cadabra::NotebookWindow, 260
on_run_runall
cadabra::NotebookWindow, 260
on_run_runtocursor
cadabra::NotebookWindow, 260
on_run_stop
cadabra::NotebookWindow, 260
on_scroll
cadabra::NotebookWindow, 260
on_scroll_size_allocate
cadabra::NotebookWindow, 260
on_show
cadabra::CodeInput::exp_input_tv, 182
cadabra::TeXView, 358
on_socket_init
Server, 305
on_tex_error
cadabra::NotebookWindow, 260
on_text_scaling_factor_changed
cadabra::NotebookWindow, 260
on_view_close
cadabra::NotebookWindow, 260
on_view_split
cadabra::NotebookWindow, 260
on_vscroll_changed
cadabra::NotebookWindow, 260
one
cadabra, 52
cadabra::sort_spinors, 317
one_index
cadabra::expand, 184
one_step_
cadabra::reduce_delta, 290
only_col_
cadabra::TableauSymmetry, 348
only_column_exchange
cadabra::TableauBase, 346
cadabra::TableauSymmetry, 347
only_expand
cadabra::join_gamma, 231
only_one_on_derivative

cadabra::canonicalise, 95
only_rhs
cadabra::evaluate, 165
open_brackets
preprocessor, 278
open_help
Cadabra, 92
operator!=
cadabra::index_iterator, 215
yngtab::filled_tableau::in_column_iterator, 212
yngtab::filled_tableau::iterator, 228
operator<
cadabra, 52
cadabra::DataCell::id_t, 206
cadabra::str_node, 328
Compare.cc, 429
Compare.hh, 430
Exchange.cc, 435
yngtab::filled_tableau::in_column_iterator, 213
yngtab::filled_tableau::iterator, 228
operator<<
cadabra::TerminalStream, 351
combin, 56
Compare.cc, 429
Compare.hh, 430
PreProcessor.cc, 441
PreProcessor.hh, 442
preprocessor, 277
snoop::Snoop, 309
snoop::SnoopImpl, 313
Stopwatch, 325
Stopwatch.cc, 478
Stopwatch.hh, 479
Storage.cc, 480
Storage.hh, 481
yngtab, 59
operator<=
yngtab::filled_tableau::in_column_iterator, 213
operator>
yngtab::filled_tableau::in_column_iterator, 213
yngtab::filled_tableau::iterator, 228
operator>>
Parser.cc, 438
Parser.hh, 439
PreProcessor.cc, 441
PreProcessor.hh, 442
preprocessor, 277
operator>=
yngtab::filled_tableau::in_column_iterator, 213
operator*
yngtab::filled_tableau::in_column_iterator, 212
yngtab::filled_tableau::iterator, 228
operator()
cadabra::Ex_is_equivalent, 176
cadabra::Ex_is_less, 177
cadabra::indexsort::less_indexed_treenode, 239
cadabra::nset_it_less, 264
cadabra::tree_equal_obj, 361
cadabra::tree_exact_equal_mod_prel_obj, 362
cadabra::tree_exact_equal_obj, 362
cadabra::tree_exact_less_for_indexmap_obj, 363
cadabra::tree_exact_less_mod_prel_obj, 364
cadabra::tree_exact_less_no_wildcards_mod_↩
prel_obj, 364
cadabra::tree_exact_less_no_wildcards_obj, 365
cadabra::tree_exact_less_obj, 366
cadabra::tree_less_modprel_obj, 367
cadabra::tree_less_obj, 367
snoop::Snoop, 309
snoop::SnoopImpl, 313
yngtab::filled_tableau, 195, 196
operator+
Python Core, 33
yngtab::filled_tableau::in_column_iterator, 212
yngtab::filled_tableau::iterator, 228
operator++
cadabra::index_iterator, 215
yngtab::filled_tableau::in_column_iterator, 212
yngtab::filled_tableau::iterator, 228
operator+=
cadabra::index_iterator, 215
yngtab::filled_tableau::in_column_iterator, 212
yngtab::filled_tableau::iterator, 228
operator-
Python Core, 33
yngtab::filled_tableau::in_column_iterator, 212
yngtab::filled_tableau::iterator, 228
operator->
yngtab::filled_tableau::in_column_iterator, 213
yngtab::filled_tableau::iterator, 228
operator--
yngtab::filled_tableau::in_column_iterator, 212
yngtab::filled_tableau::iterator, 228
operator-=
yngtab::filled_tableau::in_column_iterator, 212
yngtab::filled_tableau::iterator, 228
operator=
cadabra::index_iterator, 215
exec_stream_t, 180
yngtab::filled_tableau, 196
yngtab::tableau, 342
operator==
cadabra::Ex, 169
cadabra::index_iterator, 215
cadabra::str_node, 328
combin, 56
ProgressMonitor::Total, 359
yngtab::filled_tableau::in_column_iterator, 213
yngtab::filled_tableau::iterator, 228
operator[]
combin::combinations, 104
combin::symmetriser, 337
yngtab::filled_tableau, 196
order
cadabra::order, 266
preprocessor::accu_t, 62

order_arrow
preprocessor, 276
order_colon
preprocessor, 276
order_comma
preprocessor, 276
order_conditions
preprocessor, 276
order_dot
preprocessor, 276
order_equals
preprocessor, 276
order_factorial
preprocessor, 276
order_frac
preprocessor, 276
order_greater_than
preprocessor, 276
order_labels
preprocessor, 276
order_less_than
preprocessor, 276
order_minus
preprocessor, 276
order_names
preprocessor, 278
order_plus
preprocessor, 276
order_pow
preprocessor, 276
order_prod
preprocessor, 276
order_set_option
preprocessor, 276
order_tilde
preprocessor, 276
order_unequals
preprocessor, 276
order_wedge
preprocessor, 276
orders
preprocessor, 278
ordersign
combin, 56
combin::combinations, 104
original
combin::combinations_base, 108
combin::symmetriser, 338
originals
combin::symmetriser, 338
our_connection_hdl
cadabra::ComputeThread, 117
snoop::SnoopImpl, 314
out
exec_stream_t, 180
out_
cadabra::TerminalStream, 351
snoop::Snoop, 310
outbox
cadabra::VisualCell, 371
output
cadabra::DataCell, 121
cadabra::DisplayBase, 134
Server::Block, 91
output_ipython
PythonCdb.cc, 476
owner_
combin::symm_helper, 333
combin::symm_val_helper, 335

p_components
cadabra::str_node, 327
p_exponent
cadabra::str_node, 327
p_none
cadabra::str_node, 327
p_property
cadabra::str_node, 327
p_sub
cadabra::str_node, 327
p_super
cadabra::str_node, 327
parent_name
cadabra::Indices, 219
parent_rel
cadabra::str_node::flag_t, 198
parent_rel_t
cadabra::str_node, 326
parse
cadabra::Depends, 126
cadabra::DifferentialForm, 132
cadabra::EpsilonTensor, 161
cadabra::FilledTableau, 197
cadabra::GammaMatrix, 201
cadabra::ImplicitIndex, 210
cadabra::Indices, 219
cadabra::Integer, 221
cadabra::InverseMetric, 225
cadabra::LaTeXForm, 238
cadabra::Metric, 246
cadabra::Spinor, 319
cadabra::Tableau, 341
cadabra::TableauSymmetry, 348
cadabra::Weight, 371
cadabra::WeightInherit, 375
cadabra::labelled_property, 237
cadabra::property, 288
Integral, 222
parse_
preprocessor, 277
parse_internal_
preprocessor, 277
parse_one_argument
cadabra::property, 288
parse_python_exception
PythonException.cc, 477
PythonException.hh, 478

parse_to_keyvals
cadabra::property, 288
ParseException, 267
ParseException, 267
ParseExceptionType
PythonCdb.cc, 476
Parser
cadabra::Parser, 269
Parser.cc
operator>>, 438
Parser.hh
operator>>, 439
part1_class
cadabra::split_index, 323
part1_coord
cadabra::split_index, 323
part1_coord_node
cadabra::split_index, 323
part1_is_number
cadabra::split_index, 323
part2_class
cadabra::split_index, 323
part2_coord
cadabra::split_index, 323
part2_coord_node
cadabra::split_index, 323
part2_is_number
cadabra::split_index, 323
parts
cadabra::Parser, 269
preprocessor::accu_t, 62
passing.cc
fun, 439
k, 439
main, 439
pat_prop_pair_t
cadabra::Properties, 282
pats
cadabra::Properties, 283
pattern
cadabra::pattern, 271
pattern_map_t
cadabra::Properties, 282
pclose2
popen2.cc, 389
popen2.hh, 389
pending_actions
cadabra::DocumentThread, 151
Perm, 272
apply, 272
find, 272
perm, 272
perm
Perm, 272
permutation_sign
combin::symmetriser, 338
PermutationException, 273
PermutationException, 273
permute
combin::combinations_base, 107
permute_blocks
combin::symmetriser, 338
permuted_sets_t
combin::combinations, 104
pid
snoop::Snoop::AppEntry, 88
pixbuf
cadabra::TeXView, 358
pm
cadabra::Algorithm, 84
PythonCdb.cc, 476
pointer
yngtab::filled_tableau::iterator_base, 229
pop_history
cadabra::Ex, 169
popen2
popen2.cc, 389
popen2.hh, 389
popen2.cc
pclose2, 389
popen2, 389
READ, 389
WRITE, 389
popen2.hh
pclose2, 389
popen2, 389
port
cadabra::ComputeThread, 117
pos
cadabra::ActionAddCell, 64
cadabra::ActionPositionCursor, 70
pos_tab_t
cadabra::young_project, 378
Position
cadabra::ActionAddCell, 63
cadabra::ActionPositionCursor, 69
position_cursor
cadabra::GUIBase, 204
cadabra::NotebookWindow, 260
cadabra_osx::NotebookWindow, 254
position_t
cadabra::Indices, 218
position_type
cadabra::Indices, 219
cadabra::canonicalise, 95
positive
cadabra::Spinor, 319
possible_singlets
cadabra::exchange, 178
post_order_iterator
cadabra::Algorithm, 79
post_process
cadabra2_defaults, 54
post_process_enabled
PythonCdb.cc, 476
power

cadabra::WeightInherit, 374
pre_clean_dispatch
cadabra, 52
pre_clean_dispatch_deep
cadabra, 52
PreProcessor.cc
operator<<, 441
operator>>, 441
PreProcessor.hh
operator<<, 442
operator>>, 442
tok_arrow, 442
tok_declare, 442
tok_pow, 442
tok_sequence, 442
tok_set_option, 442
tok_siblings, 442
tok_unequals, 442
tok_wedge, 442
preamble_string
cadabra::TeXEngine, 354
PreferencesWindow, 273
mainbox, 274
preferred
cadabra::eliminate_converter, 157
cadabra::rewrite_indices, 295
Prefs
cadabra::NotebookWindow::Prefs, 274
prefs
cadabra::NotebookWindow, 262
preprocessor, 274
accus, 277
bracket_strings_, 276
close_brackets, 277
cur, 277
cur_pos, 278
cur_str, 278
current_bracket_, 276
default_is_product_, 276
eat_initial_whitespace_, 278
erase, 277
get_token_, 277
is_already_bracketed_, 277
is_bracket_, 277
is_closing_bracket_, 277
is_digits_, 277
is_infix_operator_, 277
is_link_, 277
is_opening_bracket_, 277
next_is_product_, 278
open_brackets, 278
operator<<, 277
operator>>, 277
order_arrow, 276
order_colon, 276
order_comma, 276
order_conditions, 276
order_dot, 276
order_equals, 276
order_factorial, 276
order_frac, 276
order_greater_than, 276
order_labels, 276
order_less_than, 276
order_minus, 276
order_names, 278
order_plus, 276
order_pow, 276
order_prod, 276
order_set_option, 276
order_tilde, 276
order_unequals, 276
order_wedge, 276
orders, 278
parse_, 277
parse_internal_, 277
preprocessor, 276
print_stack, 277
show_and_throw_, 277
strip_outer_brackets, 277
unwind_, 277
verbatim_, 278
preprocessor::accu_t, 62
accu, 62
accu_t, 62
bracket, 62
erase, 62
head_is_generated, 62
is_index, 62
order, 62
parts, 62
previous
cadabra::ActionPositionCursor, 69
print
core/CMakeLists.txt, 426
ProgressMonitor, 280
print_arrowlike
cadabra::DisplaySympy, 136
cadabra::DisplayTerminal, 139
cadabra::DisplayTeX, 142
print_children
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_classify_indices
cadabra::Algorithm, 83
print_closing_bracket
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_commalike
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_commutator
cadabra::DisplayTerminal, 140

cadabra::DisplayTeX, 143
print_components
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_conditional
cadabra::DisplayTeX, 143
print_entire_tree
cadabra::Ex, 169
print_equalitylike
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_fraclike
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_ftableau
cadabra::DisplayTeX, 143
print_indexbracket
cadabra::DisplayTeX, 143
print_intlike
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_matrix
cadabra::DisplaySympy, 137
print_multiplier
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_opening_bracket
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_other
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_parent_rel
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_partial
cadabra::DisplaySympy, 137
print_powlike
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_productlike
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 143
print_python
cadabra::Ex, 169
print_recursive_treeform
cadabra::Ex, 170
print_relation
cadabra::DisplayTeX, 143
print_stack
preprocessor, 277
print_star
cadabra::DisplayTeX, 144
print_sumlike
cadabra::DisplaySympy, 137
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 144
print_tableau
cadabra::DisplayTeX, 144
print_tree
PythonCdb.cc, 475
print_wedgeproduct
cadabra::DisplayTeX, 144
procedure_by_name
cadabra::Ex, 170
process_action_queue
cadabra::DocumentThread, 149
process_data
cadabra::GUIBase, 204
cadabra::NotebookWindow, 260
cadabra_osx::NotebookWindow, 254
process_todo_queue
cadabra::NotebookWindow, 260
prod_unwrap_single_term
cadabra::Algorithm, 83
prod_wrap_single_term
cadabra::Algorithm, 83
prodnode
cadabra::product_rule, 279
product_rule
cadabra::product_rule, 279
progress
ProgressMonitor, 280
ServerWrapper, 307
ProgressMonitor, 279
∼ProgressMonitor, 280
call_stack, 280
call_totals, 280
group, 280
print, 280
progress, 280
ProgressMonitor, 280
totals, 280
ProgressMonitor::Block, 90
Block, 90
name, 90
started, 90
step, 90
total_steps, 90
ProgressMonitor::Total, 358
call_count, 359
name, 359
operator==, 359
str, 359
time_spent, 359
time_spent_as_long, 359

Total, 359
total_steps, 359
ProgressMonitor_totals_helper
PythonCdb.cc, 475
Server.cc, 390
progressbar
cadabra::NotebookWindow, 262
project_onto_initial_symmetries
cadabra::decompose_product, 125
projector
yngtab::filled_tableau, 196
projector_normalisation
yngtab::filled_tableau, 196
prop
Property, 285
prop1
cadabra::fierz, 192
prop2
cadabra::fierz, 192
prop3
cadabra::fierz, 192
prop4
cadabra::fierz, 192
propagate_zeroes
cadabra::Algorithm, 83
Properties, 36
properties
cadabra::Ex_comparator, 176
cadabra::Ex_is_equivalent, 176
cadabra::Ex_is_less, 177
cadabra::Kernel, 234
cadabra::index_iterator, 215
cadabra::tree_equal_obj, 361
cadabra::tree_exact_equal_mod_prel_obj, 362
cadabra::tree_exact_equal_obj, 362
cadabra::tree_exact_less_mod_prel_obj, 364
cadabra::tree_exact_less_no_wildcards_mod_↩
prel_obj, 364
cadabra::tree_exact_less_no_wildcards_obj, 365
cadabra::tree_exact_less_obj, 366
cadabra::tree_less_modprel_obj, 367
cadabra::tree_less_obj, 367
Property
for_obj, 285
latex_, 285
prop, 285
Property, 285
repr_, 285
str_, 285
property
cadabra::DocumentThread, 148
cadabra::property, 287
Property< T >, 284
property_map_t
cadabra::Properties, 282
props
cadabra::Properties, 283
ptest.cc
BOOST_PYTHON_MODULE, 469
fun, 469
make_x_and_inject_y, 469
pull_in
Python Core, 33
push_back
cadabra::keyval_t, 235
push_down_multiplier
cadabra, 52
push_history
cadabra::Ex, 170
pushup_multiplier
cadabra::Algorithm, 83
py_what
ArgumentException, 89
InternalError, 224
NonScalarException, 251
NotYetImplemented, 264
python
cadabra::DataCell, 121
Python Core, 31
__eq__Ex_Ex, 31
__eq__Ex_int, 31
call_post_process, 32
create_scope, 32
Ex_latex_, 32
Ex_str_, 32
get_kernel_from_scope, 33
inject_defaults, 33
operator+, 33
operator-, 33
pull_in, 33
python_group
ServerWrapper, 308
python_progress
ServerWrapper, 308
python_recurse
cadabra, 52
PythonCdb.cc
ArgumentExceptionType, 475
BOOST_PYTHON_MODULE, 473
ConsistencyExceptionType, 475
construct_Ex_from_int, 473
construct_Ex_from_int_2, 473
construct_Ex_from_string, 473
construct_Ex_from_string_2, 474
create_empty_scope, 474
create_scope_from_global, 474
createExceptionClass, 474
def_algo_1, 474
def_prop, 474
dispatch_base, 474
dispatch_ex, 474
dispatch_string, 474
Ex_getitem, 474
Ex_getslice, 474
Ex_len, 474
Ex_repr_, 474

Ex_setitem, 474
fetch_from_python, 474
init_ipython, 474
InternalErrorType, 475
lhs, 475
list_properties, 475
make_Ex_from_int, 475
make_Ex_from_string, 475
map_sympy_wrapper, 475
NonScalarExceptionType, 475
NotYetImplementedType, 475
output_ipython, 476
ParseExceptionType, 476
pm, 476
post_process_enabled, 476
print_tree, 475
ProgressMonitor_totals_helper, 475
rhs, 475
RuntimeExceptionType, 476
terms, 475
to_std_vector, 475
translate_ArgumentException, 475
translate_ConsistencyException, 475
translate_InternalError, 475
translate_NonScalarException, 475
translate_NotYetImplemented, 475
translate_ParseException, 475
translate_RuntimeException, 475
PythonCdb.hh
create_empty_scope, 477
create_scope_from_global, 477
Ex_repr_, 477
fetch_from_python, 477
PythonException.cc
parse_python_exception, 477
PythonException.hh
parse_python_exception, 478

queue_action
cadabra::DocumentThread, 149
quit
Server.cc, 390
quit_safeguard
cadabra::NotebookWindow, 261

READ
popen2.cc, 389
range_t
cadabra::Algorithm, 79
combin, 56
range_vector_t
cadabra::Algorithm, 79
combin, 56
rat_set
cadabra, 53
raw_ints
cadabra::sym, 331
read_only
cadabra::NotebookWindow, 262
reads_as_operator
cadabra::DisplayTeX, 144
receive_millis
snoop::Snoop::AppEntry, 88
snoop::Snoop::LogEntry, 241
redo_stack
cadabra::DocumentThread, 151
reduce
cadabra::epsilon_to_delta, 160
reduce_delta
cadabra::reduce_delta, 290
ref
cadabra::ActionAddCell, 64
cadabra::ActionPositionCursor, 70
reference
yngtab::filled_tableau::iterator_base, 229
reference_child_index
cadabra::ActionRemoveCell, 71
reference_parent_cell
cadabra::ActionRemoveCell, 71
refresh_all
cadabra::NotebookCanvas, 252
regexp_tester.cc
main, 390
register_property
cadabra::Properties, 283
registered
cadabra::DocumentThread, 151
registered_properties
cadabra::Properties, 283
regroup_indices_
cadabra::join_gamma, 230
remember_display_hook
cadabra2_defaults, 54
remove_all_cells
cadabra::GUIBase, 204
cadabra::NotebookWindow, 261
cadabra_osx::NotebookWindow, 254
remove_box
yngtab::filled_tableau, 196
yngtab::tableau, 343
yngtab::tableau_base, 344
remove_cell
cadabra::GUIBase, 204
cadabra::NotebookWindow, 261
cadabra_osx::NotebookWindow, 254
remove_empty_nodes
cadabra::Parser, 269
remove_multiplicity_zero
combin::symmetriser, 337
remove_nullifying_traces
yngtab::tableaux, 349
remove_traceless_traces
cadabra::canonicalise, 95
remove_traces
cadabra::young_project, 378
remove_vanishing_numericals
cadabra::canonicalise, 95

remove_zeroed_terms
cadabra::collect_terms, 102
removed_text
cadabra::ActionEraseText, 67
removed_tree
cadabra::ActionRemoveCell, 71
rename_dummies, 291
apply, 292
cadabra::rename_dummies, 293
can_apply, 292
rename_dummies, 292
rename_replacement_dummies
cadabra::Algorithm, 83
repdelta
cadabra::epsilon_to_delta, 160
replace_index
cadabra::Ex, 170
replace_match
cadabra::replace_match, 294
replacement_map
cadabra::Ex_comparator, 176
replacement_map_t
cadabra::Ex_comparator, 172
report
cadabra::Ex_comparator, 175
report_progress
cadabra::Algorithm, 83
report_progress_stopwatch
cadabra::Algorithm, 84
repr_
Property, 285
requests
cadabra::TeXEngine, 354
reset
Stopwatch, 324
reset_state
cadabra::Ex, 170
restart_kernel
cadabra::ComputeThread, 116
restarting_kernel
cadabra::ComputeThread, 117
restore_weights
combin::combinations_base, 107
result_t
cadabra::Algorithm, 79
cadabra::Ex, 167
reuse_generating_set
cadabra::canonicalise, 95
revert
cadabra::ActionAddCell, 64
cadabra::ActionBase, 65
cadabra::ActionEraseText, 67
cadabra::ActionInsertText, 68
cadabra::ActionPositionCursor, 70
cadabra::ActionRemoveCell, 71
cadabra::ActionSetRunStatus, 72
cadabra::ActionSplitCell, 74
rewrite_indices
cadabra::rewrite_indices, 295
rhs
PythonCdb.cc, 475
rhs_contains_dummies
cadabra::substitute, 330
RiemannTensor
cadabra::RiemannTensor, 296
roof
cadabra::index_iterator, 215
row_number
yngtab::filled_tableau::in_column_iterator, 213
yngtab::filled_tableau::iterator, 228
row_size
yngtab::filled_tableau, 196
yngtab::tableau, 343
yngtab::tableau_base, 344
row_stack
yngtab::filled_tableau, 194
rows
yngtab::filled_tableau, 196
yngtab::tableau, 343
rset_t
cadabra, 49
rules
cadabra::take_match, 350
run
cadabra::ComputeThread, 116
Server, 305
UI, 368
run_string
Server, 305
runner
Server, 306
running
cadabra::DataCell, 122
running_cells
cadabra::ComputeThread, 117
RuntimeException, 296
RuntimeException, 297
RuntimeExceptionType
PythonCdb.cc, 476

s_all
exec_stream_t, 179
s_child
exec_stream_t, 179
s_err
exec_stream_t, 179
s_in
exec_stream_t, 179
s_out
exec_stream_t, 179
safestring
Snoop.cc, 392
satisfies_conditions
cadabra::Ex_comparator, 175
save
cadabra::NotebookWindow, 261
Scalar algebra, 34

apply, 34
Ex_to_Sympy, 34
invert_matrix, 34
map_sympy, 34
scale
cadabra::NotebookWindow, 263
scale_
cadabra::CodeInput::exp_input_tv, 182
cadabra::TeXEngine, 354
scoped_lock
ComputeThread.hh, 386
scroll
cadabra::NotebookCanvas, 252
scroll_current_cell_into_view
cadabra::NotebookWindow, 261
scrollbox
NotebookController, 253
seconds
Stopwatch, 324
selectors
cadabra::Ex, 170
selfdual_column
yngtab::tableau_base, 345
send
cadabra2_defaults::Server, 301
Server, 305
send_json
Server, 305
sensitive
cadabra::DataCell, 122
separated_by_derivative
cadabra::Algorithm, 83
seq_numbers_of_first_indices
cadabra::exchange::identical_tensors_t, 207
serial_mutex
cadabra::DataCell, 123
serial_number
cadabra::DataCell, 123
cadabra::Properties, 283
Server, 302
∼Server, 304
architecture, 304
block_available, 306
block_available_mutex, 306
block_queue, 306
catchErr, 306
catchOut, 306
cells_ran, 306
ConnectionMap, 304
connections, 306
current_hdl, 306
current_id, 306
dispatch_message, 304
handles, 304
init, 304
job, 306
main_module, 306
main_namespace, 306
on_block_error, 304
on_block_finished, 304
on_close, 304
on_kernel_fault, 305
on_message, 305
on_open, 305
on_socket_init, 305
run, 305
run_string, 305
runner, 306
send, 305
send_json, 305
Server, 304
server_stopwatch, 306
socket_name, 306
started, 306
stop_block, 305
sympy_stopwatch, 306
wait_for_job, 305
WebsocketServer, 304
ws_mutex, 306
wserver, 306
server
cadabra2_defaults, 54
ServerWrapper, 308
Server.cc
ProgressMonitor_totals_helper, 390
quit, 390
Server::Block, 90
Block, 91
cell_id, 91
error, 91
hdl, 91
input, 91
output, 91
Server::CatchOutput, 95
CatchOutput, 96
clear, 96
collect, 96
str, 96
write, 96
Server::Connection, 117
Connection, 118
hdl, 118
uuid, 118
server_
snoop::SnoopImpl, 314
server_pid
cadabra::ComputeThread, 117
server_status
snoop::Snoop::AppEntry, 88
snoop::Snoop::LogEntry, 241
server_stderr
cadabra::ComputeThread, 117
server_stdout
cadabra::ComputeThread, 117
server_stopwatch
Server, 306

ServerWrapper, 307
group, 307
progress, 307
python_group, 308
python_progress, 308
server, 308
ServerWrapper, 307
set_binary_mode
exec_stream_t, 180
set_buffer_limit
exec_stream_t, 180
set_compute_thread
cadabra::DocumentThread, 150
set_font_size
cadabra::CodeInput, 98
cadabra::TeXEngine, 354
set_geometry
cadabra::TeXEngine, 354
set_image_from_base64
cadabra::ImageView, 208
set_master
cadabra::ComputeThread, 116
set_multiplicity
combin::symmetriser, 337
set_name
cadabra::Indices, 219
cadabra::NotebookWindow, 261
set_names
cadabra::ImplicitIndex, 211
set_progress_monitor
cadabra::Algorithm, 83
set_scale
cadabra::TeXEngine, 354
set_stop_sensitive
cadabra::NotebookWindow, 261
set_sync_immediately
snoop::Snoop, 309
set_text_mode
exec_stream_t, 180
set_title_prefix
cadabra::NotebookWindow, 261
set_unit_sublengths
combin::combinations_base, 107
set_user_details
cadabra::DocumentThread, 150
set_wait_timeout
exec_stream_t, 180
settings
cadabra::NotebookWindow, 263
setup_css_provider
cadabra::NotebookWindow, 261
sh_
combin::symmetriser, 338
shift_enter_pressed
cadabra::CodeInput::exp_input_tv, 182
should_swap
cadabra::Ex_comparator, 175
cadabra::sort_sum, 318
show_and_throw_
preprocessor, 277
show_hide_requested
cadabra::TeXView, 358
sibling_iterator
cadabra::Algorithm, 79
sibtab_t
cadabra::decompose_product, 124
sibtabs_t
cadabra::decompose_product, 124
sign
cadabra::AntiCommuting, 85
cadabra::Commuting, 110
cadabra::CommutingBehaviour, 112
cadabra::NonCommuting, 250
cadabra::SelfAntiCommuting, 298
cadabra::SelfCommuting, 299
cadabra::SelfCommutingBehaviour, 300
cadabra::SelfNonCommuting, 300
cadabra::sym, 331
signal_handler
Cadabra.cc, 488
signal_window
Cadabra.cc, 488
signature
cadabra::InverseMetric, 226
cadabra::Metric, 246
cadabra::epsilon_to_delta, 160
combin::symmetriser, 337
simplify_components
cadabra::evaluate, 165
size
cadabra::AntiSymmetric, 86
cadabra::DAntiSymmetric, 120
cadabra::Derivative, 130
cadabra::KroneckerDelta, 236
cadabra::PartialDerivative, 271
cadabra::SatisfiesBianchi, 297
cadabra::Symmetric, 335
cadabra::TableauBase, 346
cadabra::TableauSymmetry, 348
combin::combinations, 104
combin::symmetriser, 337
size_type
yngtab::filled_tableau::iterator_base, 229
slice_cell
cadabra::CodeInput, 98
SlotSpawnChildSetup
ComputeThread.cc, 385
Snoop
snoop::Snoop, 309
snoop, 57
email, 57
error, 57
fatal, 57
flush, 57
info, 57
log, 57

warn, 58
Snoop.cc
BOOST_SPIRIT_THREADSAFE, 392
safestring, 392
u64_millis, 392
Snoop.hh
LOC, 393
snoop::Flush, 200
snoop::Snoop, 308
∼Snoop, 309
get_user_uuid, 309
impl, 310
init, 309
operator<<, 309
operator(), 309
out_, 310
set_sync_immediately, 309
Snoop, 309
SnoopImpl, 310
sync_immediately_, 310
sync_logs_with_server, 309
sync_runs_with_server, 310
sync_with_server, 310
snoop::Snoop::AppEntry, 87
app_name, 88
app_version, 88
AppEntry, 88
connected, 88
create_millis, 88
from_json, 88
id, 88
ip_address, 88
machine_id, 88
pid, 88
receive_millis, 88
server_status, 88
to_json, 88
user_id, 88
uuid, 88
snoop::Snoop::LogEntry, 240
client_log_id, 241
create_millis, 241
from_json, 241
id, 241
loc_file, 241
loc_line, 241
loc_method, 241
log_id, 241
LogEntry, 241
message, 241
receive_millis, 241
server_status, 241
to_json, 241
type, 241
uuid, 241
snoop::SnoopImpl, 310
∼SnoopImpl, 312
call_mutex, 314
connection, 314
connection_attempt_failed, 314
connection_cv, 314
connection_is_open, 314
connection_mutex, 314
create_tables, 312
db, 314
get_app_registrations, 312
get_user_uuid, 312
id_for_uuid_statement, 314
init, 312
insert_statement, 314
obtain_uuid, 312
on_client_close, 313
on_client_fail, 313
on_client_message, 313
on_client_open, 313
operator<<, 313
operator(), 313
our_connection_hdl, 314
server_, 314
snoop_, 314
SnoopImpl, 312
sqlite_mutex, 314
start_websocket_client, 313
store_app_entry, 313
store_app_entry_without_lock, 313
store_log_entry, 313
sync_logs_with_server, 313
sync_runs_with_server, 314
sync_with_server, 314
this_app_, 314
this_log_, 314
wsclient, 314
wsclient_thread, 314
snoop_
snoop::SnoopImpl, 314
SnoopImpl
snoop::Snoop, 310
snoop::SnoopImpl, 312
SnoopPrivate.hh
message_ptr, 393
WebsocketClient, 393
socket_name
Server, 306
sort_columns
yngtab::filled_tableau, 196
sort_product
cadabra::sort_product, 315
sort_product_
cadabra::substitute, 330
sort_spinors
cadabra::sort_spinors, 316
sort_sum
cadabra::sort_sum, 317
sort_within_columns
yngtab::filled_tableau, 196
spin1

cadabra::fierz, 192
spin2
cadabra::fierz, 192
spin3
cadabra::fierz, 192
spin4
cadabra::fierz, 192
spino
cadabra::exchange::identical_tensors_t, 207
Spinor
cadabra::Spinor, 319
spinor_list
cadabra::fierz, 192
spinordim
cadabra::fierz, 192
split, 320
apply, 321
can_apply, 321
split, 321
split_gamma
cadabra::split_gamma, 321
split_index
cadabra::split_index, 323
sqlite_mutex
snoop::SnoopImpl, 314
stack_mutex
cadabra::DocumentThread, 151
standard_form
yngtab::tableaux, 349
start
exec_stream_t, 180, 181
Stopwatch, 324
start_
combin::combinations_base, 108
start_websocket_client
snoop::SnoopImpl, 313
start_wrap
cadabra::TeXEngine::TeXRequest, 356
started
ProgressMonitor::Block, 90
Server, 306
state
cadabra::Ex, 170
state_
cadabra::Ex, 170
status_label
cadabra::NotebookWindow, 263
NotebookController, 253
status_mutex
cadabra::NotebookWindow, 263
status_string
cadabra::NotebookWindow, 263
statusbarbox
cadabra::NotebookWindow, 263
step
ProgressMonitor::Block, 90
stop
cadabra::ComputeThread, 116
Stopwatch, 324
stop_block
Server, 305
stopit
test_talk_to_server.cc, 395
stopped
Stopwatch, 324
stopped_
Stopwatch, 325
Stopwatch, 324
checkpoint_, 324
diffsec, 325
diffusec, 325
operator<<, 325
reset, 324
seconds, 324
start, 324
stop, 324
stopped, 324
stopped_, 325
Stopwatch, 324
tv2, 325
tz, 325
useconds, 325
Stopwatch.cc
operator<<, 478
Stopwatch.hh
operator<<, 479
storage
combin::combinations, 104
yngtab::tableaux, 349
Storage.cc
operator<<, 480
Storage.hh
operator<<, 481
store
cadabra::Properties::registered_property_map_t, 291
store_app_entry
snoop::SnoopImpl, 313
store_app_entry_without_lock
snoop::SnoopImpl, 313
store_log_entry
snoop::SnoopImpl, 313
str
cadabra::Parser, 269
ProgressMonitor::Total, 359
Server::CatchOutput, 96
str_
Property, 285
str_node
cadabra::str_node, 327
stream_kind_t
exec_stream_t, 179
string2tree
cadabra::Parser, 269
strip_outer_brackets
preprocessor, 277

sub_problem_blocksize
combin::combinations_base, 108
sublengths
combin::combinations_base, 108
combin::symmetriser, 338
sublengths_scattered
combin::symmetriser, 338
substitute
cadabra::substitute, 329
subtree_compare
Expression comparison, 29
subtree_equal
cadabra, 53
subtree_exact_equal
cadabra, 53
subtree_exact_less
cadabra, 53
subtree_less
cadabra, 53
subtree_match
cadabra::Ex_comparator, 173
subtree_replacement_map
cadabra::Ex_comparator, 176
subtree_replacement_map_t
cadabra::Ex_comparator, 172
sum_of_sublengths
combin::combinations_base, 107
supermainbox
cadabra::NotebookWindow, 263
suppress_normal_output
cadabra::Algorithm, 84
svh_
combin::symmetriser, 338
swap_columns
yngtab::filled_tableau, 196
sym
cadabra::sym, 331
cadabra::young_project, 378
cadabra::young_project_tensor, 380
symm_helper
combin::symm_helper, 333
symm_helper< T >
combin::symmetriser, 338
symm_val_helper
combin::symm_val_helper, 334
symm_val_helper< T >
combin::symmetriser, 338
symmap
cadabra::DisplaySympy, 138
cadabra::DisplayTerminal, 140
cadabra::DisplayTeX, 144
symmetrise
yngtab::tableaux, 349
symmetriser
combin::symmetriser, 337
sympy, 58
cadabra2_defaults, 54
sympy_stopwatch
Server, 306
sync_immediately_
snoop::Snoop, 310
sync_logs_with_server
snoop::Snoop, 309
snoop::SnoopImpl, 313
sync_runs_with_server
snoop::Snoop, 310
snoop::SnoopImpl, 314
sync_with_server
snoop::Snoop, 310
snoop::SnoopImpl, 314

t1
cadabra::decompose_product, 125
t1tab
cadabra::decompose_product, 125
t2
cadabra::decompose_product, 125
t2tab
cadabra::decompose_product, 125
tab
cadabra::Ex_comparator, 175
cadabra::exchange::identical_tensors_t, 207
cadabra::young_project, 378
yngtab::filled_tableau::in_column_iterator, 213
yngtab::filled_tableau::iterator, 228
tab1
cadabra::lr_tensor, 243
tab2
cadabra::lr_tensor, 243
tab_basics
cadabra::tab_basics, 340
tab_t
cadabra::TableauBase, 346
tableau_base
yngtab::tableau_base, 344
tableau_container_t
yngtab::tableaux, 349
tabs
cadabra::TableauSymmetry, 348
tabs_to_singlet_rules
cadabra::tab_basics, 340
tabs_to_tree
cadabra::tab_basics, 340
take_match
cadabra::take_match, 350
tb
cadabra::indexsort, 217
cadabra::young_project_tensor, 380
TeXEngine
cadabra::TeXEngine, 353
cadabra::TeXEngine::TeXRequest, 356
TeXException
cadabra::TeXEngine::TeXException, 355
TeXRequest
cadabra::TeXEngine::TeXRequest, 356
TeXView
cadabra::TeXView, 357

temparr
combin::combinations_base, 108
tensors
cadabra::exchange::identical_tensors_t, 207
term_hash
cadabra::collect_terms, 102
cadabra::factor_in, 189
term_hash_iterator_t
cadabra::collect_terms, 102
cadabra::factor_in, 188
term_hash_t
cadabra::collect_terms, 102
cadabra::factor_in, 188
term_ratio
cadabra::Ex_comparator, 176
TerminalStream
cadabra::TerminalStream, 351
terminate
cadabra::ComputeThread, 116
terms
PythonCdb.cc, 475
terms_
cadabra::keep_terms, 232
test
cadabra2_defaults::Server, 301
test_client.cc
client, 394
main, 394
ui, 394
test_permutations.cc
main, 483
test_preprocessor.cc
main, 483
testit, 483
test_talk_to_server.cc
client, 395
main, 395
message_ptr, 395
on_message, 395
on_open, 395
stopit, 395
test_tex.cc
main, 486
testit
test_preprocessor.cc, 483
tex_error
cadabra::TeXView, 358
texify
cadabra::DisplayTeX, 144
text
cadabra::ActionInsertText, 68
textbuf
cadabra::DataCell, 123
this_app_
snoop::SnoopImpl, 314
this_cell
cadabra::ActionEraseText, 67
cadabra::ActionInsertText, 68
cadabra::ActionRemoveCell, 71
cadabra::ActionSetRunStatus, 73
cadabra::ActionSplitCell, 74
this_log_
snoop::SnoopImpl, 314
tight_plus
cadabra::DisplayTeX, 144
tight_star
cadabra::DisplayTeX, 144
time_spent
ProgressMonitor::Total, 359
time_spent_as_long
ProgressMonitor::Total, 359
timeout_t
exec_stream_t, 179
title_prefix
cadabra::NotebookWindow, 263
to
cadabra::Integer, 221
to_factor_out
cadabra::factor_out, 190
to_factor_out_t
cadabra::factor_out, 190
to_json
snoop::Snoop::AppEntry, 88
snoop::Snoop::LogEntry, 241
to_long
cadabra, 53
to_pos
cadabra::ActionEraseText, 67
to_rational
cadabra::Ex, 170
to_right
cadabra::factor_out, 190
to_std_vector
PythonCdb.cc, 475
to_string
cadabra, 53
tok_arrow
PreProcessor.hh, 442
tok_declare
PreProcessor.hh, 442
tok_pow
PreProcessor.hh, 442
tok_sequence
PreProcessor.hh, 442
tok_set_option
PreProcessor.hh, 442
tok_siblings
PreProcessor.hh, 442
tok_unequals
PreProcessor.hh, 442
tok_wedge
PreProcessor.hh, 442
topbox
cadabra::NotebookWindow, 263
Total
ProgressMonitor::Total, 359

total_dimension
yngtab::tableaux, 349
total_permutations
combin::combinations_base, 107
total_steps
ProgressMonitor::Block, 90
ProgressMonitor::Total, 359
totals
ProgressMonitor, 280
tr
cadabra::Algorithm, 84
cadabra::indexsort::less_indexed_treenode, 239
traceless
cadabra::exchange::identical_tensors_t, 207
translate_ArgumentException
PythonCdb.cc, 475
translate_ConsistencyException
PythonCdb.cc, 475
translate_InternalError
PythonCdb.cc, 475
translate_NonScalarException
PythonCdb.cc, 475
translate_NotYetImplemented
PythonCdb.cc, 475
translate_ParseException
PythonCdb.cc, 475
translate_RuntimeException
PythonCdb.cc, 475
tree
cadabra::DisplayBase, 134
cadabra::Parser, 269
tree_equal
cadabra, 53
tree_equal_obj
cadabra::tree_equal_obj, 361
tree_exact_equal
cadabra, 53
tree_exact_equal_mod_prel_obj
cadabra::tree_exact_equal_mod_prel_obj, 362
tree_exact_equal_obj
cadabra::tree_exact_equal_obj, 362
tree_exact_less
cadabra, 53
tree_exact_less_mod_prel_obj
cadabra::tree_exact_less_mod_prel_obj, 364
tree_exact_less_no_wildcards_mod_prel_obj
cadabra::tree_exact_less_no_wildcards_mod_↩
prel_obj, 364
tree_exact_less_no_wildcards_obj
cadabra::tree_exact_less_no_wildcards_obj, 365
tree_exact_less_obj
cadabra::tree_exact_less_obj, 366
tree_less
cadabra, 53
tree_less_modprel_obj
cadabra::tree_less_modprel_obj, 367
tree_less_obj
cadabra::tree_less_obj, 367
tree_to_numerical_tab
cadabra::tab_basics, 340
trim
CodeInput.cc, 488
try_connect
cadabra::ComputeThread, 116
try_spawn_server
cadabra::ComputeThread, 117
tv2
Stopwatch, 325
two
cadabra::sort_spinors, 317
type
snoop::Snoop::LogEntry, 241
tz
Stopwatch, 325

u64_millis
Snoop.cc, 392
UI, 368
run, 368
ui
test_client.cc, 394
uimanager
cadabra::NotebookWindow, 263
uinttab_t
cadabra::tab_basics, 340
uinttabs_t
cadabra::tab_basics, 340
undo
cadabra::DocumentThread, 150
undo_stack
cadabra::DocumentThread, 151
undoable
cadabra::ActionBase, 65
cadabra::ActionSetRunStatus, 73
unichar
cadabra, 53
DisplayTeX.cc, 433
unnamed_argument
cadabra::Depends, 126
cadabra::ImplicitIndex, 210
cadabra::Indices, 219
cadabra::Integer, 221
cadabra::LaTeXForm, 238
cadabra::Weight, 372
cadabra::WeightInherit, 375
cadabra::property, 288
unwind_
preprocessor, 277
unwrap
cadabra::unwrap, 368
update_buffer
cadabra::CodeInput, 98
update_cell
cadabra::GUIBase, 205
cadabra::NotebookWindow, 261
cadabra_osx::NotebookWindow, 254
update_image

cadabra::TeXView, 358
update_state
cadabra::Ex, 170
update_title
cadabra::NotebookWindow, 261
update_weights
combin::combinations_base, 107
use_generalised_delta_
cadabra::join_gamma, 231
use_rule
cadabra::substitute, 330
useconds
Stopwatch, 325
useprops_t
cadabra::Ex_comparator, 173
user_id
snoop::Snoop::AppEntry, 88
utf8_output
cadabra::DisplayTeX, 144
uuid
Server::Connection, 118
snoop::Snoop::AppEntry, 88
snoop::Snoop::LogEntry, 241

validate
cadabra::InverseMetric, 226
cadabra::Metric, 246
cadabra::RiemannTensor, 296
cadabra::WeylTensor, 376
cadabra::property, 288
value
cadabra::Derivative, 130
cadabra::Weight, 372
cadabra::WeightBase, 373
cadabra::WeightInherit, 375
value_
cadabra::Weight, 372
value_permute
combin::symmetriser, 338
value_self
cadabra::WeightInherit, 375
value_type
cadabra::keyval_t, 235
yngtab::filled_tableau, 194
yngtab::filled_tableau::iterator_base, 229
values
cadabra::Indices, 219
values_to_locations
combin::symmetriser, 337
vary
cadabra::vary, 369
vbox
cadabra::TeXView, 358
vector_generated
combin::combinations, 104
combin::combinations_base, 107
combin::symm_helper, 333
combin::symm_val_helper, 334
vector_generated_called_
combin::combinations_base, 108
vector_prod
combin, 56
vector_prod_fact
combin, 56
vector_sum
combin, 56
verbatim
cadabra::DataCell, 121
verbatim_
preprocessor, 278
visualcells
cadabra::NotebookCanvas, 252

WRITE
popen2.cc, 389
WSClient
ComputeThread.hh, 386
wait_for_job
Server, 305
walk
cadabra::index_iterator, 215
warn
snoop, 58
was_running_
cadabra::ActionSetRunStatus, 73
WebsocketClient
SnoopPrivate.hh, 393
WebsocketServer
Server, 304
weight
cadabra::drop_keep_weight, 152
weight_cond
combin::combinations_base, 106
weight_conditions
combin::combinations_base, 108
weight_equals
combin::combinations_base, 106
weight_greater
combin::combinations_base, 106
weight_less
combin::combinations_base, 106
WeightException
cadabra::WeightInherit::WeightException, 373
weights
combin::combinations_base, 108
weights_t
combin, 56
weyl
cadabra::Spinor, 320
WeylTensor
cadabra::WeylTensor, 376
wgh
cadabra::drop_keep_weight, 152
what
exec_stream_t::error_t, 162
width
cadabra::TeXEngine::TeXRequest, 356
width_

cadabra::TeXEngine::TeXRequest, 356
window
AppDelegate(), 87
wrap
cadabra::integrate_by_parts, 223
wrap_scalar_in_components_node
cadabra::evaluate, 165
wrappers
cadabra::unwrap, 369
write
Server::CatchOutput, 96
ws_mutex
Server, 306
wsclient
cadabra::ComputeThread, 117
snoop::SnoopImpl, 314
wsclient_thread
snoop::SnoopImpl, 314
wserver
Server, 306

X, 376

Y, 377
yngint_t
YoungTab.hh, 485
yngrat_t
YoungTab.hh, 485
yngtab, 58
add_box, 59
keeptrack_tab_t, 59
LR_add_box, 59
LR_tensor, 59
legal_box, 59
operator<<, 59
yngtab::filled_tableau
~filled_tableau, 194
add_box, 194
begin, 195
begin_column, 195
box_row, 194
canonicalise, 195
clear, 195
compare_without_multiplicity, 195
copy_shape, 195
end, 195
end_column, 195
find, 195
Garnir_set, 195
has_nullifying_trace, 195
nonstandard_loc, 195
number_of_rows, 195
operator(), 195, 196
operator=, 196
operator[], 196
projector, 196
projector_normalisation, 196
remove_box, 196
row_size, 196
row_stack, 194
rows, 196
sort_columns, 196
sort_within_columns, 196
swap_columns, 196
value_type, 194
yngtab::filled_tableau< T >, 193
yngtab::filled_tableau< T >::in_column_iterator, 211
yngtab::filled_tableau< T >::iterator, 227
yngtab::filled_tableau< T >::iterator_base, 229
yngtab::filled_tableau::in_column_iterator
column_number, 213
filled_tableau< T >, 213
in_column_iterator, 212
operator!=, 212
operator<, 213
operator<=, 213
operator>, 213
operator>=, 213
operator*, 212
operator+, 212
operator++, 212
operator+=, 212
operator-, 212
operator->, 213
operator--, 212
operator-=, 212
operator==, 213
row_number, 213
tab, 213
yngtab::filled_tableau::iterator
column_number, 228
filled_tableau< T >, 228
iterator, 228
operator!=, 228
operator<, 228
operator>, 228
operator*, 228
operator+, 228
operator++, 228
operator+=, 228
operator-, 228
operator->, 228
operator--, 228
operator-=, 228
operator==, 228
row_number, 228
tab, 228
yngtab::filled_tableau::iterator_base
difference_type, 229
iterator_category, 229
pointer, 229
reference, 229
size_type, 229
value_type, 229
yngtab::tableau, 341
~tableau, 342
add_box, 342

clear, 342
number_of_rows, 342
operator=, 342
remove_box, 343
row_size, 343
rows, 343
yngtab::tableau_base, 343
~tableau_base, 344
add_box, 344
add_row, 344
clear, 344
column_size, 344
dimension, 344
hook_length, 344
hook_length_prod, 344
multiplicity, 345
number_of_rows, 344
remove_box, 344
row_size, 344
selfdual_column, 345
tableau_base, 344
yngtab::tableaux
add_tableau, 349
back_insert_iterator, 349
get_back_insert_iterator, 349
remove_nullifying_traces, 349
standard_form, 349
storage, 349
symmetrise, 349
tableau_container_t, 349
total_dimension, 349
yngtab::tableaux< T >, 348
young_project
cadabra::young_project, 378
young_project_product
cadabra::young_project_product, 379
young_project_tensor
cadabra::young_project_tensor, 380
YoungTab.hh
yngint_t, 485
yngrat_t, 485

Z, 381
Z, 381
zero
cadabra, 53
zwnbsp
DisplayTeX.cc, 433