Nuove funzionalit nella versione 2.18 rispetto alla 2.16
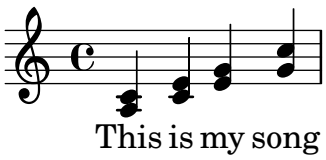
- La funzionalit PostScript di regolazione del tratto non viene pi applicata automaticamente ma lasciata alla discrezione dello strumento PostScript (di default, Ghostscript la usa per le risoluzioni fino a 150dpi quando genera immagini raster). Quando abilitata, un algoritmo di disegno pi complesso, progettato per beneficiare della regolazione del tratto, viene usato soprattutto per i gambi e le stanghette.

  La regolazione del tratto pu essere forzata tramite l'opzione da linea di comando '`-dstrokeadjust`'. Quando si generano file `PDF`, questo produrr solitamente anteprime `PDF` dall'aspetto notevolmente migliore ma anche file di dimensione ben superiore. La qualit della stampa ad alte risoluzioni non ne risente.

- Esiste un nuovo tipo di contesto chiamato `NullVoice` che, sebbene non appaia nell'output, pu essere usato per allineare il testo vocale. Ci particolarmente utile se usato in parallelo a un costrutto `\partcombine`.

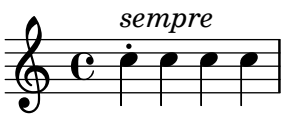
```
soprano = \relative c' { c e g c }
alto = \relative c' { a c e g }
verse = \lyricmode { This is my song }

\score {
  \new Staff <<
    \partcombine \soprano \alto
    \new NullVoice = "aligner" \soprano
    \new Lyrics \lyricsto "aligner" \verse
  >>
  \layout {}
}
```



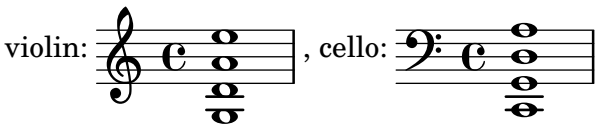
- Varie articolazioni possono essere inserite in una sola variabile o restituite da una funzione di tipo evento:

```
sempreStacc = -. ^\markup \italic sempre
\relative { c''4\sempreStacc c c c }
```

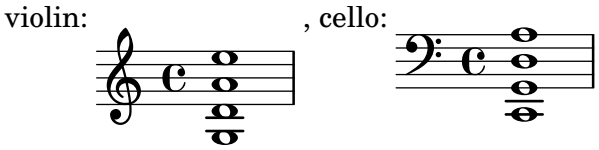


- La linea di base delle partiture interne a un blocco \score viene presa ora dal punto di riferimento (di solito il centro del rigo) del primo sistema invece che dal punto pi alto del rettangolo confinante. L'esempio seguente

```
\markup {
  violin: \score { \new Staff { <g d' a' e''>1 }
                   \layout { indent=0 } } ,
  cello: \score { \new Staff { \clef "bass" <c, g, d a> }
                  \layout { indent=0 } }
}
```



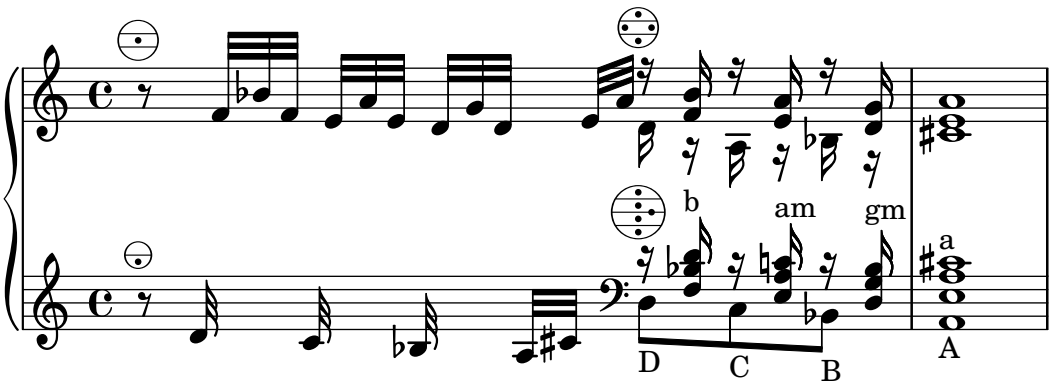
precedentemente appariva cos
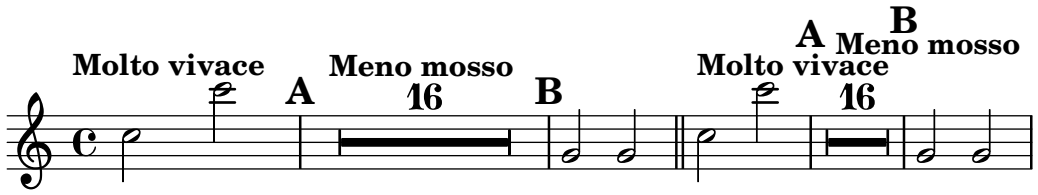


e non c'era un modo affidabile per allineare le due partiture.

- LilyPond non inferisce pi automaticamente un contesto '`\defaultchild`' in una definizione di contesto che ha delle clausole '`\accepts`'. Qualsiasi definizione di contesto priva di una definizione esplicita o ereditata di '`\defaultchild`' viene considerata un contesto '`Bottom`' e sar idonea per eventi ritmici e sovrascritture senza causare la creazione implicita di altri contesti. Assicurati di specificare un '`\defaultchild`' per i contesti diversi da '`Bottom`', se li definisci da zero.
- Ora sono completamente supportati i simboli di registro basso e discant per fisarmonica nel modulo '`scm accreg`', vedi Sezione "Accordion Registers" in *Guida alla Notazione*.

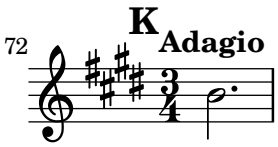
```
#(use-modules (scm accreg))
\new PianoStaff
<<
  \new Staff \relative
  { \clef "treble"  \discant "10"
    r8 s32 f'[ bes f] s e[ a e] s d[ g d] s16 e32[ a]
    \discant "121"
    << { r16 <f bes> r <e a> r <d g> } \\
       { d r a r bes r } >> |
    <cis e a>1
  }
  \new Staff \relative
  { \clef "treble"  \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef "bass"  \stdBass "Master"
    << { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
         <e a cis>1^"a" } \\
       { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```



- I nuovi comandi `markLengthOn` e `markLengthOff` controllano la spaziatura orizzontale dei segni di tempo e di chiamata.



- I segni di chiamata all'inizio di una linea sono ora posizionati a destra della chiave e dell'armatura di chiave. Come nelle versioni precedenti, `break-alignable-interface` ne controlla il comportamento.



- I numeri decimali possono ora essere scritti direttamente nella musica, senza il segno di cancelletto. Grazie alla precedente modifica del modo in cui le propriet di un oggetto sono specificate, il codice per cambiare la lunghezza dei gambi cambiato da cos:

```
\override Stem #'length = #5.6
e' f' g' a'
```

a cos:

```
\override Stem.length = 5.6
e' f' g' a'
```

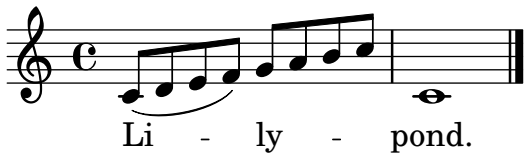
Bisogna scrivere un numero a entrambi i lati del punto – valori come `4.` o `-.3` non sono permessi.

Le frazioni decimali non sono accettate in modalit `\chordmode`.

- Varie abbreviazioni come `(`, `)`, `|`, `[`, `]`, `~`, `\(`, `\)` e altre possono ora essere ridefinite liberamente come normali comandi. Ad esempio

```
"\\{" = (
"\\}" = )
"(" = \melisma
")" = \melismaEnd
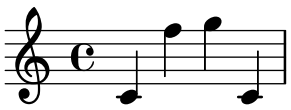
\new Staff <<
  \relative c' {
    c8 \{ d e f \} % con legatura di portamento
    g ( a b c ) % nessuna legatura, c' il melisma
    c,1 \bar "|."
   }
   \addlyrics { Li -- ly -- pond. }
>>
```



- La forma abbreviata dell'articolazione `\staccatissimo` stata rinominata da `-|` a `-!`.

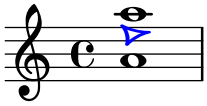
- L'ampiezza della variazione di tempo ora si scrive `\tempo 4 = 60 - 68` invece di `\tempo 4 = 60 ~ 68`.
- Il grob `OctavateEight` stato rinominato `ClefModifier`. Le relative propriet di contesto sono state rinominate da `xxxOctavationyyy` a `xxxTranspositionyyy`.
- C' un nuovo comando `\absolute` che contrassegna in modo esplicito il fatto che la musica sia inserita con altezze assolute. Sebbene quella assoluta sia la modalit di inserimento implicita predefinita, un comando esplicito `\absolute` impedisce la reinterpretazione quando il passaggio posto dentro un blocco `\relative`:

```
\relative c { c'4 \absolute { f'' g'' } c }
```
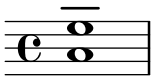


- Se si usa `\relative` senza un'esplicita altezza di riferimento, ora l'altezza di riferimento il centro della prima ottava, rendendo la prima altezza inserita indistinguibile dall'altezza assoluta. Precedentemente, se si ometteva l'altezza di riferimento veniva presa come riferimento l'altezza `c'`. Dato che questa scelta era in qualche modo arbitraria, si consigliava di specificare sempre l'altezza di riferimento.
- Si pu usare un nuovo comando `\single` per convertire la sovrascrittura di una propriet in una modifica da applicare a una singola espressione musicale:

```
<a \single\voiceTwoStyle e' a>1
```
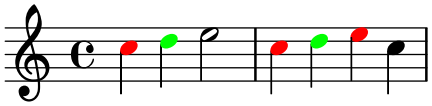


- I due modi per non far apparire un oggetto grafico nell'output sono sovrascrivere la sua propriet `transparent` con `#t` (mantenendo la spaziatura originaria) o sovrascrivere la sua propriet `stencil` con `#f` (per non occupare alcun spazio). Queste due operazioni hanno ora delle forme abbreviate chiamate rispettivamente `\hide` (nascondi) e `\omit` (ometti). Prendono come argomento un'espressione musicale da modificare o il nome di un oggetto grafico per cui si deve creare una sovrascrittura (per specificare entrambi, usare `\single` nella forma con override):

```
\new Staff \with { \omit Clef }
\relative c'' <a e' \hide a>1
```



- Un nuovo comando `\temporary` pu essere applicato alle sovrascritture in modo che queste non sostituiscano precedenti impostazioni delle propriet. Se successivamente viene applicato un `\revert` alla stessa propriet, riappare l'impostazione precedente:

```
\override NoteHead.color = #red c4
\override NoteHead.color = #green d
\revert NoteHead.color e2
\override NoteHead.color = #red c4
\temporary\override NoteHead.color = #green d
\revert NoteHead.color e
\revert NoteHead.color c
```



Questo utile principalmente per scrivere funzioni musicali che hanno bisogno di modificare alcune propriet solo per la durata della funzione.

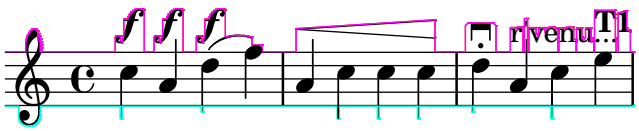
- `\tag`, `\removeWithTag` e `\keepWithTag` possono ora accettare una lista di simboli invece di un solo simbolo per contrassegnare, togliere o mantenere la musica usando un qualsiasi numero di etichette (tag). Ci importante specialmente per `\keepWithTag` dato che non si pu ottenere lo stesso risultato usando consecutivamente molteplici comandi `\keepWithTag`.
- L'opzione '`-d old-relative`' stata tolta. Non pi veramente accessibile dalla linea di comando, veniva usata soltanto per interpretare `\relative` nei file LilyPond convertiti automaticamente dalle versioni 1.8 o precedenti. Non chiaro quanto di questo fosse ancora realmente funzionante.
- Il significato di `instrumentTransposition` stato rovesciato. Dopo

  ```
  \set instrumentTransposition = #{ b #}
  ```

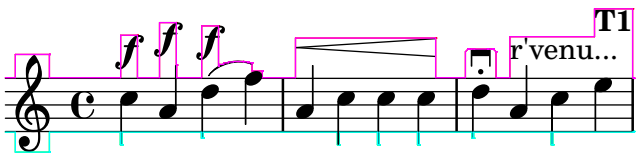
  un `c'` ora suona come un `b`. Precedentemente, sarebbe stato l'esatto contrario. Questo e il cambiamento seguente dovrebbero rendere pi facile il lavoro con gli strumenti traspositori.
- La musica generata dai comandi `\set` e `\override` non pi influenzata da `\transpose`. La conseguenza principale che `\transpose` trasporr l'intonazione reale e quella a stampa della stessa quantit anche quando la musica trasposta contiene `\transposition`. Precedentemente

  ```
  \transpose c' f' \transposition bes'
  ```

  era equivalente a `\transposition f'`. Ora resta equivalente a `\transposition bes'`.
- Quando fa dei controlli per le collisioni, LilyPond non considera pi gli oggetti come rettangoli. Invece approssima la forma reale degli oggetti usando un approccio simile ai numeri interi. Questo produce generalmente un posizionamento degli oggetti e dei sistemi pi uniforme e compresso:
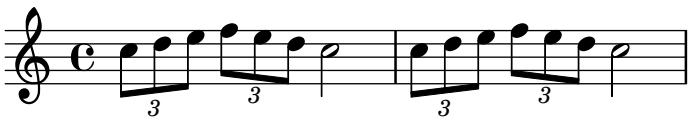


Precedentemente, il frammento precedente appariva cos:



Gli oggetti interessati comprendono `Accidentals`, `Beams`, `Clefs`, `Dynamics`, `FiguredBass`, `Flags`, `Glissandos`, `Lyrics`, `MetronomeMarks`, `OttavaBrackets`, `Pedals`, `RehearsalMarks`, `Rests`, `Scripts`, `TextScripts`, `Ties`, `Tuplets` e `VoltaBrackets`.

- I gruppi irregolari sono ora creati col comando `\tuplet`, che prende come argomento una frazione `t/n` per specificare che t note sono suonate nel tempo solitamente concesso a n note. Un comando `\tuplet` pu creare vari gruppi irregolari se la loro durata inserita dopo la frazione.

  ```
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { f e d } c2
  \tuplet 3/2 4 { c8 d e f e d } c2
  ```



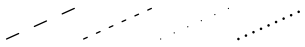
Il comando `\times` con la sua frazione invertita `n/t` ancora disponibile.

- Vengono introdotti due nuovi comandi markup: `\draw-dashed-line` e `\draw-dotted-line`.

  La linea tratteggiata (dashed-line) si estende per l'intera lunghezza data da *dest*, se `full-length` impostato su `#t` (predefinito) senza alcuno spazio all'inizio o alla fine. `off` viene poi modificato per entrarci. Per insistere sui valori assegnati (o predefiniti) di `on`, `off` usare `\override #'(full-length . #f)`. Sono possibili anche le impostazioni manuali per `on`, `off` e `phase`.
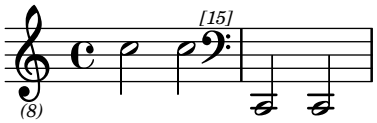
  La linea puntata (dotted-line) si estende sempre per l'intera lunghezza data da *dest*, senza alcun spazio all'inizio o alla fine. Sono possibili le impostazioni manuali per `off` per definire uno spazio pi grande o pi piccolo tra i punti. Il valore assegnato (o predefinito) di `off` verr modificato per riempire la lunghezza della linea.

```
\markup {
  \draw-dashed-line #'(5.1 . 2.3)
  \override #'(on . 0.3)
  \override #'(off . 0.5)
  \draw-dashed-line #'(5.1 . 2.3)
  \draw-dotted-line #'(5.1 . 2.3)
  \override #'(thickness . 2)
  \override #'(off . 0.2)
  \draw-dotted-line #'(5.1 . 2.3)
}
```



- A partire dalla versione 2.17.10, i messaggi di errore e gli URI `textedit` usati per la funzionalit punta-e-clicca specificano numeri di colonna che iniziano con 1 invece che con 0. L'offset di byte (anch'esso parte degli URI `textedit`) inizia ancora a 0.
- Il comando `\clef` supporta la trasposizione:

```
\clef "treble_(8)"
c2 c
\clef "bass^[15]"
c2 c
```



- La sintassi LilyPond delle parole separate da punto `Voice.Accidental` stata resa intercambiabile con `#'(Voice Accidental)`, una lista Scheme di simboli. Una delle conseguenze che codice come

```
\override Voice.TextSpanner #'(bound-details left text) = "rit."
```

  equivale ora a

```
\override Voice.TextSpanner bound-details.left.text = "rit."
```

  o anche a

```
\override #'(Voice TextSpanner) bound-details.left.text = "rit."
```

- Non pi necessario specificare il percorso del grob e della propriet del grob in due argomenti separati per comandi come '`\override`' e `\revert`, rendendo possibile la sintassi

```
\override Voice.TextSpanner.bound-details.left.text = "rit."
```

Dato che funzioni musicali complementari come '`\overrideProperty`' non possono supportare forme con e senza spazio di separazione allo stesso tempo, l'uso del percorso unico separato da punti la forma preferita. Specificare separatamente il percorso del grob e quello della sua propriet, attualmente ancora supportato con '`\override`' e '`\revert`' per ragioni di compatibilit, deprecato.
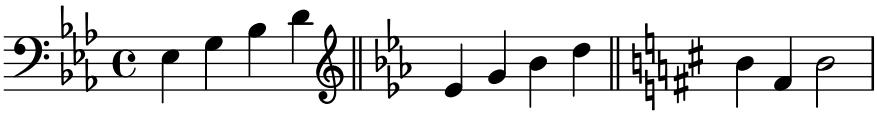
- Dal momento che le parole ora vengono accettate come argomenti di funzioni di simbolo, le interfacce di '`\accidentalStyle`', '`\alterBroken`', '`\footnote`' e '`\tweak`' sono state ridefinite quando erano presenti degli argomenti opzionali di simbolo. Controllare la documentazione di ciascuna funzione musicale per conoscere i dettagli.
- Vari comandi ora accettano liste di simboli (inseriti opportunamente come parole separate da punti) per vari tipi di argomenti. Questi comprendono '`\accidentalStyle`', '`\alterBroken`', '`\footnote`', '`\hide`', '`\omit`', '`\overrideProperty`', '`\shape`' e '`\tweak`'.
- L'interfaccia utente della stanghetta cambiata. I glifi della battuta ora rispecchiano l'aspetto della stanghetta, quindi un segno di ripetizione posto a sinistra deve essere inserito con `.|:`. Il comando `\defineBarLine` offre un modo semplice per definire ulteriori stili per le stanghette.
- Le alterazioni nell'armatura di chiave possono essere stampate in ottave diverse dalle loro posizioni tradizionali, oppure in ottave multiple.

