

0.1 Introduction

This document presents a brief overview of LilyPond features. When the text correspond with the shown notation, we consider LilyPond Officially BugFree (tm). This document is intended for finding bugs, and documenting bugfixes.

TODO: order of tests (file names!), test only one feature per test. Smaller and neater tests.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/+.ly’:`

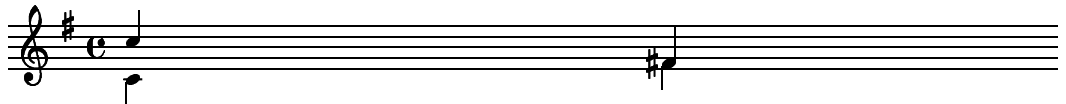
Automatic beamer behaves nicely. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/abe.ly’:`



Cautionary accidentals are indicated using either smaller accidentals (default) or parentheses. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/accidental-cautionary.ly’:`



If two forced accidentals happen at the same time, only one sharp sign is printed. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/accidental-double.ly’:`



Ledger lines are shortened when there are accidentals `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/re`



This shows how accidentals in different octaves are handled. FIXME: Shorten and docu

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/accidental-octave.ly’:`

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

15 \property Score.autoAccidentals = #'(Staff (same-octave . 0))

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

29 \property Score.autoAccidentals = #'(Staff (same-octave . 1))

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

43 \property Score.autoAccidentals = #'(Staff (any-octave . 0))

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

57 \property Score.autoAccidentals = #'(Staff (any-octave . 1))

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

71 \modernAccidentals

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

85 \modernCautionaries

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

99 \noResetKey

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

113 \forgetAccidentals

gis g g gis gis g g gis g gis g fis f f fis fis f f fis f fis

Accidentals are placed as closely as possible. Accidentals in corresponding octaves are aligned. The top accidental should be nearest to the chord. The flats in a sixth should be staggered. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/accidental-placement.ly’:`



A sharp sign after a double sharp sign, as well as a flat sign after a double flat sign is automatically prepended with a natural sign. `/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/acc`



gisis' gis' gisis' ges' geses' ges' geses' gis' gisis' g' geses' g' gis' ges' gis' g'



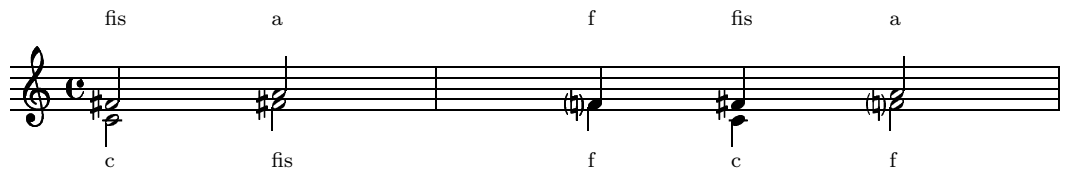
gisis' gis' gisis' ges' geses' ges' geses' gis' gisis' g' geses'g' gis' ges' gis' g'

Tied accidentaled notes (which cause reminder accidentals) don't wreak havoc in the spacing when unbroken. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/accidental-unbrok`

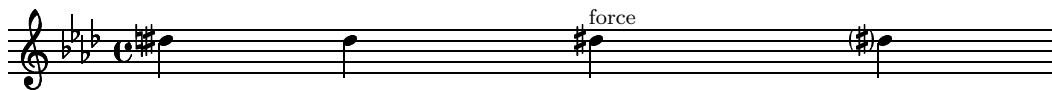


This shows how modern cross voice auto cautionary accidentals are handled. The first two fisses get accidentals because they belong to different voices. The first f gets cautionary natural because of previous measure. The last f gets cautionary natural because fis was only in the other voice.

```
'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/accidental-voice.ly':
```



Accidentals work: the second note does not get a sharp. The third and fourth show forced and courtesy accidentals. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/accidental.ly’`:



dis" dis" dis" dis"

This shows how accidentals are handled. `/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/a`



d dis dis dis d d dis disisd dis d des disisd d dis desesd dis dis dis disd dis



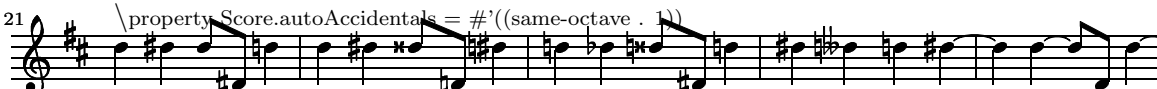
dis dis cis c c cis cisis cis c ces cisis c cis cesesc cis cis cis cis cis



d dis dis dis d d dis disisd dis d des disisd d dis desesd dis dis dis disd dis



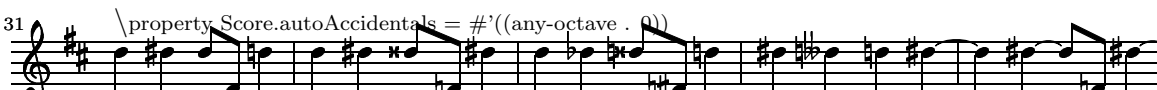
dis dis cis c c cis cisis cis c ces cisis c cis cesesc cis cis cis cis cis



d dis dis dis d d dis disisd dis d des disisd d dis desesd dis dis dis disd dis



dis dis cis c c cis cisis cis c ces cisis c cis cesesc cis cis cis cis cis



d dis disdis d d dis disisd dis d des disisd d dis desesd dis dis dis dis d dis



dis dis cis c c cis cisis cis c ces cisis c cis cesesc cis cis cis cis cis

41 `\property Score.autoAccidentals = #'((any-octave . 1))`

d dis disdis d d dis disisd dis d des disisd d dis desesd dis dis dis dis d dis

46

dis dis cis c c cis cisis cis c ces cisis c cis cesesc cis cis cis cis cis

51 `\modernAccidentals`

d dis disdis d d dis disisd dis d des disisd d dis desesd dis dis dis dis d dis

56

dis dis cis c c cis cisis cis c ces cisis c cis cesesc cis cis cis cis cis

61 `\modernCautionaries`

d dis disdis d d dis disisd dis d des disisd d dis desesd dis dis dis dis d dis

66

dis dis cis c c cis cisis cis c ces cisis c cis cesesc cis cis cis cis cis

71 `\noResetKey`

d dis disdis d d dis disisd dis d des disisd d dis desesd dis dis dis dis d dis

76

dis dis cis c c cis cisis cis c ces cisis c cis cesesc cis cis cis cis cis

81 `\forgetAccidentals`

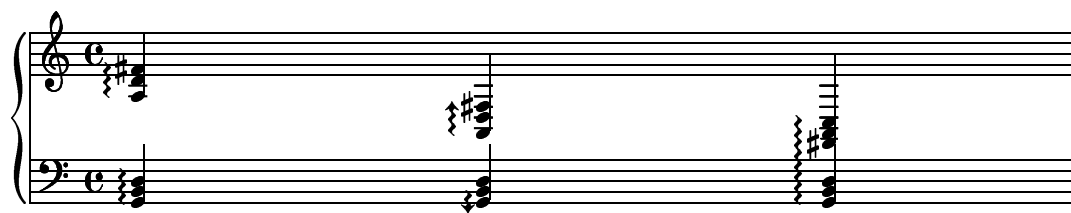
d dis disdis d d dis disisd dis d des disisd d dis desesd dis dis dis dis d dis



Arpeggio stays clear of accidentals and flipped note heads. Since Arpeggio engraver is Voice, it does nothing for voice collisions. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/arp`



Arpeggios are supported, both cross-staff and broken single staff. `‘/home/inger/RPM/BUILD/lilypond-1.6.`



The first two a8 notes should not be beamed. Also, no automatic beaming accross bar lines. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/auto-beam-bar.ly’:`



Tuplet-spanner should not put (visible) brackets on beams even if they’re auto generated. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/auto-beam-tuplets.ly’:`



Auto change piano staff switches voices between up and down staves automatically rests are switched along with the coming note. When central C is reached, we don’t switch (by default).

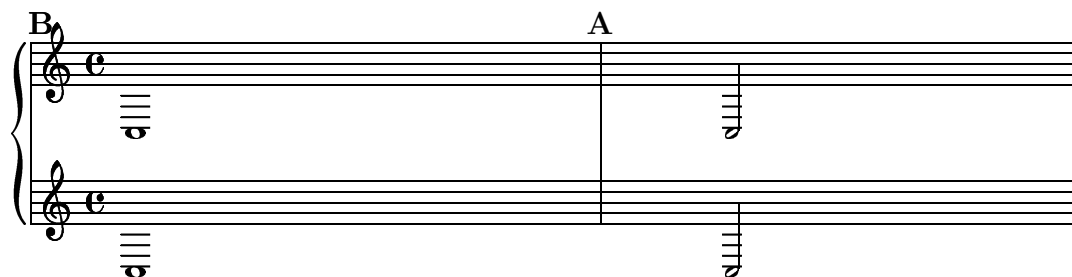
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/auto-change.ly’:`



Bar number settable and padding adjustable. Bar numbers start counting after the anacrusis. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/bar-number.ly’:`



Markings that are attached to (invisible) barlines are delicate: the are attached to the rest of the score without the score knowing it. Consequently, they fall over often. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/bar-scripts.ly’:`



Automatic kneeling. A knee is made when a horizontal beam fits in a gap between note heads that is larger than a predefined threshold. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-auto-knee.ly’:`



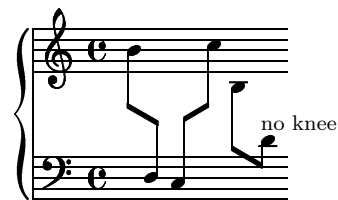
Simple beams on middle staffline be allowed to be slightly sloped. Beams reaching beyond middle line can have bigger slope. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-cent`



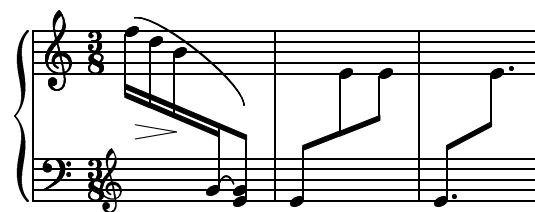
Concave beams should be horizontal. informally spoken, concave refers to the shape of the notes that are opposite a beam. If an up-beam has high notes on its center stems, then we call it concave. This example shows borderline cases. Only the beams that are marked ‘horiz’ should be printed horizontally. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-concave.ly’:`



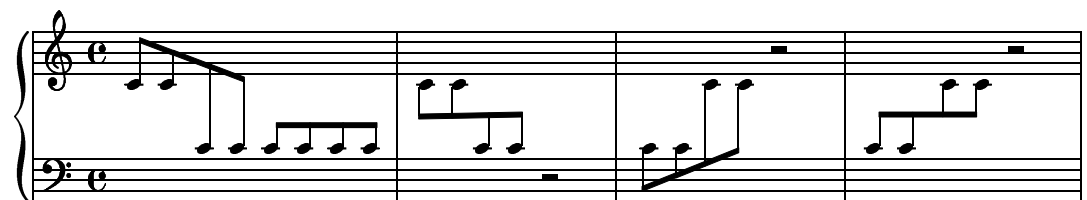
Automatic cross-staff knees also work (here we see them with explicit staff switches).
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-cross-staff-auto-knee.ly’:



Cross staff (kneed) beams don’t cause extreme slopes ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/re



Beams can be typeset over fixed distance aligned staves, beam beautification doesn’t really work, but knees do. Beams should behave well, wherever the switching point is.
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-cross-staff.ly’:



Beamed stems have standard lengths if possible. ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regre



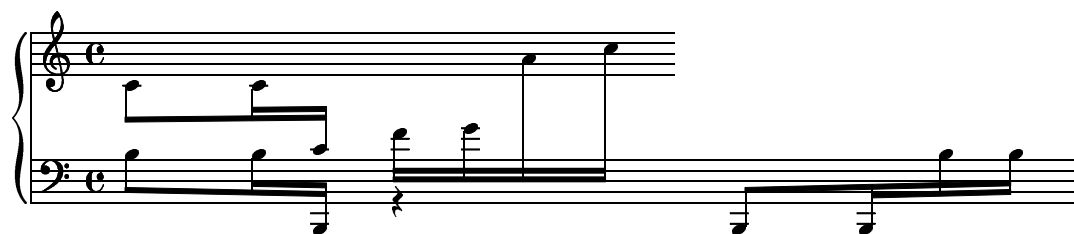
Beams should behave reasonably well, even under extreme circumstances. Stems may be short, but noteheads should never touch the beam. Note that under normal circumstances, these beams would get knees here Beam.auto-knee-gap was set to false.

[illegible]

The first staff of music is written on a five-line treble clef. It begins with a common time signature (C). The melody consists of a series of eighth notes: C4, D4, E4, F4, G4, A4, B4, and C5. The notes are beamed in groups of four, with a slight upward curve over the final two notes of each group.

Kneed beams (often happens with cross-staff beams) should look good when there are multiple beams: all the beams should go on continuously at the staff change. Stems in both staves reach up to the last beam.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-multiple-cross-staff.ly’:`



explicit beams may cross barlines. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-o`



Beams should always reach the middle staff line. The second beam counting from the note head side, should never be lower than the second staff line. This does not hold for grace note beams. Override with `noStemExtend`.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-position.ly’:`



Quarter notes may be beamed: the beam is halted momentarily. `‘/home/inger/RPM/BUILD/lilypond-1.6.6`

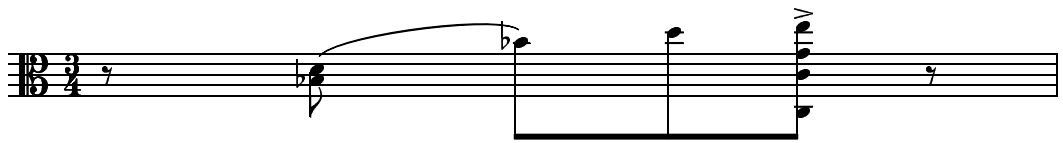


Beams in unnatural direction, have shortened stems, but do not look too short.

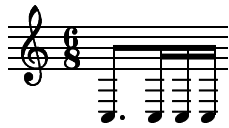
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beam-shortened-lengths.ly’:`



Hairy case for beam, chord, and automatic knees. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regre`



automatic beaming also works in ternary time sigs. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/reg`



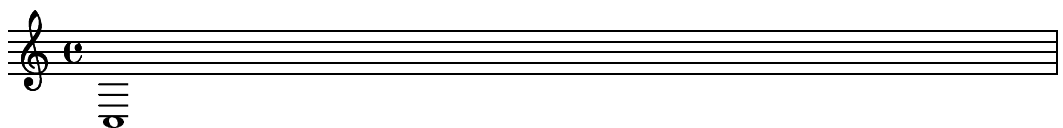
Beaming is generated automatically. Beams may cross bar lines. In that case, line breaks are forbidden. Yet clef and key signatures are hidden just as with breakable bar lines. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beaming.ly’:`



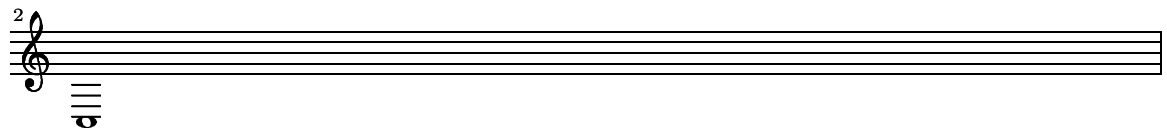
beams (simple) `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/beams.ly’:`



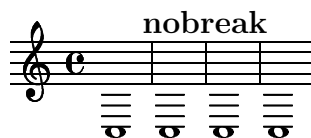
The same mechanism can be used to force pagebreaks. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/r`



(pagebreak)



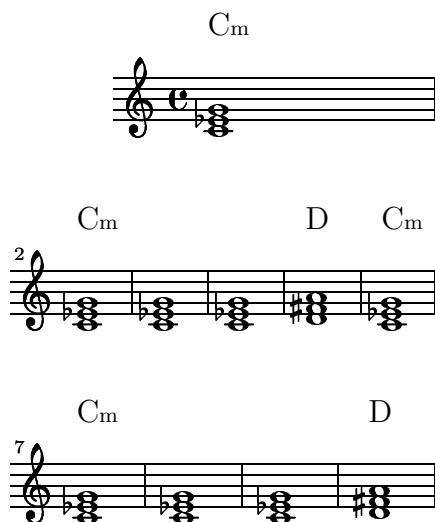
Breaks can be encouraged and discouraged using `\break` and `\noBreak`. They are abbrevs for `\penalty` commands. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/break.ly’:`



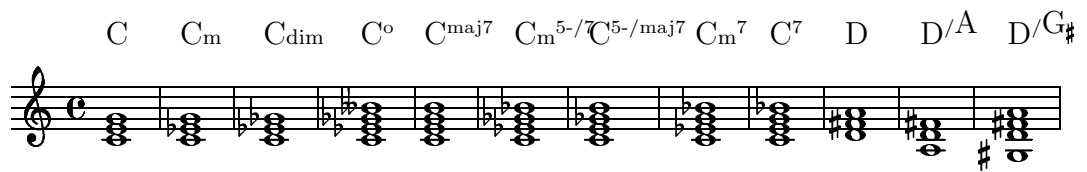
Breathing signs, also used for phrasing, do normally not influence global spacing – only if space gets tight, notes are shifted to make room for the breathing sign. Breathing signs break beams running through their voice. In the following example, the notes in the first two measures all have the same distance from each other: `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/breathing-sign.ly’`:



property `chordChanges`: only display chord names when there’s a change in the chords scheme, but always display the chord name after a line break. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/chord-changes.ly’`:

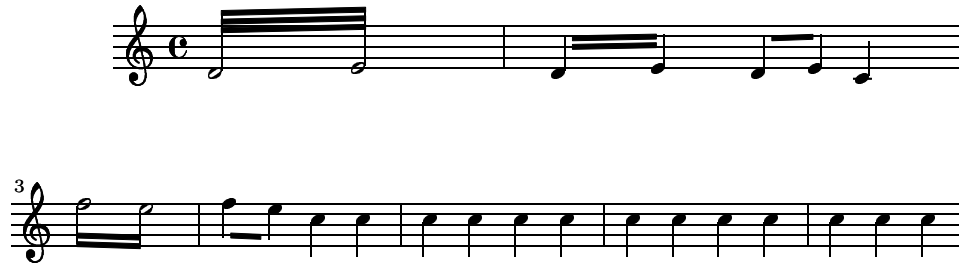


Chord names are generated from a list pitches, and are customisable from `guile`. For some unlogical names, `guile` customisation is used by default. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/chord-names.ly’`:



Chord tremolos look like beams, but are a kind of repeat symbol. To avoid confusion, chord tremolo beams do not reach the stems, but leave a gap. Chord tremolo beams on half notes are not ambiguous, as half notes cannot appear in a regular beam, and should reach the stems.

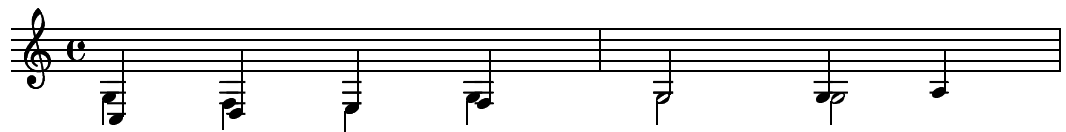
(To ensure that the spacing engine is not confused we add some regular notes as well.)
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/chord-tremolo.ly’:



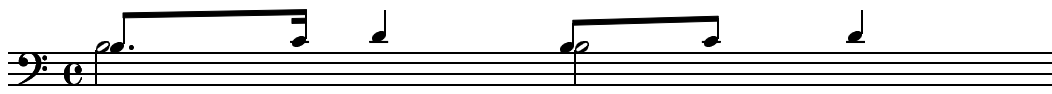
The transparent clef should not occupy any space and with style `fullSizeChanges`, the changing clef should be typeset in full size. For octaviated clefs, the “8” should appear closely above or below the clef respectively. The “8” is processed in a convoluted way, so this is fragile as well. ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/clefs.ly’:



‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/collision-2.ly’:



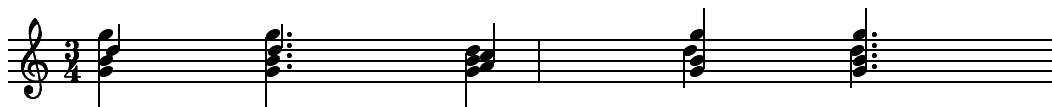
collision resolution tries to put notes with dots on the right side. ‘/home/inger/RPM/BUILD/lilypond-1.6.6



If merge-differently-headed, then open note heads may be merged with black noteheads, but only if the black note heads are from 8th or shorter notes.
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/collision-heads.ly’:



Oppositely stemmed chords, meshing into each other, are resolved.
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/collision-mesh.ly’:



Normal collisions. We have support for polyphony, where the middle voices are horizontally shifted.
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/collisions.ly’:



Completion heads are broken across bar lines. This was intended as a debugging tool, but it can be used to ease music entry. Completion heads are not fooled by polyphony with a different rhythm.
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/completion-heads-polypho’:



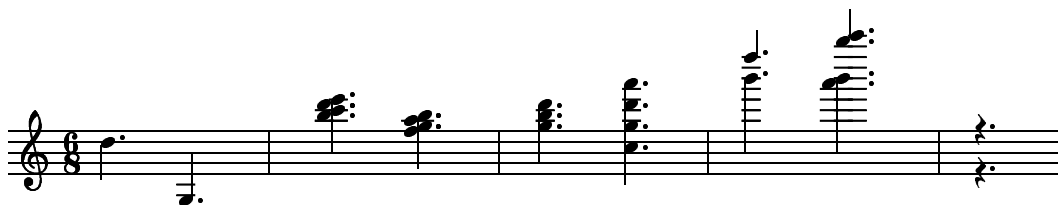
If the Note_heads_engraver is replaced by the Completion_heads_engraver, notes that cross bar lines are split into tied notes.
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/compl’:



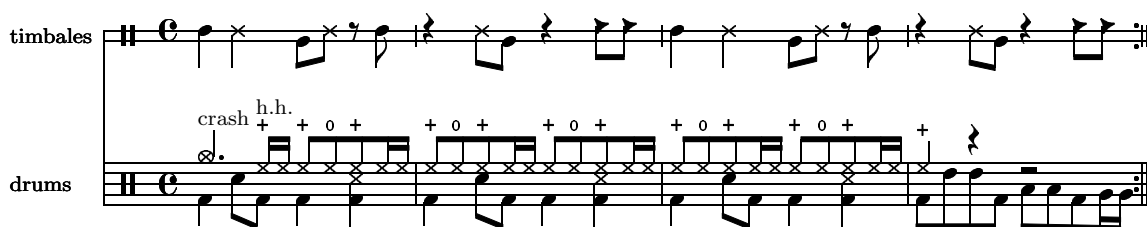
Dots move to the right when a collision with the (up)flag happens
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/dot-flag-collision.ly’:



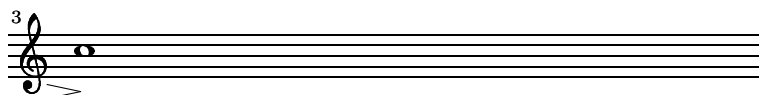
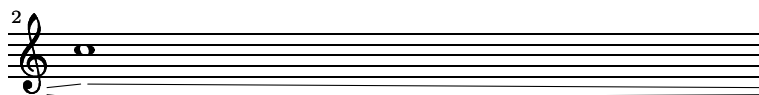
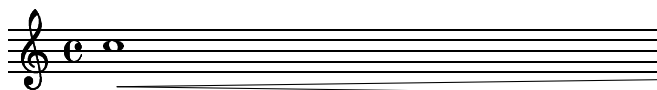
Noteheads can have dots, and rests can too. Augmentation dots should never be printed on a staff line, but rather be shifted vertically. They should go up, but in case of multiple parts, the down stems have down shifted dots. (Wanske p. 186) In case of chords, all dots should be in a column. The dots go along as rests are shifted to avoid collisions. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/dots.ly’:`



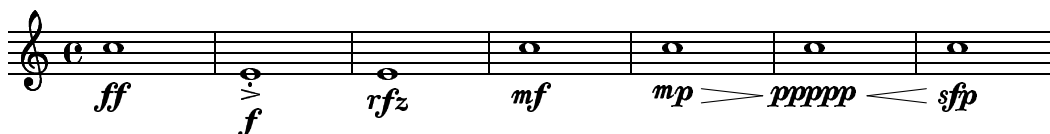
Drum notation, although kludgy, should work. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/dots.ly’:`



Broken crescendi should look be open on one side. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/dots.ly’:`

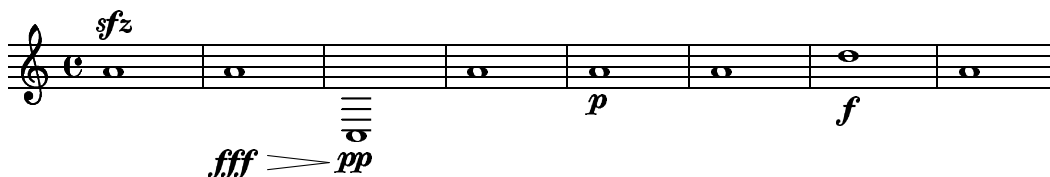


Dynamic letters are kerned, and their weight matches that of the hairpin signs. The dynamic scripts should be horizontally centered on the note head. Scripts that should appear closer to the note head (staccato, accent) are reckoned with. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/dynamics-glyphs.ly’:`



Dynamics appear below or above the staff. If multiple dynamics are linked with (de)crescendi, they should be on the same line. Isolated dynamics may be forced up or down.

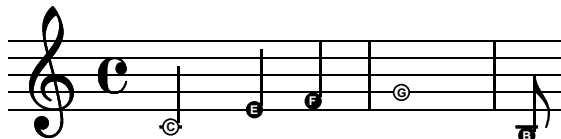
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/dynamics-line.ly’:`



Crescendi may start off-notes. In that case, they should not collapse into flat lines.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/dynamics-unbound-hairpin.ly’:`



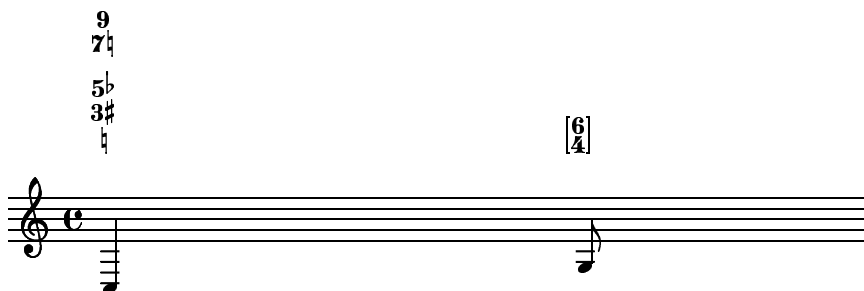
Ez-notation prints names in note heads. You also get ledger lines, of course.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/easy-notation.ly’:`



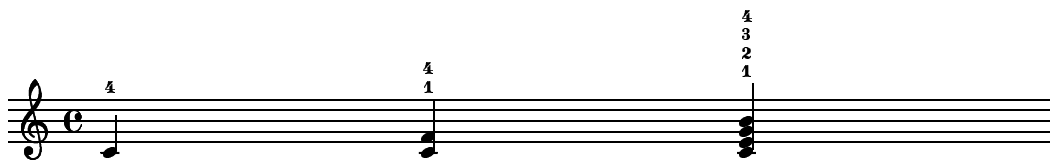
Test figured bass.

Figured bass is created by the FiguredBass context which eats figured bass requests and rest-requests. You must enter these using the special `\figures { }` mode, which allows you to type numbers, like `<4 6+>`.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/figured-bass.ly’:`



Automatic fingering tries to put fingering instructions next to noteheads.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/fingering.ly’:`



The magnification can be set for any font. Note that this doesn't change variable symbols such as beams or slurs. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/font-magnification`



As a last resort, the placement of grobs can be adjusted manually, by setting the `extra-`offset of a grob. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/generic-output-propert`



Simple glissando lines between notes are supported. The first two glissandi are not consecutive.

The engraver does no time-keeping, so it involves some trickery to get `< { s8 s8 s4 } { c4 \gliss d4 } >` working correctly.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/glissando.ly’:`



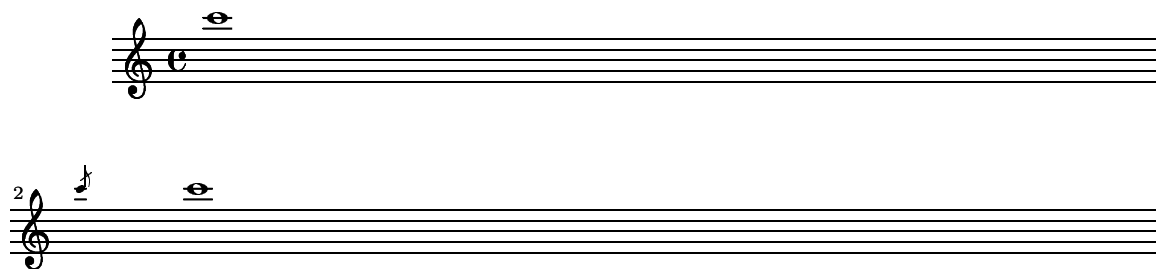
The autobeamer is not confusde by grace notes. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regres`



Bar line should come before the grace note. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression`



Grace notes do tricky things with timing. If a measure starts with a grace note, the measure does not start at 0, but earlier. Nevertheless, lily should not get confused. For example, line breaks should be possible at grace notes, and the bar number should be printed correctly. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-bar-number.ly’:`



Grace beams and normal beams may occur simultaneously. Unbeamed grace notes are not put into normal beams. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-beam.ly’:`



Dynamics on grace notes are small and behave nicely (don’t crash into to main note). `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-dynamic.ly’:`



grace code should not be confused by nested sequential musics, containing grace notes; practically speaking, this means that the end-bar and measure bar coincide in this example. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-nest.ly’:`



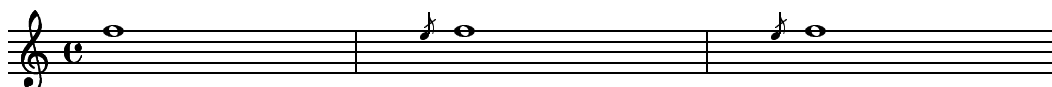
grace code should not be confused by nested sequential musics, containing grace notes; practically speaking, this means that the end-bar and measure bar coincide in this example. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-nest1.ly’:`



grace code should not be confused by nested sequential musics, containing grace notes; practically speaking, this means that the end-bar and measure bar coincide in this example.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-nest2.ly’:`



`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-nest3.ly’:`



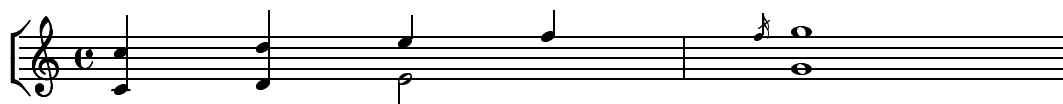
Another combination of grace note nesting. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/`



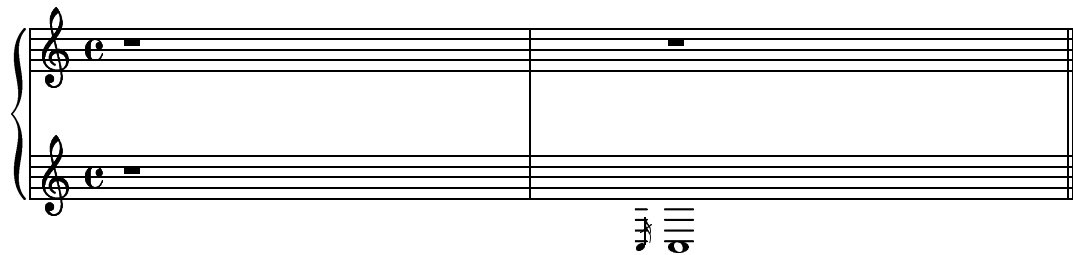
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-nest5.ly’:`



partcombiner and grace notes can go together `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/`



Stripped version of trip.ly. Staves should be of correct length. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/i`



Pieces may begin with grace notes. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-`



`startGraceMusic` should `no-stem-extend` to `true`; the two grace beams should be the same here. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-stems.ly’`:



grace notes in different voices/staves are synchronized. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/`



Grace notes and unfolded repeats. Line breaks may happen before grace notes. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-unfold-repeat.ly’`:



Repeated music can start with grace notes. Bar checks preceding the grace notes do not cause synchronization effects. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace-volta-`



Grace notes are typeset as an encapsulated piece of music. You can have beams, notes, chords, stems etc. within a `\grace` section. Slurs that start within a grace section, but aren't ended are attached to the next normal note. Grace notes have zero duration. If there are tuplets, the grace notes won't be under the brace. Grace notes can have accidentals, but they are (currently) spaced at a fixed distance. Grace notes (of course) come before the accidentals of the main note. Grace notes can also be positioned after the main note.

Grace notes without beams should have a slash, if `flagStyle` is not set. Main note scripts don't end up on the grace note.

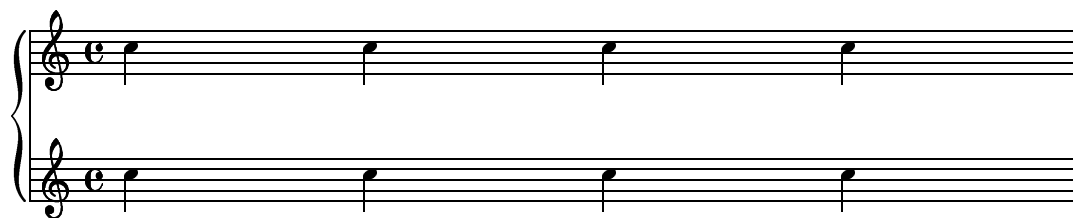
`'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/grace.ly':`



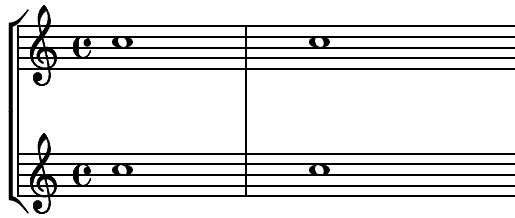
Hairpin dynamics start under notes if there are no text-dynamics. If there are text dynamics, the hairpin does not run into them. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/h`



Hara kiri should not upset fixed distance alignment like in pianostaff. In this example the middle staff is harakiried. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/hara-kiri-pianos`



Hara kiri staves kill themselves if they are empty. This example really contains two staves, but the second contains only spaces, and is therefore removed. Also, any staff brackets and braces are removed. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/hara-kiri-shor`



Key signatures can be set per pitch individually. This can be done bby setting `Staff.keySignature` directly. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/key-signatu`



Key signatures appear on key changes. They may also appear without barlines. The restoration accidentals are not printed at the start of the line. If `createKeyOnClefChange` is set, they’re also created on a clef change. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/keys.ly’:`



Lyric alignment

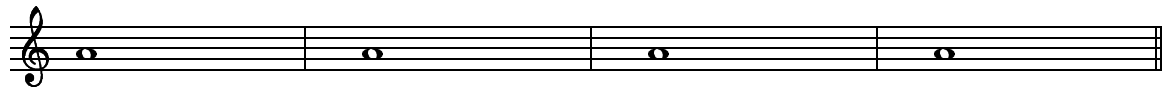
Lyric alignment is adjustable both interms of alignment between stanzas and on note-head.

If the property `alignment` is not set, there is automatic determination of alignment type based on punctuation. (ee `lyric-phrasing.ly`)

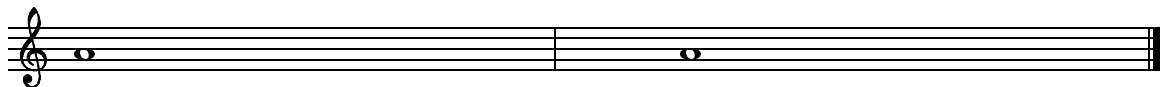
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/lyric-align.ly’:`



Particularly useful for reciting notes left centered right
with many syllables under them. l c r



left half way left one quarter left one tenth left one whole
l l l x



Very short lyrics remain in touch with their note
x

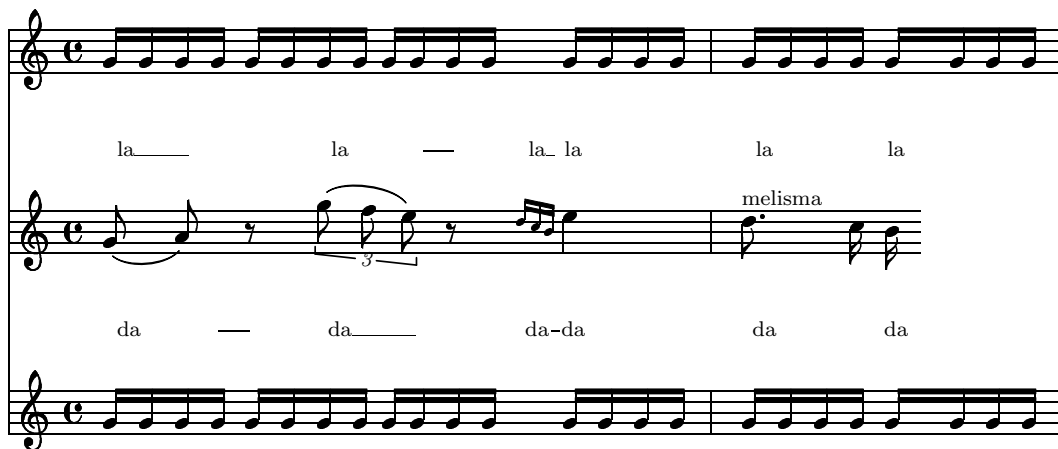
Unless ignore-length-mismatch is true
x

polyphonic rhythms and rests don't disturb \addlyrics. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input`



Do mi nus ex
Do na

Lyrics can be set to a melody automatically. Excess lyrics will be discarded. Lyrics will not be set over rests. You can have melismata either by setting a property `melismaBusy`, or by setting `automaticMelismas` (which will set melismas during slurs and ties). If you want a different order than first Music, then Lyrics, you must precook a chord of staves/lyrics and label those. Of course `\rhythm` ignores any other rhythms in the piece. Hyphens and extenders do not assume anything about lyric lengths, so they continue to work. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/lyric-combine.ly'`:



Lyric phrasing

We find start and end of phrases, and align lyrics of multiple stanzas accordingly.

Also, lyrics at start of melismata should be left aligned. (is that only lyrics that are followed by '___'? Because that seems to be the case now – jcn)

x	x	x	x	x

1:	Start	sentence	melisma	end.
2:	x	x	x_____	x

Only lyrics that are followed by '___' while there's a melisma, are left-aligned, in this case the third x. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/lyric-phrasing.ly'`:



1: Start sentence melisma end.
 2: x x x__ x.

Adding a `Bar_engraver` to the `LyricsVoice` context makes sure that lyrics don't collide with barlines. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/lyrics-bar.ly':`

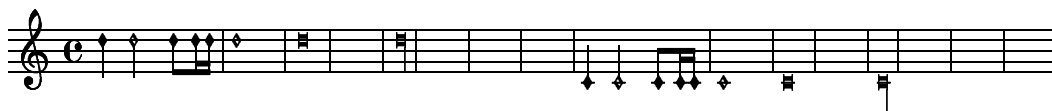
Extenders that end a staff should not extend past the staff. Also shown: a trick to get an extender at the end of the staff. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/lyrics-exte`

Lyrics syllables are aligned according to punctuation. `Stanza` and `stz` set stanza numbers. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/lyrics-multi-stanza.ly':`

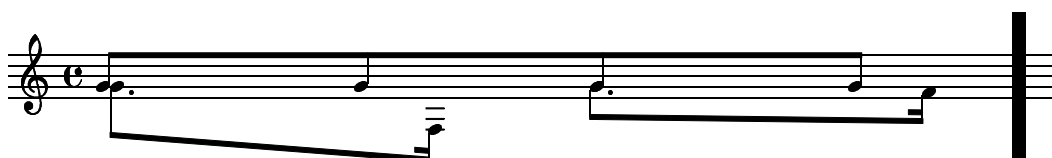
Stacking of markup scripts. `'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/markup-stac`



There is limited support for mensural notation: note head shapes are available. Mensural stems are centered on the note heads, both for up and down stems. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/mensural.ly’`:



If `NoteCollision` has `merge-differently-dotted` set, note heads that have differing dot counts may be merged anyway. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/merge-diffe`



If `Score.skipBars` is set, the signs for four, two, and one measure rest are combined to produce the graphical representation of rests for up to 10 bars. The number of bars will be written above the sign. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/mm-rests2.ly’`:

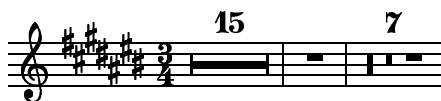


You can write molecule callbacks in Scheme, thus providing custom glyphs for notation elements. A simple example is adding parentheses to existing molecule callbacks.

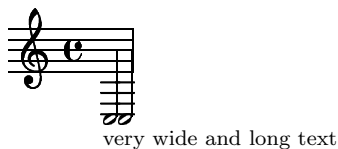
The parenthesized beam is less successful due to implementation of the Beam. The note head is also rather naive, since the extent of the parens are also not seen by accidentals. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/molecule-hacking.ly’`:



Multiple measure rests do not collide with barlines and clefs. They are not expanded when you set `Score.skipBars`. Although the multi-measure-rest is a `Spanner`, minimum distances are set to keep it colliding from barlines. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/multi-measure-rest.ly’`:



Text is set with empty horizontal dimensions. The boolean property `TextScript.no-spacing-rods` is used to control the horizontal size of text. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/non-empty-text.ly’:`



Note heads are flipped on the stem to prevent collisions. It also works for whole heads that have invisible stems. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/note-head-chord.ly’:`



Note head shapes are settable. The stem endings should be adjusted per note head. If you want different note head styles on one stem, you must create a special context called `Thread`.

Harmonic notes have a different shape and different dimensions. Nevertheless, noteheads in both styles can be combined, on either up or down stems. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/note-head-style.ly’:`



25



33



41



49



57



73



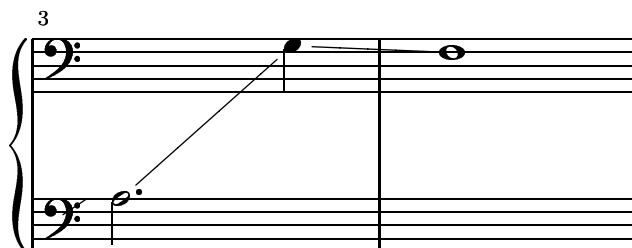
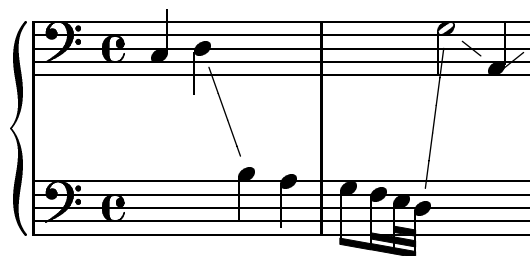
81



97



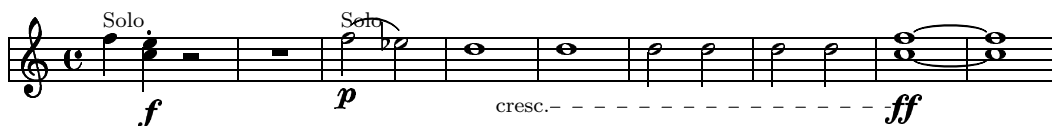
Note head lines (eg glissando) run between centers of the note heads.
'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/note-line.ly':



The number of stafflines of a staff can be set. Ledger lines both on note heads and rests are adjusted. Barlines also are adjusted. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/num`



Multi measure rests of second voice should not disappear. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/inpu`



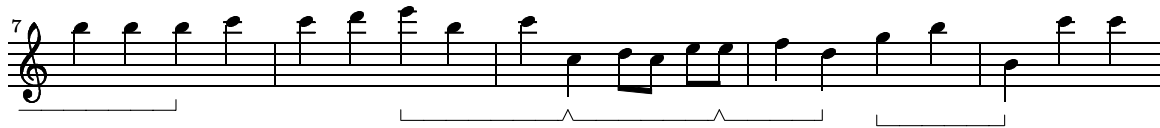
The partcombiner should not combine two small slurs into a big one. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/pc-switch-slur.ly’:`



Piano pedal symbols merge stop and start. The strings are configurable. Text style, bracket style, and a mixture of both are supported. Brackets work across line breaks.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/pedal.ly’:`





Measure and beat repeats are supported. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/`



Slurs play well with phrasing slur. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/phrasi`



The A is atop an invisible barline. The barline although invisible, is also translated be-
cause it is the last one of the break alignment. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression.`

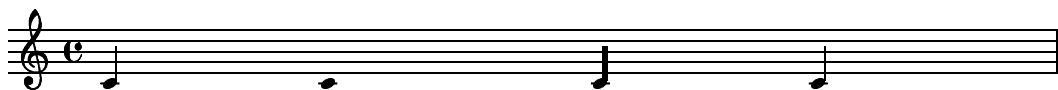


prefatory spacing

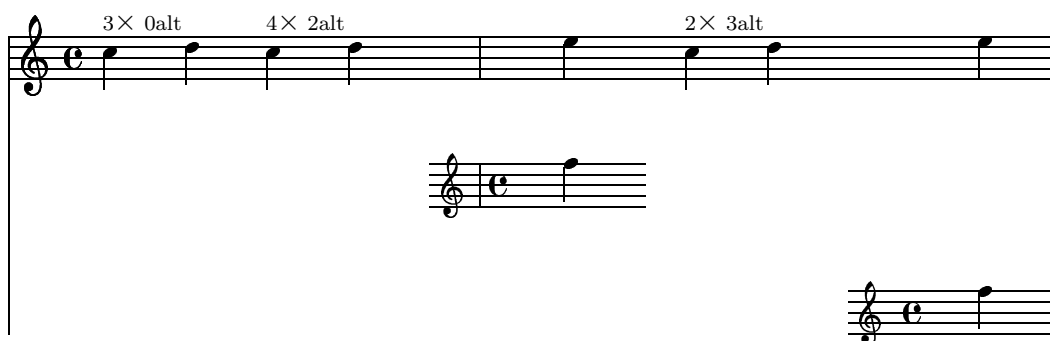
TODO: show all common combinations to check for spacing anomalies.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/prefatory-spacing-matter.ly’:`



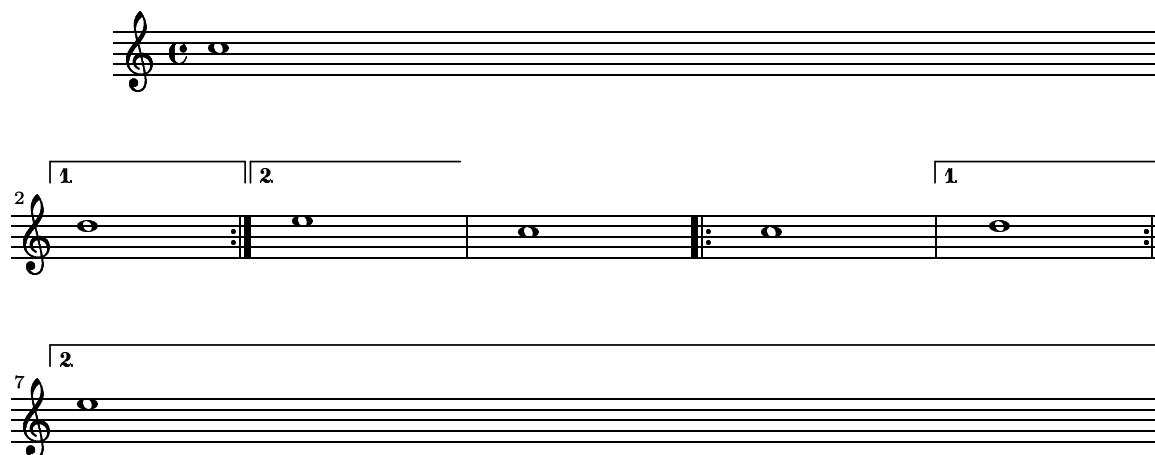
Once properties take effect during a single time step only. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/inpu`



Folded. This doesn't make sense without alternatives, but it works.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/repeat-fold.ly’:`



Across linebreaks, the left edge of a first and second alternative bracket should be equal
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/repeat-line-break.ly’:`



Repeats may be unfolded through the Scheme function `unfold-repeats`.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/repeat-unfold-all.ly’:`

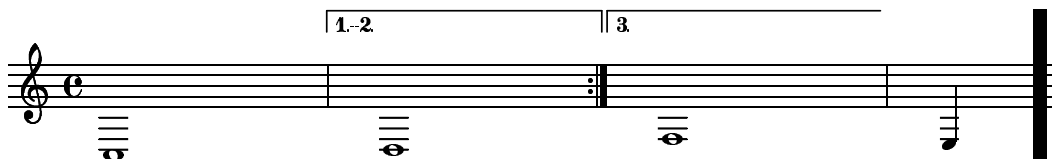


LilyPond has three modes for repeats: folded, unfolded and semi-unfolded. Unfolded repeats are fully written out. Semi unfolded repeats have the body written and all alternatives sequentially. Folded repeats have the body written and all alternatives simultaneously. If the number of alternatives is larger than the repeat count, the excess alternatives are ignored. If the number of alternatives is smaller, the first alternative is multiplied to get to the number of repeats.

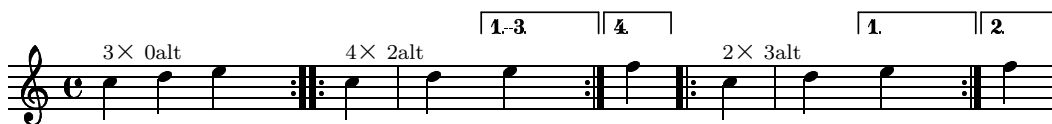
Unfolded behavior: `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/repeat-unfold.ly’:`



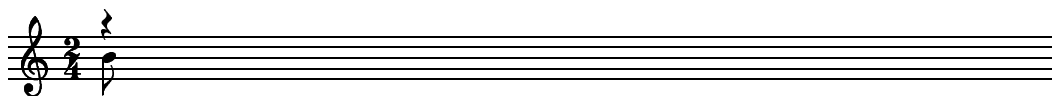
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/repeat-volta-skip-alternatives.ly’:`



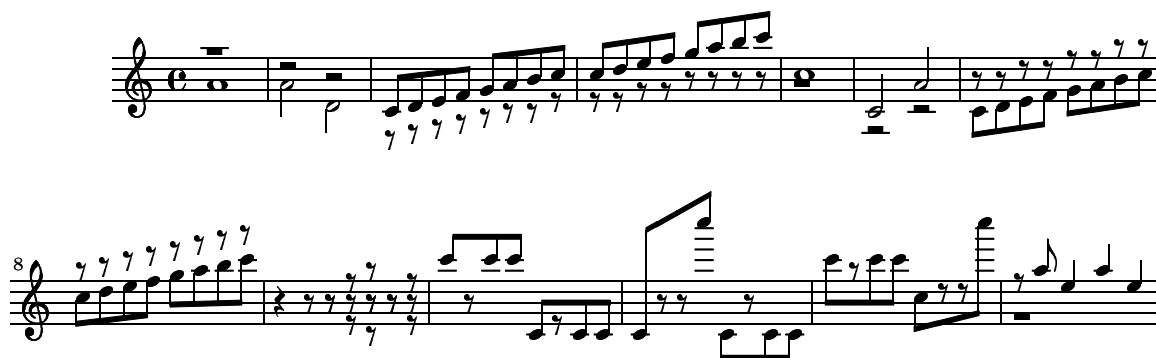
Volta (Semi folded) behavior. Voltas can start on non-barline moments. If they don't barlines should still be shown. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/repeat-volta`



Rests in collisions sit opposite of the note if no direction is specified for the voice containing the rest. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/rest-collision-default.ly’:`



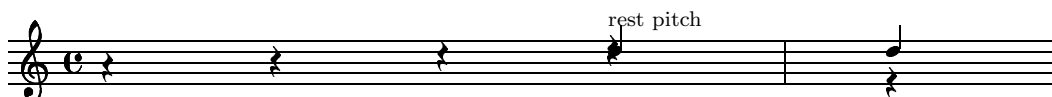
Rests should not collide with beams, stems and noteheads. Rests may be under beams. Rests should be move by integral number of spaces inside the staff, and by half spaces outside. Notice that the half and whole rests just outside the staff get ledger lines in different cases. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/rest-collision.ly’:`



whole and half rests moving outside the staff should get ledger lines
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/rest-ledger.ly’:



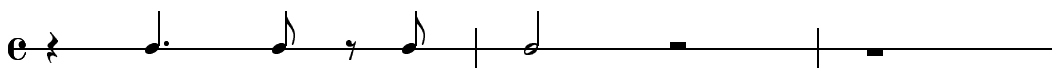
Rests can have pitches—these will be affected by transposition and relativization. If a rest has a pitch, rest collision will leave it alone.
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/rest-pitch.ly’:



Rests. Note that the dot of 8th, 16th and 32nd rests rest should be next to the top of the rest. All rests except the whole rest are centered on the middle staff line.
 ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/rest.ly’:



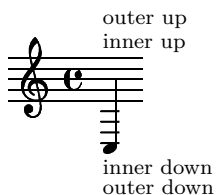
In rhythmic staves, stems should go up, and bar lines have the size for a 5 line staff. The whole note hangs from the rhythmic staff. ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/rh



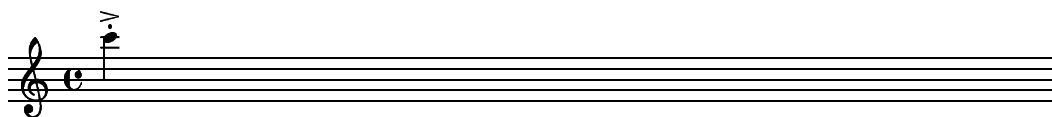
Scripts are put on the utmost head, so they are positioned correctly when there are collisions. ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/script-collision.ly’:



Scripts can be stacked. The order is determined by a priority field, but when objects have the same priority, the input order determines the order. Objects specified first are closest to the note. ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/script-stack-order.ly’:



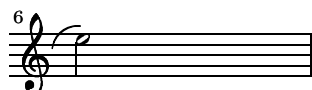
Scripts may be stacked. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/script-stacked`



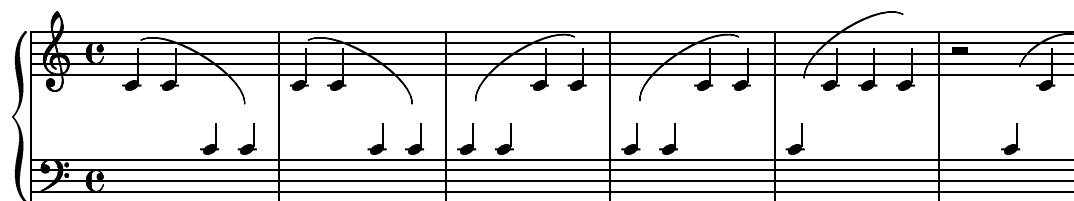
Slurs should be attached to note heads, except when they would collide with beams. Also see: opee-slurs. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-attachment.ly’:`



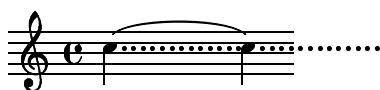
Across line breaks, slurs behave nicely. On the left, they extend to just after the preferatory matter, and on the right to the end of the staff. A slur should follow the same vertical direction it would have in unbroken state. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-broken-trend.ly’:`



The same goes for slurs. They behave decently when broken across linebreak. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-cross-staff.ly’:`



Slurs should not get confused by augmentation dots. We use a lot of dots here, to make problems more visible. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-dots.ly’:`



Slurs should look nice and symmetric. The curvature may increase only to avoid noteheads, and as little as possible. Slurs never run through noteheads or stems. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-nice.ly’:`



Slurs may be placed over rest. The slur will avoid colliding with the rest

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-rest.ly’:`



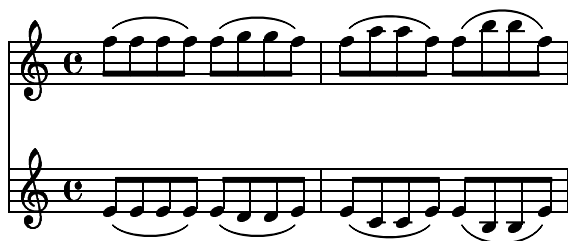
Manual hack for slur and staccato. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-s`



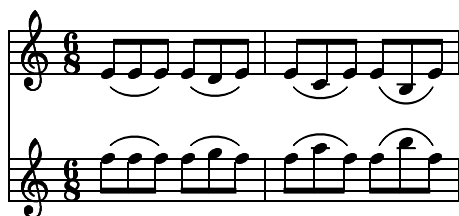
Trend of broken slur with user-overridden stem attachment should also follow the same vertical direction it would have had in unbroken state.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-stem-broken.ly’:`



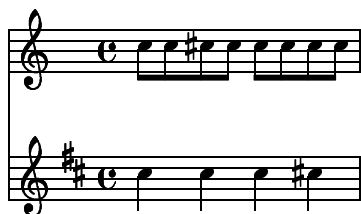
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-symmetry-1.ly’:`■



`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/slur-symmetry.ly’:`



Accidentals in different staves don't effect the spacing of the quarter notes here.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-accidental-staffs.ly’:`■



Accidentals sticking out to the left of a note will take a little more space, but only if the spacing is tight. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-accidental.ly’:`■



Downstem notes following a barline are printed with some extra space. This is an optical correction similar to juxtaposed stems.

Accidentals after the barline get some space as well. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/reg`



clef changes at the start of a line get much more space than clef changes halfway the line. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-clef-first-note.ly’:`■



A voicelet (a very short voice to get polyphonic chords correct) should not confuse the spacing engine. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-ended-voice.ly’:`■



A clef can be folded below notes in a different staff, if this doesn't disrupt the flow of the notes. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-folded-clef.ly’:`■



A clef can be folded below notes in a different staff, if there is space enough. With Paper_column molecule callbacks we can show where columns are in the score.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-folded-clef2.ly’:`



Spacing uses the duration of the notes, but disregards grace notes for this. In this example, the 8ths around the grace are spaced exactly as the other 8th notes.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-grace-duration.ly’:`



Grace note spacing. Should be tuned? `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/sp`



For knees, the spacing correction is such that the stems are put at regular distances. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-knee.ly’:`



Loose: `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-loose.ly’:`



Natural: `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-natural.ly’:`■



The flags of 8th notes take some space, but not too much: the space following a flag is less than the space following a beamed 8th head `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/s`



Rests get a little less space, since they are narrower. However, the feta quarter rest is relatively wide, causing this effect to be very small. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-rest.ly’:`



Notes that are shorter than the common shortest note, Get a space (i.e. without the space needed for the note) proportional to their duration. So 16th notes get 1/2 of the space of an eighth note. The total distance for a 16th is (including note head) is 3/4 of the eighth note. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-short-notes.ly’:`■



Upstem notes before a barline are printed with some extra space. This is an optical correction similar to juxtaposed stems. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing`



LilyPond corrects for optical spacing of stems. The overlap between to adjacent stems of different direction is used as a measure for how much to correct. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-stem-direction.ly’:`■



For juxtaposed chords with the same direction, a slight optical correction is used. It is constant, and only works if two chords have no common head-positions range.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-stem-same-direction.ly’:`■



Even if a line is very tightly spaced, there will still be room between prefatory matter and the following notes. The space after the prefatory is rigid. In contrast, the space before the barline must stretch like the space within the measure.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-tight.ly’:`



When tightly spaced, hinterfleisch -> 0. Stems may touch the bar lines, opposite stems may touch each other. We need a minimum of about a note-width/interline space in these situations, so that in tightly spaced music all vertical lines are about equally spaced.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/spacing-very-tight.ly’:`■



Span bars draw only in between staff bar lines, so setting those to transparent shows bar lines between systems only. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/span-bar.ly’:`■

bla die bla

bla die bla
foo bar foo

bla die bla
foo bar foo
foo bar foo

bla die bla
foo bar foo
foo bar foo

The staccato dot (and all scripts with follow-into-staff set), must not be on staff lines.
‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/staccato-pos.ly’:

Staff margins are also markings attached to barlines. They should be left of the staff, and be centered vertically wrt the staff. They may be on normal staves, but also on compound staves, like the PianoStaff
‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/staff-margin.ly’:

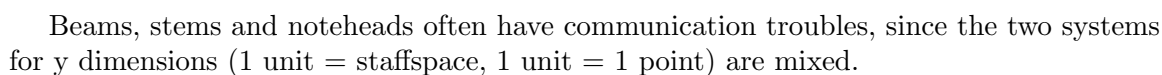
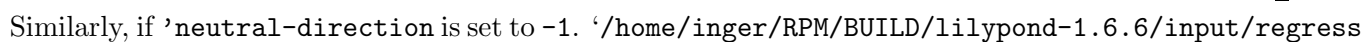
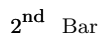
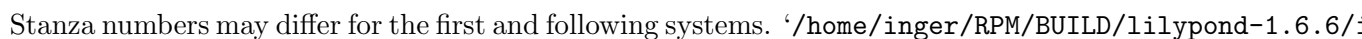
Right

Piano

Left

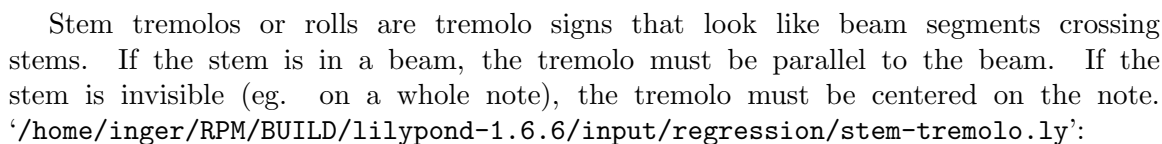
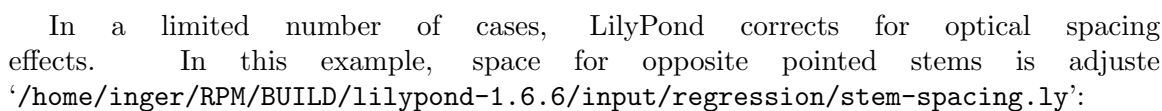
The staff is a grob, and may be adjusted as well: this one shows a staff with 6 thick line, and a slightly large staffspace. Beams remain correctly quantized.

‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/staff-tweak.ly’:



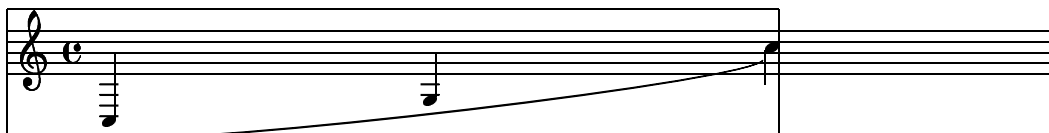
Stems, beams, ties and slurs should behave similarly, when placed on the middle staff line. Of course stem-direction is down for high notes, and up for low notes.

`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/stem-direction.ly’:`





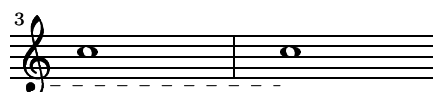
LilyPond correctly determines the size of every system. This includes postscript constructs such as slurs. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/system-extents.ly’:`



The piano brace should be shifted horizontally if it is enclosed in a bracket. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/system-start-bracket.ly’:`



`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/text-spanner.ly’:`



When tying notes with accidentals across a bar boundary, the accidental must not be drawn on the note in the next bar. Unless the tie crosses a line break, in which case the accidental is repeated if it would be different from an untied note. The next note of the same pitch in this next bar should always show the accidental (even if it’s natural). Slurring a accidentaled note to a natural one across bar boundaries should be explicit.

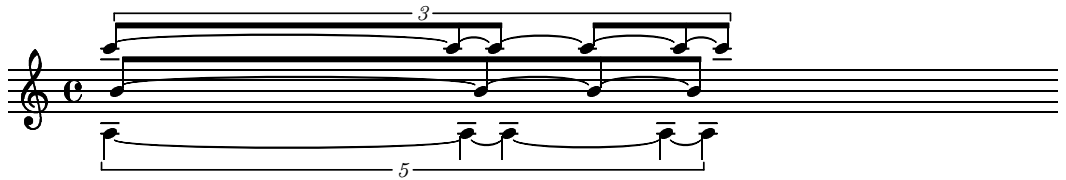
Pitches can be verified by printing them with the NoteNames context.
‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tie-accidental.ly’:

The image displays a musical score for a file named 'tie-accidental.ly'. It consists of four staves of music, each with a treble clef and a key signature of two sharps (F# and C#). The first staff contains 16 notes, with note names printed below: g', g', g', g', gis', gis', gis', gis', gis', gis', g', gis', gis', g', gis', g'. The second staff contains 16 notes, with note names printed below: gis', gis', gis', gis', gis', gis', g', gis', gis', gis', g', gis', g', gis', g'. The third staff contains 4 notes, with note names printed below: gis', gis', gis', g'. The fourth staff contains 2 notes, with note names printed below: g', g'.

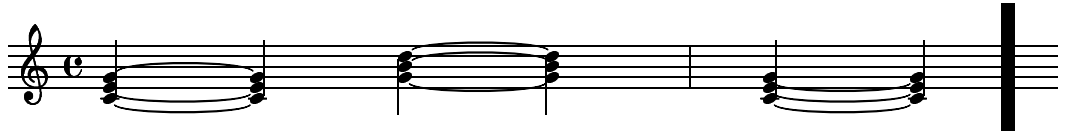
First and second bes (tied across line break) should get an accidental, but others should not. Only first B should get natural sign. ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tie-

The image displays three staves of musical notation. The first staff shows a single note head with a flat accidental. The second staff shows a sequence of notes with ties. The third staff shows a sequence of notes with ties.

Tie engraver uses **busyGrobs** to keep track of note heads. Test if this queue works by throwing many mixed tuplets at it. ‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tie-busy



When tying chords, the outer slurs point outwards, the inner slurs point away from the center of the staff. Override with `tieVerticalDirection`.
`‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tie-chord.ly’:`



Ties should not collide with dots. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tie-dot.ly’:`



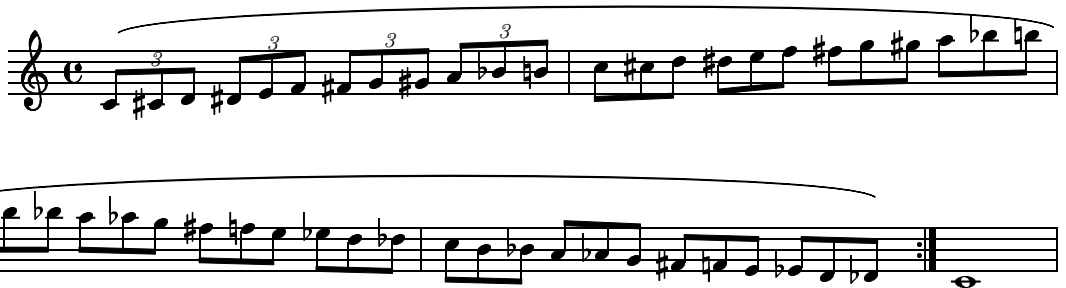
Tying a grace to the to a following grace or main note works. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tie-grace.ly’:`



Ties are strictly horizontal. They are placed in between note heads. The horizontal middle should not overlap with a staffline. `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tie-line.ly’:`



Simple beams. This broke somewhere < 1.3.110 `‘/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tie-beam.ly’:`



Tuplets are indicated by a bracket with a number. There should be no bracket if there is a beam exactly matching the length of the triplet. The bracket does not interfere with the stafflines, and the number is centered in the gap in the bracket.

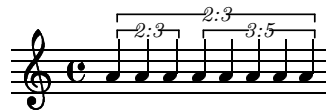
The bracket stops at the end of the stems, if the stems have the same direction as the
'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tup.ly':



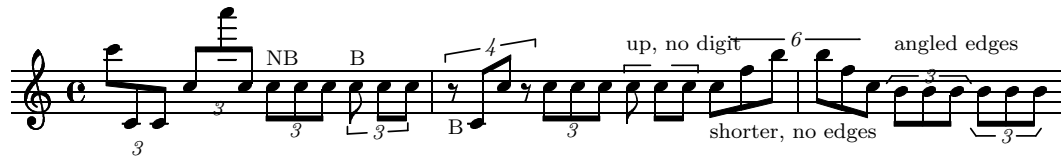
In combination with a beam, the bracket of the tuplet bracket is removed. This only happens if there is one beam, as long as the bracket.
'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tuplet-beam.ly':



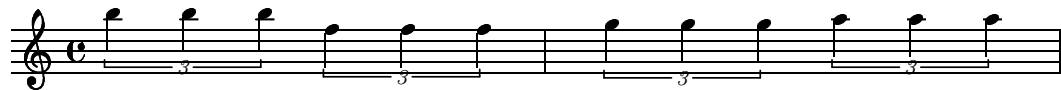
Manual hack for nested tuplets, move outer tuplet up. '/home/inger/RPM/BUILD/lilypond-1.6.6/input/r



Tuplet bracket formatting supports numerous options: NB should have no bracket, B should have bracket. '/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tuplet-properties.ly':



Horizontal tuplet brackets are shifted vertically to avoid staff line collisions.
'/home/inger/RPM/BUILD/lilypond-1.6.6/input/regression/tuplet-staffline-collision.ly':



Broken volta spanners behave correctly at left edge in all cases. '/home/inger/RPM/BUILD/lilypond-1.6.6/



