# LilyPond Contemporary Notation Cookbook: Snippets and Their Grammars

Yoshiaki Onishi
School of Music, University of Delaware
info@yoshionishi.com

Version: January 17, 2025

Cite: Onishi, Yoshiaki. "LilyPond Contemporary Notation Cookbook: Snippets and Their Grammars," (Version January 17, 2025), GitHub repository, https://github.com/yoshiakionishi/lilypond-snippets

# Contents

# Foreword

## 0.1 Preamble

This document houses all the codes I built on LilyPond since September 2024. Because I deal with contemporary notations in my compositional practice, I found myself creating codes and turning them into variables in order to repeatedly use them in my projects. I created a dedicated `.ly` file to store these codes for use, which quickly became very lengthy. I thought it would be useful to organize them into a document where I could easily consult and remind myself what they are and how to use them. This is that document.

Because I use LilyPond actively in my daily compositional and musical typesetting activities, this document is a work in progress.

## 0.2 README

This document and the codes contained herein are under the MIT License. So long as you include the copyright as well as the MIT License permission notices, please feel free to use my codes in your LilyPond files or modify them according to your specific need. Furthermore, crediting in the following manner is greatly appreciated:

```
% Original Code written by Yoshiaki Onishi
% https://github.com/yoshiakionishi/lilypond-snippets
```

I make this document public because I wish to return something useful to the LilyPond community, but also to seek and implement any improvements other users may find in my codes. Please feel free to reach out to the email address shown on the title page of this document.

In the interest of making the codes found in this document available to as many people as possible, I have avoided using copyrighted musical examples. However, wherever appropriate, I have provided bibliographical sources. Furthermore, I acknowledge that, just as academic work in humanities goes, my ideas are built on those that are formulated by

others; as such, whenever there is a direct source of inspiration for formulating a code, I provide sources.

In creating this document, I make no claim that my notational choices represent an absolute standard that everyone should adhere to. Once the basic principles of notation and typesetting are established (e.g., avoiding collisions, etc.), notation becomes a personal decision for each composer, shaped by careful study of preexisting scores and an evaluation of their musical contexts.

For example, in his book *The Bass Clarinet – A Personal History*, Harry Sparnaay lists nine variants of noteheads for the slap tongue technique.[1] In my work, I created two subcategories of the slap tongue technique: one followed by a pitch and another followed by an air sound (which produces the slap tongue effect that sounds "empty"). To distinguish between the two, I decided to use encircled noteheads—both filled and hollow—and attribute them to each subcategory. Again, this is a method that I have found works for my music, but I would be reluctant to suggest that others should follow the same.[2]

Readers are encouraged to modify my codes in order to suit their desired techniques. This document serves as a record of how I arrived at certain notational choices, because learning LilyPond meant that I would also need to become familiar with Scheme, which proved to be somewhat challenging—even though I have used Common Lisp before owing to programming in OpenMusic—because I had to make many guessworks as I navigated various Scheme codes in other snippets available online. I have also gained familiarity in PostScript language as I continued to familiarize myself with LilyPond.[3]

## 0.3 Background

After MakeMusic announced that they would cease development of the music notation software program Finale, which I had used for the past twenty-four years, I decided to explore a few other music notation programs to determine the best alternative. At the time of writing this document in late November 2024, a little under three months have passed since I started using LilyPond for my daily typesetting needs. I now open LilyPond more often than Finale and am committed to using it for the foreseeable future. LilyPond appears to me as the way forward both as a composer and a musical typesetter, as other proprietary notation programs, such as Dorico (which MakeMusic has claimed to be the leading program in the industry) and Sibelius, fall short of what I wish to accomplish.

While LilyPond is "just" a music notation software program that I happened to choose, it

1. Harry Sparnaay, *The Bass Clarinet: A Personal History* (Periferia Sheet Music, 2012), 66.

2. This particular notation becomes quickly problematic in terms of rhythmic notation when a bar is longer than a half note (e.g. 1/2, 2/4, 4/8...) For this reason, I tend to favor time signatures that avoids the use of a half note, such as 3/8 or 5/16.

3. See Appendix A for some resources I referred to for Scheme- and PostScript-related matters.

is, in a way, more than a toolkit for a composer. It appears that way to me, at least, because choosing an open-source platform with strong community support and engagement, rather than a proprietary program where desired functionality is subject to the priorities of a small group of salaried developers, reflects a critique of the capitalist/commercialist mindset that often pervades a composer's life.

For example, before transitioning to LilyPond, I briefly explored Dorico. However, as of late September 2024, its functionality for displaying straight flags was very limited; the angle of the straight flags provided by the software was too steep. I consulted the online forum and discovered that another user had posted a question similar to mine. The chief developer of Dorico responded to that post, noting that implementing improvements to this feature was possible but "not currently a high priority."[4] In this tiered structure typical of capitalism, composers may find themselves with increasingly limited creative "freedom."

MakeMusic has heavily advertised on social media platforms that Finale users should migrate to Dorico because it is the "next industry standard." However, this advertising seems to discourage thoughtful consideration of alternatives, leaving little room for reflection or exploration. I became increasingly disillusioned as I witnessed the coercion to invest in a program—however exciting it may appear—with no definite promise of its long-term security and stability.

Of course, it is not my intent to claim that all composers should abandon their proprietary programs of choice, particularly those they have invested money in and/or have been using for many years. It is, however, important to note that:

1. All proprietary programs are at the mercy of the executives who run the companies behind them. "Oh, [insert the name of a proprietary program] is operated by [insert the name of its company], and I just don't see them closing the program down," someone might say. Yet, it happened to Finale.

2. All notation programs, owing to the ways they operate, exert some degree of influence on the way composers compose. As early as the 1980s, Finale's *Mass Mover*, *Note Mover*, and MIDI playback features were already influential in shaping the way composers worked on their music.[5] On the one hand, these features may have helped composers save time. On the other hand, their ready availability may have invited overuse.

3. The lack or underdevelopment of certain functionalities may also push composers to work in certain ways rather than others. Finale benefitted from having the flexibility to implement graphical notation, but even then, many of my composer friends found it practical to use external graphical editing programs to further refine their scores.

4. See: https://forums.steinberg.net/t/straight-flags-angle/766503.
5. For example, watch from 15:20 of https://youtu.be/T1IRlg87Qks.

Even from my personal experience using Finale, I encountered situations where I had to devise creative alternatives to meet my notational goals.

These points implicitly highlight the benefits of learning an additional notation program, ideally an open-source one, alongside the program one primarily uses. LilyPond resonated with me most because of its text-based interface, which I have become increasingly familiar with through my involvement in computer programming. As other users have remarked, I have also found it to be very flexible and extensible. All the snippets I list in this document can be reused with relative ease, allowing me to save time in the long run when using specialized notations in my music. This was not necessarily the case when working on the music notation of extended techniques in Finale.

## 0.4 How This Document Is Structured

Each chapter of this document addresses a specific element of music notation, such as noteheads, stems, beams, and so on. Some chapters, however, cover topics specific to LilyPond coding, such as Markups and Spanners. Snippets that use more than one snippet covered in earlier chapters, thus simulating practical applications of these snippets, are covered in the chapter *Combinations*. Snippets that do not appear to belong to earlier chapters find their home in the chapter *Miscellanies*.

Each snippet entry includes a musical example, a description, the relevant grammar, the code required for the snippet to function, and, whenever necessary, a "Discussion" section.

## 0.5 LilyPond Version Used

The version of LilyPond used to create these snippets is `2.24.4`.

## 0.6 Acknowledgements

I thank the supportive community of LilyPond users, whose exchanges on `lilypond-user` mailing list have inspired me greatly.

Even though I have not met him, I am grateful to Ben Lemon for his generosity in creating and sharing his LilyPond tutorial videos on YouTube. These videos were immensely helpful during the initial stages of learning LilyPond.

I also want to thank my friends who inspired me to start using LilyPond. It was Cole Ingraham who first introduced me to the program in 2016. My initial attempt at using it was not successful, but more recently, Santiago Beis composed and typeset his orchestral piece *Spletna* entirely in LilyPond, which compelled me to give it another try.

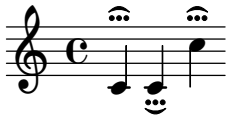
I extend my gratitude to my composition students at the University of Delaware School of Music, with whom I embarked on this journey of learning LilyPond. Even though they were not directly affected by Finale's discontinuation, they remained curious and enthusiastic about exploring this program. I hope that if the programs of their choice ever face a fate similar to Finale (though I sincerely hope they do not), they will be better equipped to adapt without the annoyance and arduous work often associated with transitioning to a new tool.

**Table of Contents**

# Chapter 1

# Articulations

## 1.1 Jeté (Ricochet)



### 1.1.1 Description

I use this notation to designate jeté/ricochet for string instruments, adding that the number of bounces are undetermined.[1]

I apply this indication *above* the note regardless of how high or low the note is; however, in case of need, I have supplied the version to be used *under* the note.

### 1.1.2 Grammar

```
NOTE \jete
NOTE \jeteUp
NOTE \jeteDown
```

1. Concerning the technique of adding articulation designs to an internal alist, I was inspired by the following thread on `lilypond-user` mailing list: https://lists.gnu.org/archive/html/lilypond-user/2015-04/msg00105.html

### 1.1.3 Code

```
\version "2.24.4"

jeteDesign =
\markup
\center-align
\combine \combine \combine
\override #'(filled . #t)
\path  #0.1
#'((moveto    -0.25 0.5)
   (curveto   0.35 1.1 0.85 1.1 1.45 0.5)
   (curveto   0.85 0.8 0.35 0.8 -0.25 0.5)
   (closepath))
\draw-circle #0.2 #0 ##t
\translate #'(0.6 . 0) \draw-circle #0.2 #0 ##t
\translate #'(1.2 . 0)\draw-circle #0.2 #0 ##t
#(append! default-script-alist
    (list
     `(jetelistUp
       . (
           (stencil . ,ly:text-interface::print)
           (text . ,#{ \markup \jeteDesign #})
           ; any other properties
           (toward-stem-shift-in-column . 1.0)
           (outside-staff-priority . #t)
           (padding . 0.5)
           (avoid-slur . around)
           (direction . ,UP))))

    (list
     `(jetelistDown
       . (
           (stencil . ,ly:text-interface::print)
           (text . ,#{ \markup \rotate #180 \jeteDesign #})
           ; any other properties
           (toward-stem-shift-in-column . 1.0)
           (outside-staff-priority . #t)
           (padding . 0.5)
           (avoid-slur . around)
           (direction . ,DOWN)))))
```

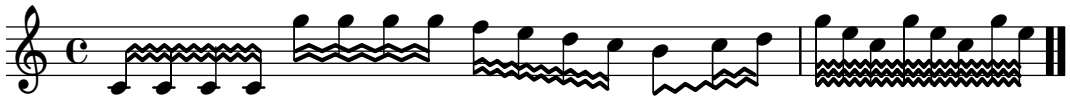
```
40
41  jete =  #(make-articulation 'jetelistUp)
42  jeteUp = #(make-articulation 'jetelistUp)
43  jeteDown = #(make-articulation 'jetelistDown)
44
45
46  {c'4\jete c'4 \jeteDown c''\jeteUp }
```

**Table of Contents**

# Chapter 2

# Beams

## 2.1 Wiggle Beam (zig-zag shaped beam)



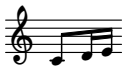
### 2.1.1 Description

Ordinary beams are replaced with zig-zag beams. A set of forward then backward beams are printed in the amount specified in the argument. I use this notation in such pieces as *jeux enjeux* (2022) for brass quintet, in order to designate somewhat uneven rhythmic figures, which are nonetheless to be played within the time frame indicated.

`\wiggleBeamOne` replaces an $8^{\text{th}}$-note beam.

`\wiggleBeamTwo` replaces a $16^{\text{th}}$-note beam.

`\wiggleBeamThree` replaces a $32^{\text{nd}}$-note beam.

`\wiggleBeam_markup` adds a zig-zag beam at will. This allows beaming of mixed note durations, such as: 

`\wiggleBeamStemAdjust` allows the adjustment of a stem length, in the event the wiggle beam and the stem do not touch each other.

### 2.1.2 Grammar

```
\wiggleBeamOne #vOffset #howMany #width #rotation
\wiggleBeamTwo #vOffset #howMany #width #rotation
\wiggleBeamThree #vOffset #howMany #width #rotation
```

**NB**

- `hOffset` = (`\wiggleBeam_markup` only) the horizontal offset value originating from where the ordinary beam is placed.
- `vOffset` = the vertical offset value originating from where the ordinary beam is placed.
- `howMany` = how many "wiggles" to print. It only accepts integers.
- `width` = how wide each "wiggle" should appear. When in doubt, start with `#1`.
- `rotation` = a positive value would rotate the beam upward, and the negative value would rotate the beam downward.

---

`NOTE \wiggleBeam_markup #hOffset #vOffset #howMany #width #rotation`

**NB**

- `hOffset` = the horizontal offset value originating from where the ordinary beam is placed.
- `vOffset` = the vertical offset value originating from where an above-staff markup is placed. Thus, `#0` would place a wiggle beam above the staff line.
- `howMany` = how many "wiggles" to print. It only accepts integers.
- `width` = how wide each "wiggle" should appear. When in doubt, start with `#1`.
- `rotation` = a positive value would rotate the beam upward, and the negative value would rotate the beam downward.
- More than one `\wiggleBeam_markup` may be added in sequence, provided that for each instance all the arguments are defined.

---

`\wiggleBeamStemAdjust #fromMiddleLine #howFar NOTE`

**NB**

- `fromMiddleLine` = (`\wiggleBeamStemAdjust` only) = determines one end of the stem, `#0` being the middle line of an ordinary 5-line staff.

- `howFar` = (`\wiggleBeamStemAdjust` only) = computes how long the stem should be extended. A positive value would draw the stem upward, and a negative value would draw the stem downward. An integer corresponds to the distance between two staff lines of an ordinary 5-line staff.

### 2.1.3 Code

```
wiggleBeamOne =
#(define-music-function (vOffset howMany howWide howTilted)
   (number? number? number? number?) #{
     \once \override Voice.Beam.stencil = #ly:text-interface::print
     \once \override Voice.Beam.text = \markup {
       \translate #(cons 0 vOffset)
       \postscript #(string-append
             "newpath
              1 setlinejoin
              1 setlinecap
              0.35 setlinewidth
              0.13 0 moveto "
             (number->string howMany)
             " {" (number->string (* 0.6 howWide))  " "
             (number->string (+ 0.5 howTilted)) " rlineto "
             (number->string (* 0.6 howWide))
             " -0.5 rlineto} repeat
              stroke"
                   )

     }
   #})

wiggleBeamTwo =
#(define-music-function (vOffset howMany howWide howTilted )
   (number? number? number? number?) #{
     \once \override Voice.Beam.stencil = #ly:text-interface::print
     \once \override Voice.Beam.text = \markup {
       \translate #(cons 0 vOffset)
       \postscript #(string-append
             "newpath
              1 setlinejoin
              1 setlinecap
              0.35 setlinewidth
```

```
                0.13 0 moveto "
               (number->string howMany)
               " {" (number->string (* 0.6 howWide)) " "
               (number->string (+ 0.5 howTilted)) " rlineto "
               (number->string (* 0.6 howWide))
               " -0.5 rlineto} repeat
                stroke newpath
                0.35 setlinewidth
                1 setlinejoin
                0.13 -0.75 moveto "
               (number->string howMany)
               " {" (number->string (* 0.6 howWide))  " "
               (number->string (+ 0.5 howTilted)) " rlineto "
               (number->string (* 0.6 howWide))
               " -0.5 rlineto} repeat
                stroke"
                       )
      }
    #})

wiggleBeamThree =
#(define-music-function (vOffset howMany howWide howTilted )
   (number? number? number? number?)
   #{
     \once \override Voice.Beam.stencil = #ly:text-interface::print
     \once \override Voice.Beam.text = \markup         {
       \translate #(cons 0 vOffset)
       \postscript #(string-append
             "newpath
              1 setlinejoin
              1 setlinecap
              0.35 setlinewidth
              0.13 0 moveto "
               (number->string howMany) " {"
               (number->string (* 0.6 howWide))  " "
               (number->string (+ 0.5 howTilted)) " rlineto "
               (number->string (* 0.6 howWide))
             " -0.5 rlineto} repeat
              stroke
              newpath
              0.35 setlinewidth
```

```
76                1 setlinejoin
77                0.13 -0.75 moveto "
78                 (number->string howMany) " {"
79                 (number->string (* 0.6 howWide))  " "
80                 (number->string (+ 0.5 howTilted)) " rlineto "
81                 (number->string (* 0.6 howWide))
82                 " -0.5 rlineto} repeat
83                stroke
84                newpath
85                0.35 setlinewidth
86                1 setlinejoin
87                0.13 -1.5 moveto "
88                 (number->string howMany) " {"
89                 (number->string (* 0.6 howWide)) " "
90                 (number->string (+ 0.5 howTilted)) " rlineto "
91                 (number->string (* 0.6 howWide))
92                 " -0.5 rlineto} repeat
93                stroke"
94                       )
95       }
96     #})
97
98  wiggleBeam_markup =
99  #(define-music-function (hOffset vOffset howMany howWide howTilted )
100    (number? number? number? number? number?)
101    #{
102      ^\markup         {
103        \translate #(cons hOffset vOffset)
104        \postscript #(string-append
105              "newpath
106               1 setlinejoin
107               1 setlinecap
108               0.35 setlinewidth
109               0.17 0 moveto "
110              (number->string howMany) " {"
111              (number->string (* 0.6 howWide))  " "
112              (number->string (+ 0.5 howTilted)) " rlineto "
113              (number->string (* 0.6 howWide))
114              " -0.5 rlineto} repeat
115               stroke"
116                      )
```

```
117
118       }
119     #})
120
121 wiggleBeamStemAdjust =
122 #(define-music-function (fromMiddleLine howFar)
123    (number? number?)
124    #{
125      \once \override Stem.stencil = #ly:text-interface::print
126      \once \override Stem.text = \markup {
127        \postscript #(string-append
128              "newpath
129               0.12 setlinewidth
130               0 " (number->string fromMiddleLine) " moveto
131               0 " (number->string howFar) " rlineto
132               stroke"
133                     )
134      }
135    #})
136
137 {
138   \wiggleBeamTwo #0 #9 #1.01 #0 c'16 c'
139   \wiggleBeamStemAdjust #-3 #3.4 c' c'
140   \wiggleBeamTwo #0 #5 #1.82 #0 g''
141   \wiggleBeamStemAdjust #2.5 #-3 g''
142   \wiggleBeamStemAdjust #2.5 #-3 g'' g''
143   \wiggleBeamTwo #-1 #9 #1.01 #-0.15 f''
144   \wiggleBeamStemAdjust #1.5  #-3.5 e''
145   \wiggleBeamStemAdjust #1 #-3.5 d''
146   \wiggleBeamStemAdjust #0.5 #-3.5 c''
147   \wiggleBeamOne #-3.5 #5 #1.4 #0.15 b'8
148   c''16 \wiggleBeam_markup #0 #-4.8 #2 #1.4 #0.15 d''
149   \wiggleBeamThree #-1.3 #19 #0.73 #0  g''32
150   \wiggleBeamStemAdjust #1.5  #-4 e''
151   \wiggleBeamStemAdjust #0.5  #-3 c'' g'' e''
152   \wiggleBeamStemAdjust #0.5  #-3  c''
153   \wiggleBeamStemAdjust #2.5  #-5 g'' e''
154   \bar ".."
155 }
```

### 2.1.4 Discussion

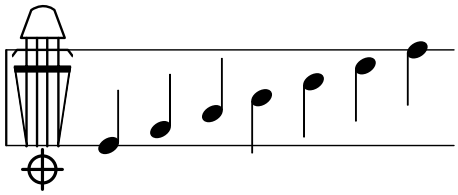
1. Admittedly, while the current setup allows great flexibility in making the wiggle beams appear, it is entirely possible that some of the parameters be automated.
2. When using many wiggle beams, it may be easier to make the score proportionally notated, in order to avoid the micromanagement of the parameters.

**Table of Contents**

# Chapter 3

# Clefs

## 3.1 String Position Clef



### 3.1.1 Description

String position clef to indicate bowing position. See Discussion for the associated command, `\normalClef`.

### 3.1.2 Grammar

```
\stringPositionClef
```

### 3.1.3 Code

```
stringPositionClefDesign = #(ly:make-stencil (list 'embedded-ps
"gsave
currentpoint translate
/fingboardpath
{
```

```
6  newpath
7
8  -0.55 7.5 moveto
9  0 -3 rlineto
10 1 -6.5 rlineto
11 -1 -1 rlineto
12 0 -3 rlineto
13 4.1 0 rlineto
14 0 3 rlineto
15 -1 1 rlineto
16 1 6.5 rlineto
17 0 3 rlineto
18 closepath
19
20 } def
21
22 fingboardpath clip
23 newpath
24 0.15 setlinewidth
25 0.5 4.75 moveto
26 0 -6.8 rlineto
27 -0.75 5 rlineto
28 3.5 0 rlineto
29 -0.75 -5 rlineto
30 0. 6.8 rlineto
31 stroke
32 0.35 setlinewidth
33 -0.4 2.75 moveto
34 3.75 0 rlineto
35 stroke
36
37 %inner two line
38 newpath
39 0.15 setlinewidth
40 1.16 4.75 moveto
41 0. -6.8 rlineto
42 1.8 4.75 moveto
43 0. -6.8 rlineto
44 stroke
45
46 %bridge
```

```
47 newpath
48 -0.4 3.6 moveto
49 0.3 0.4 rlineto
50 3.2 0 rlineto
51 0.3 -0.4 rlineto
52 stroke
53
54 %tailpiece
55 0.15 4.75 moveto
56 1 setlinecap
57 1 setlinejoin
58 2.75 0 rlineto
59 -0.65 1.75 rlineto
60 -0 -0  -0.6 0.55 -1.45 0  rcurveto
61 closepath
62 stroke
63
64 %mutesign
65 newpath
66 0.2 setlinewidth
67 1 setlinecap
68 1.5 -2.25 moveto
69 0 -2.5 rlineto
70 0.25 -3.5 moveto
71 2.5 0 rlineto
72 stroke
73 newpath
74 1.5 -3.5 0.85 0 360 arc
75 stroke
76 grestore")
77         (cons 0 3)
78         (cons 0 1))
79
80 stringPositionClefSize =
81 #(lambda (grob)
82    (let* ((sPCS (ly:grob-property grob 'font-size 0.0))
83           (mult (magstep sPCS)))
84      (ly:stencil-scale
85       stringPositionClef
86       mult mult)))
87
```

```
stringPositionClef = {
  \override Staff.Clef.stencil = \stringPositionClefDesign
}

normalClef = {
  \revert Staff.Clef.stencil
}

{
  \override Staff.StaffSymbol.line-positions = #'(6  -6)
  \override Staff.LedgerLineSpanner.stencil = ##f
  \override Staff.TimeSignature.stencil = ##f
  \override Staff.BarLine.stencil = ##f
  \stringPositionClef c'4 e' g' b' d'' f'' a''
}
```

### 3.1.4 Discussion

1. With the current design, `c'` would place a note at the lower end of the fingerboard. `a''` would place a note on the same line as the bridge.

2. The current design comes with the mute sign. If the mute sign is not needed, remove the following portion of the code above:

```
%mutesign
newpath
0.2 setlinewidth
1 setlinecap
1.5 -2.25 moveto
0 -2.5 rlineto
0.25 -3.5 moveto
2.5 0 rlineto
stroke
newpath
1.5 -3.5 0.85 0 360 arc
stroke
```

3. Once `\stringPositionClef` is used, in order to revert back to the normal clef, `\normalClef` must be used.

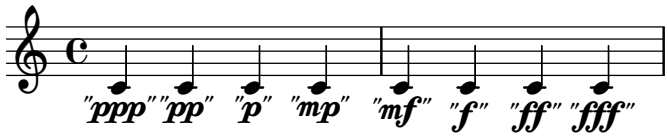
4. See Prescriptive Notation for String Instruments for a possible use of this clef.

**Table of Contents**

# Chapter 4

# Dynamics

## 4.1 Dynamics in Quotation Marks




### 4.1.1 Description

Dynamics in quotation marks, also known as *effort dynamics*, indicate those with which certain techniques must be carried on, understanding that the perceived dynamics will be quieter than what are indicated. Examples abound in scores by Helmut Lachenmann and others for such techniques as air sound, bowing directly on the bridge, etc..

### 4.1.2 Grammar

```
NOTE \qppp
NOTE \qpp
NOTE \qp
NOTE \qmp
NOTE \qmf
NOTE \qf
NOTE \qff
NOTE \qfff
```

### 4.1.3 Code

```
\version "2.24.4"

qmp = #(make-dynamic-script
        (markup #:combine
                #:combine
                #:translate '(-0.85 . -0.1)
                #:normal-text (#:italic #:fontsize 0.75 "\"")
                #:dynamic "mp"
                #:translate '(3.25 . -0.1)
                #:normal-text (#:italic #:fontsize 0.75 "\"")))
qp = #(make-dynamic-script
       (markup #:combine
               #:combine
               #:translate '(-0.95 . -0.1)
               #:normal-text (#:italic #:fontsize 0.75 "\"")
               #:dynamic "p"
               #:translate '(1.35 . -0.1)
               #:normal-text (#:italic #:fontsize 0.75 "\"")))
qpp = #(make-dynamic-script
        (markup #:combine
                #:combine
                #:translate '(-0.95 . -0.1)
                #:normal-text (#:italic #:fontsize 0.75 "\"")
                #:dynamic "pp"
                #:translate '(2.75 . -0.1)
                #:normal-text (#:italic #:fontsize 0.75 "\"")))
qppp = #(make-dynamic-script
         (markup #:combine
                 #:combine
                 #:translate '(-0.95 . -0.1)
                 #:normal-text (#:italic #:fontsize 0.75 "\"")
                 #:dynamic "ppp"
                 #:translate '(4.25 . -0.1)
                 #:normal-text (#:italic #:fontsize 0.75 "\"")))

qmf = #(make-dynamic-script
        (markup #:combine
                #:combine
                #:translate '(-0.85 . 0)
```

```
                #:normal-text (#:italic #:fontsize 0.75 "\"")
                #:dynamic "mf"
                #:translate '(3.25 . 0)
                #:normal-text (#:italic #:fontsize 0.75 "\""))))
qf = #(make-dynamic-script
       (markup #:combine
               #:combine
               #:translate '(-0.75 . 0)
               #:normal-text (#:italic #:fontsize 0.75 "\"")
               #:dynamic "f"
               #:translate '(1.65 . 0)
               #:normal-text (#:italic #:fontsize 0.75 "\"")))
qff = #(make-dynamic-script
        (markup #:combine
                #:combine
                #:translate '(-0.75 . 0)
                #:normal-text (#:italic #:fontsize 0.75 "\"")
                #:dynamic "ff"
                #:translate '(2.75 . 0)
                #:normal-text (#:italic #:fontsize 0.75 "\"")))
qfff = #(make-dynamic-script
         (markup #:combine
                 #:combine
                 #:translate '(-0.75 . 0)
                 #:normal-text (#:italic #:fontsize 0.75 "\"")
                 #:dynamic "fff"
                 #:translate '(3.85 . 0)
                 #:normal-text (#:italic #:fontsize 0.75 "\"")))

{

  c'4\qppp
  c'4\qpp
  c'4\qp
  c'4\qmp

  c'4\qmf
  c'4\qf
  c'4\qff
  c'4\qfff

```

```
81  }
82
83  \layout {
84    \context {
85      \Score    proportionalNotationDuration = #(ly:make-moment 1/9)
86    }
87  }
```

### 4.1.4 Discussion

In scores by Lachenmann, in concordance with German quotation marks (*Anführungszeichen*), the opening quotation mark points left, and placed on the bottom line, and the closing quotation mark points right and sits at the top of the last character. It would be possible to achieve this by adjusting the parameters in the Scheme code.[1]

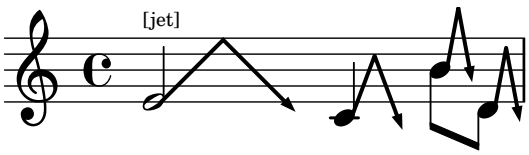
**Table of Contents**

1. See: https://lilypond.org/doc/v2.24/Documentation/extending/markup-construction-in-scheme

# Chapter 5

# Noteheads

## 5.1 Jet Whistle (for flute)




### 5.1.1 Description

Implementation of the jet whistle, as described in *The Techniques of Flute Playing* by Carin Levine and Christina Mitropoulos-Bott.[1]

### 5.1.2 Grammar

```
\jet NOTE #X-length
```

### 5.1.3 Code

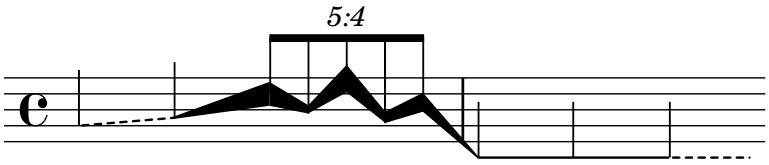
```
jet = #(define-music-function (pitchthing width) (ly:music? number?)
          (define p1 (ly:music-property pitchthing 'pitch))
          (define steps (+ -6 (ly:pitch-steps p1)))
          (define radToDeg (* 180 (/ 1 3.141592653589793)))
          #{ #pitchthing ^\markup {
            \postscript
```

1. Carin Levine and Christina Mitropoulos-Bott, *The techniques of flute playing = Die Spieltechnik der Flöte* (Kassel ; New York: Bärenreiter, 2003), 18.

```
            #(string-append "gsave newpath 0.2 setlinewidth 1.15 "
                            (number->string
                                    (+ -2.5 (* 0.5 steps))) " moveto "
                            (number->string
                                    (* 0.5 width)) " 4 rlineto "
                            (number->string
                                    (* 0.5 width)) " -4 rlineto
                    stroke
                    newpath
                    0.1 setlinewidth "
                            (number->string (+ 1.15 width)) " "
                            (number->string (+ -2.55 (* 0.5 steps)))
                            " moveto "
                            (number->string
                             (* radToDeg (atan (/ (* width 0.5) 4))))
                            " rotate
                    0 -1 rlineto
                    -0.35 1 rlineto
                    0.7 0 rlineto
                    -0.35 -1 rlineto
                    closepath
                    fill
                    grestore
                    ")
          } #})

\score {
  {
    \jet e'2^\markup {\fontsize #-5 {[jet]}} #8
    \jet c'4 #3
    \stemDown  \jet b'8 #1.5
    \jet d'8 #1.5
  }

  \layout {
    \context {
      \Score proportionalNotationDuration = #(ly:make-moment 1/10)
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}
```

**Table of Contents**

## 5.2 Line as a Notehead




### 5.2.1 Description

These functions replace an ordinary notehead with a dashed or a continuous line. For the continuous line, it is possible to adjust the beginning and ending thicknesses.

### 5.2.2 Grammar

```
\dashedLineNotehead NOTE1 PITCH #x-dist
\modularLineNotehead NOTE1 PITCH #beginningThick #endingThick #x-dist
```

**NB**

1. `NOTE1` specifies with which note the line starts. If necessary, the duration must be set, as well.
2. `PITCH` specifies with which pitch the line ends. Enter only the pitch; this information is used to determine the angle of the line, and it has no effect in displaying the rhythm.
3. `x-dist` specifies how long the line is.
4. `beginningThick` (for `modularLineNotehead` only) specifies how thick the beginning part of the line should be. `#15` gives a thin line, similar to the `\dashedLineNotehead` line. `#100` is as thick as a space between two neighboring lines of a staff.
5. `endingThick` (for `modularLineNotehead` only) specifies how thick the ending part of the line should be. `#15` gives a thin line, similar to the `\dashedLineNotehead` line. `#100` is as thick as a space between two neighboring lines of a staff.

### 5.2.3 Code

```
1
2 % See the entry on "Noteheadless" for its code;
3 % it is required for the snippet to run properly.
4
5 dashedLineNotehead =
6 #(define-music-function
7   (beginning end x-distance) (ly:music? ly:music? number?)
```

```
  (let*
   (
     (p1 (ly:music-property beginning 'pitch))
     (p2 (ly:music-property end 'pitch))
     (steps
      (-
       (+ (* 7 (ly:pitch-octave p2)) (ly:pitch-notename p2))
       (+ (* 7 (ly:pitch-octave p1)) (ly:pitch-notename p1))
       )
      )
     )
   #{
     {

       \once \override Voice.NoteHead.stencil = #ly:text-interface::print
       \once \override Voice.NoteHead.stem-attachment = #'(0 . 0)
       \once \override Staff.LedgerLineSpanner.stencil = ##f
       \once \override Voice.NoteHead.text = \markup         {
         % \translate #(cons 0 0)
         \postscript
         #(string-append
           "newpath 1 setlinecap
              0.15 setlinewidth
              0 0 moveto
              [.4 .4 .4 .4] 3 setdash "
           (number->string x-distance)  " " (number->string (* steps 0.5))
           " rlineto stroke"
           )
       }
       #beginning
       \revert Voice.NoteHead.stencil
       \revert Staff.LedgerLineSpanner.stencil
     }
   #})
  )


modularLineNotehead =
#(define-music-function
  (beginning end beginningThickness endingThickness x-distance)
  (ly:music? ly:music? number? number? number?)
```

```
49    (let*
50     (
51       (p1 (ly:music-property beginning 'pitch))
52       (p2 (ly:music-property end 'pitch))
53       (steps
54        (-
55         (+ (* 7 (ly:pitch-octave p2)) (ly:pitch-notename p2))
56         (+ (* 7 (ly:pitch-octave p1)) (ly:pitch-notename p1))
57         )
58        )
59       )
60     #{
61       {
62
63         \once \override Voice.NoteHead.stencil = #ly:text-interface::print
64         \once \override Voice.NoteHead.stem-attachment = #'(0 . 0)
65         \once \override Voice.LedgerLineSpanner.transparent = ##t
66         \once \override Voice.NoteHead.text = \markup         {
67           \postscript
68           #(string-append
69             "newpath 1 setlinecap 0.1 setlinewidth -0.05 0 moveto 0 "
70             (number->string (* beginningThickness 0.005)) " rlineto "
71             (number->string x-distance) " "
72             (number->string (+ (- (* endingThickness 0.005)
73                                   (* beginningThickness 0.005) )
74                                (* steps 0.5)))
75             " rlineto 0 "
76             (number->string  (* endingThickness -0.01))  " rlineto "
77             (number->string (* -1 x-distance))  " "
78             (number->string (- (* endingThickness 0.005)
79                                (* beginningThickness 0.005)
80                                (* steps 0.5)))
81             " rlineto
82                closepath
83                fill"
84             )
85         }
86         #beginning
87         \revert Voice.NoteHead.stencil
88         \revert Staff.LedgerLineSpanner.stencil
89       }
```

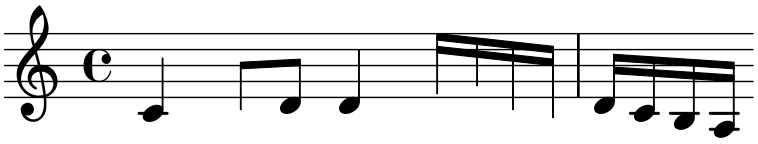
```
    #})
  )


\score {
  {
    \omit Staff.Clef
    \dashedLineNotehead g'4 a' #6
    \modularLineNotehead a' d'' #15 #150 #6
    \override TupletNumber.text = #tuplet-number::calc-fraction-text

    \stemUp  \tuplet 5/4 {
      \modularLineNotehead d''8 b' #150 #50 #2.5
      \modularLineNotehead b' f'' #50 #175 #2.5
      \modularLineNotehead f'' a' #175 #70 #2.5
      \modularLineNotehead a' c'' #70 #120 #2.5
      \modularLineNotehead c'' c' #120 #15 #3.5
    }
    |
    \modularLineNotehead c'4 c' #15 #15 #12
    \noteheadless c'
    \dashedLineNotehead c' c' #5
  }

  \layout {
    \context {
      \Score proportionalNotationDuration = #(ly:make-moment 1/10)
      \override SpacingSpanner.uniform-stretching = ##t
    }
  }
}


```

### 5.2.4 Discussion

See Prescriptive Notation for String Instruments for a possible use of this notehead.

**Table of Contents**

## 5.3 Noteheadless



### 5.3.1 Description

This snippet is hardly my own idea, as I largely quoted this technique from one of the snippets available on LSR.[2] However, I list it here because:

1. it took a while for me to find the workaround for maintaining the musical spacing as a result of omitting noteheads. It is worth noting that because merely disabling `NoteHead.stencil` will render the spacing to be squished, the approach of specifying `##t` for `NoteHead.transparent` (which itself will *not* eliminate the ledger lines) then `##t` for `NoteHead.no-ledgers` is effective in maintaining the general spacing.
2. I use this in conjunction with other notehead alterations, e.g. Line as a notehead.

### 5.3.2 Grammar

```
\noteheadless NOTE
\noteheadlessOn NOTE
\noteheadlessOff
```

**NB**

1. `\noteheadless` affects only one note immediately following.
2. For a group of notes, use `\noteheadlessOn` to toggle on the function. `\noteheadlessOff` will toggle off the function.

### 5.3.3 Code

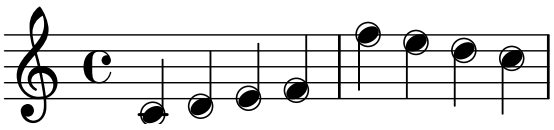
```
1
2  %% Inspired by:
3  %% http://lsr.di.unimi.it/LSR/Item?id=796
4
5
6  noteheadless = {
7    \once \override Voice.NoteHead.transparent = ##t
8    \once \override Voice.NoteHead.no-ledgers = ##t
```

2. See: http://lsr.di.unimi.it/LSR/Item?id=796

```
}

noteheadlessOn = {
  \override Voice.NoteHead.transparent = ##t
  \override Voice.NoteHead.no-ledgers = ##t
}
noteheadlessOff = {
  \revert Voice.NoteHead.transparent
  \revert Voice.NoteHead.no-ledgers
}


{
  c'4 \noteheadless c'8 d' d'4
  \noteheadlessOn e'16 f' c' b |
  \noteheadlessOff d' c' b a
}

```

Table of Contents

## 5.4 Slap Tongue, Type A



### 5.4.1 Description

In my music, I use encircled noteheads to denote slap tongues. Type A, encircled filled notehead, is used for a slap tongue with a regular note immediately following.

### 5.4.2 Grammar

`\slapA NOTE`

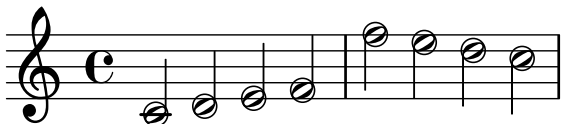
**NB** It only affects one note, owing to the `\once \override` functions within the code.

### 5.4.3 Code

```
slapA = #(define-music-function (note)   (ly:music?)
            #{ \once \override Voice.NoteHead.stencil =
               #ly:text-interface::print
               \once \override Voice.NoteHead.text =
               \markup {
                 \concat {
                   \musicglyph "noteheads.s2"
                   \postscript "newpath
                   -0.675 0.025 0.75 0 360 arc
                   closepath stroke"
                 }
               }
               $note #})

{
  \slapA c'4 \slapA d' \slapA e' \slapA f'
  \slapA f'' \slapA e'' \slapA d'' \slapA c''
}

```

Table of Contents

## 5.5 Slap Tongue, Type B



### 5.5.1 Description

In my music, I use encircled noteheads to denote slap tongues. Type B, encircled hollow notehead, is used for a slap tongue with an air sound immediately following.

### 5.5.2 Grammar

`\SlapB NOTE`

**NB** It only affects one note, owing to the `\once \override` functions within the code.

### 5.5.3 Code

```
slapB = #(define-music-function (note)   (ly:music?)
            #{ \once \override Voice.NoteHead.stencil =
               #ly:text-interface::print
               \once \override Voice.NoteHead.text =
               \markup {
                 \concat {
                   \musicglyph "noteheads.s1"
                   \postscript "newpath
                   -0.675 0.025 0.75 0 360 arc
                   closepath stroke"
                 }
               }
               $note #})
{
  \SlapB c'4 \SlapB d' \SlapB e' \SlapB f'
  \SlapB f'' \SlapB e'' \SlapB d'' \SlapB c''
}

```

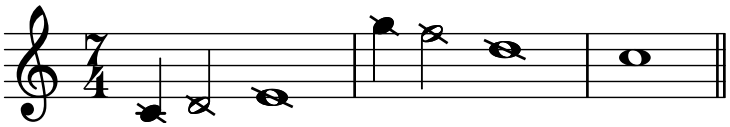
### 5.5.4 Discussion

As the musical example shows, when the Type B Slap Tongue notehead is applied to a quarter note, it could invite confusion in terms of rhythm. As a slap tongue itself is a short

sound, I only use the slap tongue noteheads on eighth notes or shorter note durations.

**Table of Contents**

## 5.6 Slashed Notehead



### 5.6.1 Description

Noteheads with backslashes applied.[3] I use this notehead to indicate, for example, notes on the piano whose strings are prepared, thus producing pitch/sound different from what is expected normally.

### 5.6.2 Grammar

```
\slashNote NOTE
\slashNoteOn NOTE
\slashNoteOff
```

**NB** `\slashNote` only affects one note, owing to the `\once \override` functions within the code. For a group of notes to have slashes applied, use `\slashNoteOn`. `\slashNoteOff` cancels the application.

### 5.6.3 Code

```
1
2  % Inspired by the code provided by Jean Abou Samra
3  % https://lists.gnu.org/archive/html/lilypond-user/2022-11/msg00333.html
4
5  slashNote =
6  \once \override Voice.NoteHead.stencil =
7  #(grob-transformer
8    'stencil
9    (lambda (grob original)
10     (let* ((added-markup
11             #{
12               \markup \general-align #Y #CENTER
13               #(case (ly:grob-property grob 'duration-log)
14                  ((0) #{ \markup \concat {
15                    \musicglyph "noteheads.s0"
16                    \postscript
```

3. The code provided by Jean Abou Samra in the following discussion thread on lilypond-user was very helpful in creating this code: https://lists.gnu.org/archive/html/lilypond-user/2022-11/msg00333.html

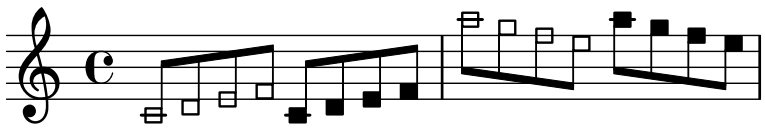
```
                    "gsave
                     0.17 setlinewidth
                     -2.3 0.6 moveto
                     0.3 -0.6 lineto
                     stroke
                     grestore"
                       } #})

                  ((1) #{ \markup \concat {
                    \musicglyph "noteheads.s1"
                    \postscript
                    "gsave
                     0.17 setlinewidth
                     -1.5 0.6 moveto
                     0.3 -0.6 lineto
                     stroke
                     grestore"
                       } #})

                  ((2) #{ \markup \concat {
                    \musicglyph "noteheads.s2"
                    \postscript
                    "gsave
                     0.17 setlinewidth
                     -1.5 0.6 moveto
                     0.3 -0.6 lineto
                     stroke
                     grestore"
                       } #}))
             #})
            (added-stencil (grob-interpret-markup grob added-markup)))
      (if (ly:stencil? original)
          (ly:stencil-add original added-stencil)
          added-stencil))))


slashNoteOn =
\override Voice.NoteHead.stencil =
#(grob-transformer
  'stencil
```

```
58    (lambda (grob original)
59      (let* ((added-markup
60              #{
61                \markup \general-align #Y #CENTER
62                #(case (ly:grob-property grob 'duration-log)
63                   ((0) #{ \markup \concat {
64                     \musicglyph "noteheads.s0"
65                     \postscript
66                     "gsave
67                      0.17 setlinewidth
68                      -2.3 0.6 moveto
69                      0.3 -0.6 lineto
70                      stroke
71                      grestore"
72                        } #})
73                   ((1) #{ \markup \concat {
74                     \musicglyph "noteheads.s1"
75                     \postscript
76                     "gsave
77                      0.17 setlinewidth
78                      -1.5 0.6 moveto
79                      0.3 -0.6 lineto
80                      stroke
81                      grestore"
82                        } #})
83                   ((2) #{ \markup \concat {
84                     \musicglyph "noteheads.s2"
85                     \postscript
86                     "gsave
87                      0.17 setlinewidth
88                      -1.5 0.6 moveto
89                      0.3 -0.6 lineto
90                      stroke
91                      grestore"
92                        } #}))
93              #})
94             (added-stencil (grob-interpret-markup grob added-markup)))
95        (if (ly:stencil? original)
96            (ly:stencil-add original added-stencil)
97            added-stencil))))
98
```

```
99
100 slashNoteOff = \revert Voice.NoteHead.stencil
101
102 {
103   \time 7/4
104   \slashNote c'4
105   \slashNote d'2
106   \slashNote e'1
107   \slashNoteOn g''4 f''2 d''1
108   \slashNoteOff c''1 \bar "||"
109 }
```

Table of Contents

## 5.7 Square Notehead



### 5.7.1 Description

Filled and hollow square noteheads.

### 5.7.2 Grammar

```
\squareHollowNotehead NOTE
\squareHollowNoteheadOn NOTES
\squareHollowNoteheadOff
\squareFilledNotehead NOTE
\squareFilledNoteheadOn NOTES
\squareFilledNoteheadOff

\slashNoteOn NOTE
\slashNoteOff
```

**NB** `\squareHollowNotehead` and `\squareFilledNotehead` only affect one note, owing to the `\once \override` functions within the code. For a group of notes, use `\squareHollowNoteheadOn` or `\squareFilledNoteheadOn`. `\squareHollowNoteheadOff` and `\squareFilledNoteheadOff` cancel the application.

### 5.7.3 Code

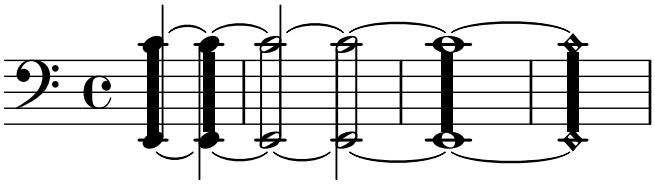
```
\version "2.24.4"

% See also: https://lsr.di.unimi.it/LSR/Item?id=516

squareHollowNoteheadDesign =
#(ly:make-stencil '(path 0.15 (moveto  0.05 0.425
                                       rlineto 1. 0
                                       rlineto 0 -0.875
                                       rlineto -1. 0
                                       closepath)
                         )
                  (cons -0.025 1.125)
                  (cons -1 1))
```

```
14
15 squareHollowNotehead =
16 #(define-music-function (note) (ly:music?)
17    #{\once \override Voice.NoteHead.stencil =
18      \squareHollowNoteheadDesign $note #})
19
20 squareHollowNoteheadOn =
21 #(define-music-function (note) (ly:music?)
22    #{\override Voice.NoteHead.stencil =
23      \squareHollowNoteheadDesign $note #})
24
25 squareHollowNoteheadOff = \revert Voice.NoteHead.stencil
26
27 squareFilledNoteheadDesign =
28 #(ly:make-stencil '(path 0.15 (moveto  0.05 0.425
29                                        rlineto 1. 0
30                                        rlineto 0 -0.875
31                                        rlineto -1. 0
32                                        closepath)
33                           round
34                           round
35                           #t)
36                   (cons -0.025 1.125)
37                   (cons -1 1))
38
39
40 squareFilledNotehead =
41 #(define-music-function (note) (ly:music?)
42    #{\once \override Voice.NoteHead.stencil =
43      \squareFilledNoteheadDesign $note #})
44 squareFilledNoteheadOn =
45 #(define-music-function (note) (ly:music?)
46    #{\override Voice.NoteHead.stencil =
47      \squareFilledNoteheadDesign $note #})
48
49 squareFilledNoteheadOff = \revert Voice.NoteHead.stencil
50
51 {
52   \squareHollowNotehead c'8
53   \squareHollowNoteheadOn d' e' f'
54   \squareHollowNoteheadOff
```

```
55    \squareFilledNotehead c'8
56    \squareFilledNoteheadOn d' e' f'
57    \squareFilledNoteheadOff
58    \squareHollowNotehead a''8
59    \squareHollowNoteheadOn g'' f'' e''
60    \squareHollowNoteheadOff
61    \squareFilledNotehead a''8
62    \squareFilledNoteheadOn g'' f'' e''
63    \squareFilledNoteheadOff
64  }
```

**Table of Contents**

## 5.8 Tone Cluster



### 5.8.1 Description

Inspired by the tone cluster notation of Henry Cowell and others. See **Discussion**.

### 5.8.2 Grammar

```
\toneClusterBar NOTE1 NOTE2 yOffset yLengthAdjust
\toneClusterBarHollow NOTE1 NOTE2 yOffset yLengthAdjust
\toneClusterBarWhole NOTE1 NOTE2 yOffset yLengthAdjust
```

**NB**

1. The order of pitch boundaries as shown by `NOTE1` and `NOTE2` does not matter; `NOTE1` can be upper or lower pitch boundary, and vice versa for `NOTE2`. See **Code**.

2. `yOffset` indicates where the upper part of the cluster sign begins. When set to `#0`, it starts right at the top line of the ordinary 5-line staff. Each positive/negative integer will bring the beginning point up/down by a space of two neighboring lines of the staff.

3. `yLengthAdjust` indicates any value by which the cluster bar may be extended or reduced. When set to `#0`, the cluster bar will be as long as the distance between the lower boundary of the upper notehead and upper boundary of the lower notehead. Each positive/negative integer will add/reduce the length of the bar by a space of two neighboring lines of the staff.

   For this reason, when the tone cluster sign is applied to a quarter-note dyad, you may wish to set the upper part of the cluster bar right in the middle of the notehead. In the snippet shown, the first cluster's `yOffset` is set to `#1`. `yLengthAdjust` is also set to `#1`, meaning that the cluster bar will go down to the center of the lower notehead. The second cluster intentionally shows what happens when the bar only touches the two boundaries of the noteheads.

4. `\toneClusterBarHollow` shows the notation (quite à la Cowell) specifically for hollowed noteheads. Some people may prefer this notation, instead.

5. `\toneClusterBarWhole` is specifically for the tone cluster notation as applied to a whole-note dyad, owing to width being wider than the quarter or half noteheads.

6. These functions may be used in tandem with other noteheads, as well as ties. See **Code**.

### 5.8.3 Code

```
1
2  toneClusterBar =
3  #(define-music-function (note1 note2 yOffset yLengthAdjust)
4     (ly:music? ly:music? number? number?)
5     (let* (
6             (note1p (ly:music-property note1 'pitch))
7             (note2p (ly:music-property note2 'pitch))
8             (note1pnumber (+ (* 7 (ly:pitch-octave note1p))
9                              (ly:pitch-notename note1p)))
10            (note2pnumber (+ (* 7 (ly:pitch-octave note2p))
11                             (ly:pitch-notename note2p)))
12            (pitchDistance (abs (- note1pnumber note2pnumber)))
13            )
14       #{
15         < #note1
16         #note2 > ^\markup {
17           \postscript
18           #(string-append
19             "gsave
20              newpath
21              0.3 " (number->string (- yOffset 0.5)) " moveto
22              0.7 0 rlineto
23              0 " (number->string (- (* -0.5 pitchDistance)
24                                     (- yLengthAdjust 1))) " rlineto
25              -0.7 0 rlineto
26              closepath
27              fill
28              grestore")
29         }
30       #}
31       )
32    )
33
34
```

```
35 toneClusterBarHollow =
36 #(define-music-function (note1 note2 yOffset yLengthAdjust)
37    (ly:music? ly:music? number? number?)
38    (let* (
39            (note1p (ly:music-property note1 'pitch))
40            (note2p (ly:music-property note2 'pitch))
41            (note1pnumber (+ (* 7 (ly:pitch-octave note1p))
42                             (ly:pitch-notename note1p)))
43            (note2pnumber (+ (* 7 (ly:pitch-octave note2p))
44                             (ly:pitch-notename note2p)))
45            (pitchDistance (abs (- note1pnumber note2pnumber)))
46            )
47      #{
48        < #note1
49        #note2 > ^\markup {
50          \postscript
51          #(string-append
52            "gsave
53             newpath
54             0.1 " (number->string (- yOffset 0.5)) " moveto
55             0 " (number->string (- (* -0.5 pitchDistance)
56                                    (+ 0.5 yLengthAdjust))) " rlineto
57             0.125 setlinewidth
58             1.3 "(number->string (+ 0.75 (- yOffset 0.5))) " moveto
59             0 " (number->string (- (* -0.5 pitchDistance)
60            (+ 0.75 yLengthAdjust))) " rlineto
61             stroke
62             grestore")
63        }
64      #}
65      )
66    )
67
68
69 toneClusterBarWhole =
70 #(define-music-function (note1 note2 yOffset yLengthAdjust)
71    (ly:music? ly:music? number? number?)
72    (let* (
73            (note1p (ly:music-property note1 'pitch))
74            (note2p (ly:music-property note2 'pitch))
75            (note1pnumber (+ (* 7 (ly:pitch-octave note1p))
```

```
                              (ly:pitch-notename note1p)))
            (note2pnumber (+ (* 7 (ly:pitch-octave note2p))
                              (ly:pitch-notename note2p)))
            (pitchDistance (abs (- note1pnumber note2pnumber)))
            )
      #{
        < #note1
        #note2 > ^\markup {
          \postscript
          #(string-append
            "gsave
             newpath
             0.125 setlinewidth
             0.55 " (number->string (- yOffset 0.5)) " moveto
             0 " (number->string (- (* -0.5 pitchDistance)
                                    (- yLengthAdjust 1))) " rlineto
             0.75 0 rlineto
             0 " (number->string (abs (- (* -0.5 pitchDistance)
                 (- yLengthAdjust 1)))) " rlineto
             closepath fill
             grestore")
        }
      #}
      )
    )


{
  \time 4/4
  \partial 2
  \clef "F"
  \stemUp \toneClusterBar c'4~ e,~ #1 #1
  \stemDown \toneClusterBar e,~ c'4~ #0.5 #0
  \stemUp \toneClusterBarHollow c'2~ e,~ #0.5 #-0.5
  \stemDown \toneClusterBarHollow c'2~ e,~ #0.5 #-0.5
  \toneClusterBarWhole c'1~ e,~ #0.5 #0
  \toneClusterBar c'1~\harmonic e,~\harmonic #0.5 #0
}
```

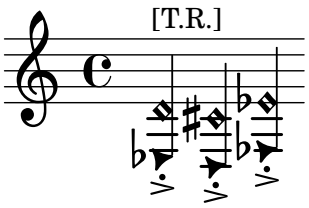
### 5.8.4 Discussion

There have been some discussions on `lilypond-user` mailing list in the past that readers may consult for further ideas on implementing different types of tone cluster notation:

- https://lists.gnu.org/archive/html/lilypond-user/2008-10/msg00484.html (This one in particular lists other notational conventions established by other composers)
- https://lists.gnu.org/archive/html/lilypond-user/2020-12/msg00130.html

**Table of Contents**

## 5.9 Tongue Ram (for flute)




### 5.9.1 Description

Implementation of the tongue ram notation, as described in *The Techniques of Flute Playing* by Carin Levine and Christina Mitropoulos-Bott.[4]

### 5.9.2 Grammar

```
\tgrWithIndication NOTE
\tgr NOTE
```

**NB**

1. `\language "english"` needs to be specified.
2. `\tgr` and `\tgrWithIndication` are followed by a pitch to be fingered on the instrument. The code will copy and reproduce a resultant pitch a major seventh down. Use `\tgrWithIndication` for showing the markup with the indication "T.R." (tongue ram). For more details, see: FluteXpansions.

### 5.9.3 Code

```
tgrWithIndication = #(define-music-function (note1) (ly:music?)
       (let*         (
                (p1 #{ #(ly:music-deep-copy note1) \harmonic #})
                (p2 #{ \transpose c df, #(ly:music-property note1 'pitch)#})
                (d1  (ly:music-property note1 'duration))
                )
          #{ < $p1
             \single \override NoteHead.stencil = #ly:text-interface::print
             \single \override NoteHead.text =
             \markup \musicglyph "noteheads.s2triangle"
             %\single \override Stem.stencil
             $p2 > $d1 ^\markup {\override #'(font-size . -2) {[T.R.]} }  #}
```

4. Levine and Mitropoulos-Bott, *The techniques of flute playing = Die Spieltechnik der Flöte*, 28.

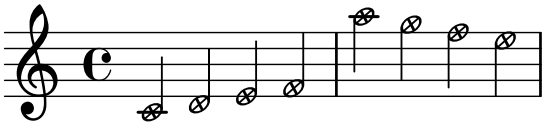
```
          ))
tgr = #(define-music-function (note1) (ly:music?)
          (let*         (
                   (p1 #{ #(ly:music-deep-copy note1) \harmonic #})
                   (p2 #{ \transpose c df, #(ly:music-property note1 'pitch)#})
                   (d1  (ly:music-property note1 'duration))
                   )
            #{ < $p1
               \single \override NoteHead.stencil = #ly:text-interface::print
               \single \override NoteHead.text =
               \markup \musicglyph "noteheads.s2triangle"
               %\single \override Stem.stencil
               $p2 > $d1  #}
            ))

{\language "english" \tgrWithIndication d'4-.-> \tgr cs'4-.-> \tgr ef'4-.->}
```

### 5.9.4 Discussion

I want to improve this code so that I can add markups to the note. It is slightly awkward at the moment.

**Table of Contents**

## 5.10 X In A Hollow Notehead



### 5.10.1 Description

While LilyPond Notation Reference provides an example of an X-in-a-circle notehead, its shape differs from the regular notehead.[5] This implementation simulates a hollow notehead with which the X notehead is combined.

### 5.10.2 Grammar

```
\cirX NOTE
```

### 5.10.3 Code

```
% Stem attachment function inspired by:
% https://lsr.di.unimi.it/LSR/Snippet?id=518
cirX = #(define-music-function (note) (ly:music?)
          #{
            \temporary \override NoteHead.stencil =
            #ly:text-interface::print
            \temporary \override NoteHead.text =
            \markup
            \translate #'(0.6 . 0)
            \pad-x #-0.22
            \rotate #35
            \scale #'(1 . 0.65)
            \combine \combine \combine \combine
            \override #'(thickness . 2)
            \draw-line #'(0.05 . 0.6)
            \override #'(thickness . 2)
            \draw-line #'(-0.05 . -0.6)
            \override #'(thickness . 2)
            \draw-line #'(0.6 . 0.1 )
            \override #'(thickness . 2)
            \draw-line #'(-0.6 . -0.1 )
            \draw-circle #0.65 #0.175 ##f
```

5. https://lilypond.org/doc/v2.24/Documentation/notation/modifying-stencils

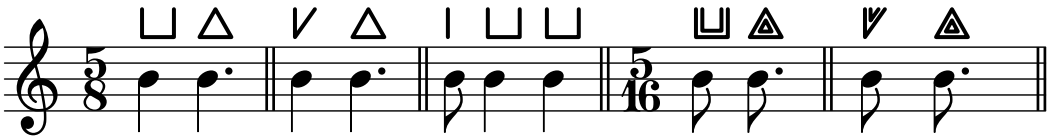
```
23
24              \temporary \override NoteHead.stem-attachment =
25              #(lambda (grob)
26                 (let* ((stem (ly:grob-object grob 'stem))
27                        (dir (ly:grob-property stem 'direction UP))
28                        (is-up (eqv? dir UP)))
29                   (cons dir (if is-up 0.2 -0.2))))
30              #note
31              \revert NoteHead.stencil
32              \revert NoteHead.text
33              \revert NoteHead.stem-attachment
34            #})
35  {
36    \cirX c'4 \cirX d' \cirX e' \cirX f'
37    \cirX a''4 \cirX g'' \cirX f'' \cirX  e''
38  }
```

**Table of Contents**

# Chapter 6

# Markups

## 6.1 Conducting Patterns



### 6.1.1 Description

Conducting patterns. While there are several examples of conducting patterns available on LSR,[1] the conducting shapes in my implementation are not affected by the horizontal length of given durations.

### 6.1.2 Grammar

```
NOTE \condOne
NOTE \condTwoA
NOTE \condTwoB
NOTE \condThree
NOTE \condDoubleTwoA
NOTE \condDoubleTwoB
NOTE \condDoubleThree
```

1. See: https://lsr.di.unimi.it/LSR/Item?id=523 and https://lsr.di.unimi.it/LSR/Item?id=259

### 6.1.3 Code

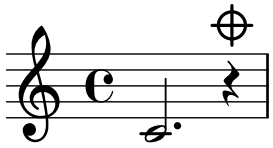
```
1
2  condOnePattern =
3  #'((moveto 0.25 1.75)
4     (rlineto 0 -1.75))
5
6  condTwoPatternA =
7  #'((moveto 0.25 1.75)
8     (rlineto 0 -1.75)
9     (rlineto 2 0)
10    (rlineto 0 1.75))
11
12 condDoubleTwoPatternA =
13 #'((moveto 0.25 1.75)
14    (rlineto 0 -1.75)
15    (rlineto 2 0)
16    (rlineto 0 1.75)
17    (moveto 0.65 1.75)
18    (rlineto 0 -1.35)
19    (rlineto 1.2 0)
20    (rlineto 0 1.35))
21
22 condTwoPatternB =
23 #'((moveto 0.25 1.75)
24    (rlineto 0 -1.75)
25    (rlineto 1.25 1.75))
26
27 condDoubleTwoPatternB =
28 #'((moveto 0.25 1.75)
29    (rlineto 0 -1.75)
30    (rlineto 1.25 1.75)
31    (moveto 0.6 1.75)
32    (rlineto 0 -0.7)
33    (rlineto 0.5 0.7))
34
35 condThreePattern =
36 #'((moveto 1.15 1.75)
37    (rlineto -1 -1.75)
38    (rlineto 2 0)
39    (closepath))
```

```
40
41  condDoubleThreePattern =
42  #'((moveto 1.15 1.75)
43     (rlineto -1 -1.75)
44     (rlineto 2 0)
45     (closepath)
46     (moveto 1.15 1.05)
47     (rlineto -0.385 -0.7)
48     (rlineto 0.75 0)
49     (closepath))
50
51
52  condOne = ^\markup {
53    \override #'(line-join-style . round)
54    \path #0.25 #condOnePattern
55  }
56
57  condTwoA = ^\markup {
58    \override #'(line-join-style . round)
59    \path #0.25 #condTwoPatternA
60  }
61  condTwoB = ^\markup {
62    \override #'(line-join-style . round)
63    \path #0.25 #condTwoPatternB
64  }
65  condDoubleTwoA = ^\markup {
66    \override #'(line-join-style . round)
67    \path #0.25 #condDoubleTwoPatternA
68  }
69
70  condDoubleTwoB = ^\markup {
71    \override #'(line-join-style . round)
72    \path #0.25 #condDoubleTwoPatternB
73  }
74
75  condThree = ^\markup {
76    \override #'(line-join-style . round)
77    \path #0.25 #condThreePattern
78  }
79
80  condDoubleThree = ^\markup {
```

```
  \override #'(line-join-style . round)
  \path #0.25 #condDoubleThreePattern
}

%% Source inspired by
%% and adapted from: http://lsr.di.unimi.it/LSR/Item?id=629
spacerVoice = \new Voice {
  \override MultiMeasureRest.transparent = ##t
  \override MultiMeasureRest.minimum-length = #14
  R16*5
}


\score {
  {
    \time 5/8
    b'4 \condTwoA b'4. \condThree \bar "||"
    b'4 \condTwoB b'4. \condThree \bar "||"
    b'8 \condOne b'4 \condTwoA b'4 \condTwoA \bar "||"
    \time 5/16
    << {b'8 \condDoubleTwoA b'8. \condDoubleThree}
            \spacerVoice >> \bar "||"
    << {b'8 \condDoubleTwoB b'8. \condDoubleThree}
            \spacerVoice >> \bar "||"
  }

}

```

Table of Contents

## 6.2 Mute Sign



### 6.2.1 Description

Implementation of the mute sign, used to indicate that vibrating strings must be dampened at a specified moment. Its provenance can be traced back to Carlos Salzedo's *Modern Study of the Harp*.[2]

### 6.2.2 Grammar

```
NOTE/REST^\mutesign
```

### 6.2.3 Code

```
mutesign = \markup {
  \translate #'(0.5 . 0)
  \postscript

  "newpath
0.2 setlinewidth
1 setlinecap
0 0 moveto
0 2.5 rlineto
-1.25 1.25 moveto
2.5 0 rlineto
stroke
newpath
0 1.25 0.85 0 360 arc
stroke"

{ c'2. r4^\mutesign }

```

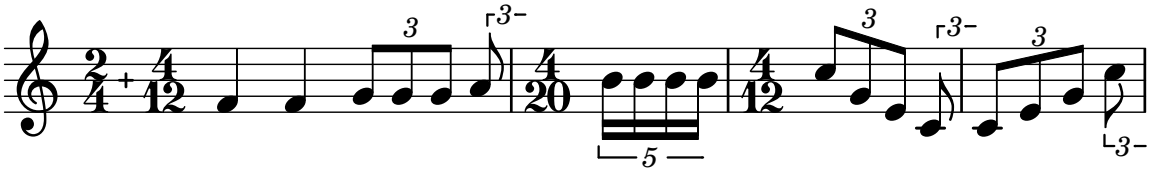
Table of Contents

2. Carlos Salzedo, *L'étude moderne de la harpe... Modern study of the harp* (New York - Boston, G. Schirmer, 1921), 19.

# Chapter 7

# Rhythm

## 7.1 Incomplete Tuplet Bracket for Irrational Time Signatures



### 7.1.1 Description

This entry supplements the irrational time signatures[1] as seen on LSR.[2]. Concerning the irrational time signatures, in her *Behind Bars: the Definitive Guide to Music Notation*, Elaine Gould suggests the use of denominator as any division of the semibreve/whole note..[3] However, in these pages there has not been a mention of the use of tuplet brackets while the non-conventional time signature is in place. There are examples, such as *Asyla* for large orchestra by Thomas Adès„[4] where tuplet brackets are placed atop ”incomplete” tuplets.

While it is still prudent to spend a paragraph explaining the nature of the irrational time signatures in the preface, my preference has also been to utilize incomplete tuplet

1. See Chapter **Time Signatures** for discussion on the variants of the irrational/fractional time signatures.
2. https://lsr.di.unimi.it/LSR/Snippet?id=552
3. Elaine Gould, *Behind bars : the definitive guide to music notation* (London: Faber Music, 2011), 180–181, Book.
4. Thomas Adès, *Asyla : for large orchestra* (Faber Music, 1997).

brackets, in order to allow the reading of the rhythm consistent and smooth from bars with ordinary time signatures. It is also helpful to have the brackets shown in cases of compound time signatures that use irrational time signatures in part (see the first measure of the example).

### 7.1.2 Grammar

```
\incompleteTupletBracket \tuplet ...
\incompleteSmallTupletBracket \tuplet ...
```

**NB**

1. For incomplete tuplets with two or more notes, use `\incompleteTupletBracket`.
2. For incomplete tuplets with one note, use `\incompleteSmallTupletBracket`. This was created specifically to ensure that the brackets appear properly in tight space that one-note tuplet customarily gives.

### 7.1.3 Code

```
\version "2.24.4"

%% "suppressWarning" function comes from:
%% http://lsr.di.unimi.it/LSR/Item?id=552

% Warnings may be suppressed using 'ly:expect-warning'
% Or use the here defined 'suppressWarning'-function, working since 2.20.

suppressWarning =
#(define-void-function (amount message)(number? string?)
   (for-each
    (lambda (warning)
      (ly:expect-warning message))
    (iota amount 1 1)))

\suppressWarning 3 "strange time signature found"

incompleteTupletBracket =  {
  \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
  \once \override Voice.TupletBracket.bracket-visibility = ##t

}
incompleteSmallTupletBracket =  {
```

```
24    \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
25    \once \override Voice.TupletBracket.bracket-visibility = ##t
26    \once \override Voice.TupletNumber.X-offset =
27    #(lambda (grob)
28       (if (= UP (ly:grob-property grob 'direction))
29           2.2
30           1.2))
31
32    \once \override Voice.TupletBracket.shorten-pair =
33    #(lambda (grob)
34       (if (= UP (ly:grob-property grob 'direction))
35           '(-0.7 . 0.15)
36           '(-0.3 . 0.8)))
37    \once \override Voice.TupletBracket.X-positions =
38    #(lambda (grob)
39       (if (= UP (ly:grob-property grob 'direction))
40           '(1.8 . 3)
41           '(0.3 . 2.7)))
42  }
43
44
45  {
46    \compoundMeter #'((2 4) (4 12))
47    f'4 f'
48    \tuplet 3/2 {g'8[ g' g']}
49    \incompleteSmallTupletBracket
50    \tuplet 3/2 {a'8 }|
51
52    \time 4/20
53    \incompleteTupletBracket
54    \tuplet 5/4 {b'16[ b' b' b']} |
55    \time 4/12
56    \tuplet 3/2 {c''8[ g' e']}
57    \incompleteSmallTupletBracket
58    \tuplet 3/2 {c'8} |
59    \tuplet 3/2 {c'8[ e' g']}
60    \incompleteSmallTupletBracket
61    \tuplet 3/2 {c''8} |
62  }
```

### 7.1.4 Discussion

In the preceding code, I have opted to notate the tuplets within the bars with irrational time signatures in an ordinary manner, using `\tuplet`. This is to ensure that the incomplete tuplet bracket appears. Compare this with the quoted LSR No. 552, which has a different way of reducing the note duration in order to fit them into the bar with irrational time signature. Observe the way duration is multiplied by fractions, e.g. Line 6.

```
1  {
2   \time 4/4
3   \tempo 4 = 60
4   fis4 fis fis fis
5   \time 2/6
6   g4*2/3 g |
7   g4*2/3 g |
8   \time 4/5
9   as4*4/5 as as as8*4/5 g |
10  \tuplet 3/2 { as4*4/5 as as } as4*4/5 as8*4/5 g |
11  \time 3/7
12  fis4*4/7 fis fis |
13  fis4*4/7 fis fis |
14 }
```

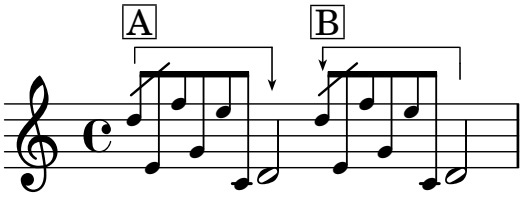
Table of Contents

# Chapter 8

# Spanners

This chapter covers snippets that take advantages of spanners (text, line, etc.) in one way or another. Because functions such as `\startTextSpan` and `\stopTextSpan` activate and deactivate these snippets, caution must be paid when using more than one of them at the same time. See Example in **Combinations** to avoid conflicts between or among the spanner snippets.

## 8.1 Grace Note Brackets I




### 8.1.1 Description

**NB**: See Grace Note Brackets II for the updated version of this code.) Replication of grace note brackets seen in scores by Pierre Boulez (e.g. *Sur Incises*,[1] *...explosante-fixe...*[2]). Bracket A in the example shows that the grace notes are to be played *before* the beat to which they are applied. Whereas Bracket B shows that the grace notes are to be played *on* the beat to which they are applied.

1. Pierre Boulez, *Sur incises : pour trois pianos, trois harpes et trois percussions-claviers (1996/1998)* (Universal Edition, 1998).
2. Pierre Boulez, *... explosante-fixe ... transitoire VII : (version 1991/93)* (Universal Edition, 1991).

### 8.1.2 Grammar

```
\graceNoteBeforeBeatOn NOTE
\graceNoteBeforeBeatOff NOTE
\graceNoteAfterBeatOn NOTE
\graceNoteAfterBeatOff NOTE
```

### 8.1.3 Code

```
\version "2.24.4"

\language "english"

% This code includes snippet for grace note
% slashes, which has been taken from:
% https://lsr.di.unimi.it/LSR/Item?id=1048


graceNoteBeforeBeatOn =
#(define-music-function (starting_note) (ly:music?)
   #{
     \once \override TextSpanner.style = #'line
     \once \override TextSpanner.bound-details.left.text =
     \markup { \draw-line #'(0 . -1) }
     \once \override TextSpanner.bound-details.right.text =
     \markup {
       \postscript
       "newpath 0 0 moveto
0 -2.5 rlineto
stroke
newpath
-0.275 -2 moveto
0.275 -0.75 rlineto
0.275 0.75 rlineto
-0.275 -0.2 rlineto
closepath
fill"
     }
     \once \override TextSpanner.Y-offset = #5
```

```
      \once \override TextSpanner.bound-details.left.padding = #0.5
      \once \override TextSpanner.bound-details.right.padding = #-0.25
      #starting_note
      \startTextSpan
   #})


graceNoteBeforeBeatOff =
#(define-music-function (ending_note) (ly:music?)
   #{
     #ending_note
     \stopTextSpan
   #})


graceNoteAfterBeatOn =
#(define-music-function (starting_note) (ly:music?)
   #{
     \once \override TextSpanner.style = #'line
     \once \override TextSpanner.bound-details.right.text =
     \markup {
       \combine \draw-line #'(0 . -1)
       \postscript "newpath
0 -1 moveto
0 -1 rlineto
stroke"
     }
     \once \override TextSpanner.bound-details.left.text =
     \markup {
       \postscript
       "newpath 0 0 moveto
0 -1 rlineto
stroke
newpath
-0.275 -0.75 moveto
0.275 -0.75 rlineto
0.275 0.75 rlineto
-0.275 -0.2 rlineto
closepath
fill"
     }
```

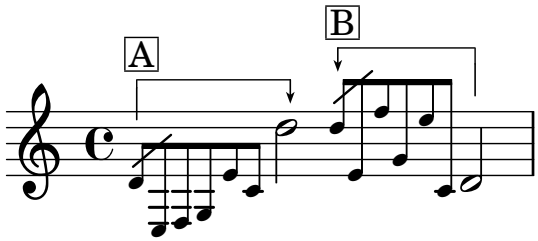
```
      \once \override TextSpanner.Y-offset = #2
      \once \override TextSpanner.bound-details.left.padding = #0.5
      \once \override TextSpanner.bound-details.right.padding = #-0.25
      #starting_note
      \startTextSpan
   #})


graceNoteAfterBeatOff =
#(define-music-function (ending_note) (ly:music?)
   #{
     #ending_note
     \stopTextSpan
   #})

%%%%%%%%%%%%%%%%%LSR SNIPPET START%%%%%%%%%%%%%%%%%

#(define (degrees->radians deg)
   (* PI (/ deg 180.0)))

slash =
#(define-music-function (ang stem-fraction protrusion)
   (number? number? number?)
   (remove-grace-property 'Voice 'Stem 'direction)
   #{
     \once \override Stem.stencil =
     #(lambda (grob)
        (let* ((x-parent (ly:grob-parent grob X))
               (is-rest? (ly:grob?
                          (ly:grob-object x-parent 'rest)))
               (beam (ly:grob-object grob 'beam))
               (stil (ly:stem::print grob)))
          (cond
           (is-rest? empty-stencil)
           ((ly:grob? beam)
            (let* ((refp (ly:grob-system grob))
                   (stem-y-ext (ly:grob-extent grob grob Y))
                   (stem-length
                    (- (cdr stem-y-ext) (car stem-y-ext)))
                   (beam-X-pos (ly:grob-property beam 'X-positions))
                   (beam-Y-pos (ly:grob-property beam 'positions))
```

```
                (beam-slope (/ (- (cdr beam-Y-pos) (car beam-Y-pos))
                               (- (cdr beam-X-pos) (car beam-X-pos))))
                (beam-angle (atan beam-slope))
                (dir (ly:grob-property grob 'direction))
                (line-dy (* stem-length stem-fraction))
                (line-dy-with-protrusions (if (= dir 1)
                               (+ (* 4 protrusion) beam-angle)
                               (- (* 4 protrusion) beam-angle)))
                (ang (if (> beam-slope 0)
                         (if (= dir 1)
                             (+ (degrees->radians ang) (* beam-angle 0.7))
                             (degrees->radians ang))
                         (if (= dir 1)
                             (degrees->radians ang)
                             (- (degrees->radians ang) (* beam-angle 0.7)))))
                (line-dx (/ line-dy-with-protrusions (tan ang)))
                (protrusion-dx (/ protrusion (tan ang)))
                (corr (if (= dir 1) (car stem-y-ext) (cdr stem-y-ext))))
           (ly:stencil-add
            stil
            (grob-interpret-markup grob
                                   (markup
                                     #:translate
                                     (cons (- protrusion-dx)
                                     (+ corr
                                        (* dir
                                           (- stem-length
                                              (+ stem-fraction protrusion)))))
                                     #:override '(thickness . 1.7)
                                     #:draw-line
                                     (cons line-dx
                                      (* dir line-dy-with-protrusions)))))))
          (else stil))))
   #})

startSlashedGraceMusic = {
  \slash 40 1 0.5
  \override Flag.stroke-style = #"grace"
}
stopSlashedGraceMusic = {
  \revert Flag.stroke-style
```

```
}

startAcciaccaturaMusic = {
  \slash 40 1 0.5
  s1*0(
  \override Flag.stroke-style = #"grace"
}
stopAcciaccaturaMusic = {
  \revert Flag.stroke-style
  s1*0)
}
%%%%%%%%%%%%%%%%%% LSR SNIPPET END %%%%%%%%%%%%%%%%%%


{
  \grace {
    \startSlashedGraceMusic
    \graceNoteBeforeBeatOn d''8^\markup{\box A} e' f'' g' e'' c'
  }
  \graceNoteBeforeBeatOff d'2
  \grace {
    \startSlashedGraceMusic
    \graceNoteAfterBeatOn d''8^\markup{\box B} e' f'' g' e'' c'
  }
  \graceNoteAfterBeatOff d'2
}
```

**Table of Contents**

## 8.2 Grace Note Brackets II




### 8.2.1 Description

This is an updated version of Grace Note Brackets I. It differs from the original version in that this version takes a list of three parameters in order to finely adjust the shape of the bracket in order to accommodate various shapes of grace notes and the actual note.

### 8.2.2 Grammar

```
\graceNoteBeforeBeatOn #'(OVERALL LEFT RIGHT) NOTE
\graceNoteBeforeBeatOff #'(OVERALL LEFT RIGHT) NOTE
\graceNoteAfterBeatOn #'(OVERALL LEFT RIGHT) NOTE
\graceNoteAfterBeatOff #'(OVERALL LEFT RIGHT) NOTE
```

**NB** The list accepts three integers as parameters, i.e.:

1. `OVERALL` is a value of the distance between the top line of the staff and the horizontal line of the grace note bracket. This value cannot be smaller than the skyline value established by the staff line and the notes; when the skyline value is greater than what is specified in this bracket, the skyline value is favored. When in doubt, start with `0`, then increase the amount gradually.
2. `LEFT` and `RIGHT` values (negative value only!) adjust the lengths of the left and right hooks.

### 8.2.3 Code

```
1 \version "2.24.4"
2
3 \language "english"
4
5 % This code includes snippet for grace note
```

```
% slashes, which has been taken from:
% https://lsr.di.unimi.it/LSR/Item?id=1048


graceNoteBeforeBeatOn =
#(define-music-function (setting-list starting_note) (list? ly:music? )
  #{
   \once \override TextSpanner.style = #'line
   \once \override TextSpanner.bound-details.left.text =
   \markup {
    \combine
    \draw-line #(cons 0 -0.5)
    \postscript #(string-append "newpath
0 -0.5 moveto
0 " (number->string (cadr setting-list)) " rlineto
stroke")
   }
   \once \override TextSpanner.bound-details.right.text =
   \markup {
    \postscript
    #(string-append "newpath 0 0 moveto
0 " (number->string (caddr setting-list))  " rlineto
stroke
newpath
-0.275 " (number->string (+ (caddr setting-list) 0.25))  " moveto
0.275 -0.75 rlineto
0.275 0.75 rlineto
-0.275 -0.2 rlineto
closepath
fill")
   }
   \once \override TextSpanner.Y-offset = #(car setting-list)
   \once \override TextSpanner.bound-details.left.padding = #0.5
   \once \override TextSpanner.bound-details.right.padding = #-0.25
   #starting_note
   \startTextSpan
  #})


graceNoteBeforeBeatOff =
#(define-music-function (ending_note) (ly:music?)
```

```
  #{
   #ending_note
   \stopTextSpan
  #})


graceNoteAfterBeatOn =
#(define-music-function (setting-list starting_note) (list? ly:music?)
  #{
   \once \override TextSpanner.style = #'line
   \once \override TextSpanner.bound-details.right.text =
   \markup {
    \combine
    \draw-line #(cons 0 -1)
    \postscript #(string-append "newpath
0 -1 moveto
0 " (number->string (caddr setting-list)) " rlineto
stroke")
   }
   \once \override TextSpanner.bound-details.left.text =
   \markup {
    \postscript
    #(string-append "newpath 0 0 moveto
0 " (number->string (cadr setting-list))  " rlineto
stroke
newpath
-0.275 " (number->string (+ (cadr setting-list) 0.25))  " moveto
0.275 -0.75 rlineto
0.275 0.75 rlineto
-0.275 -0.2 rlineto
closepath
fill")
   }
   \once \override TextSpanner.Y-offset = #(car setting-list)
   \once \override TextSpanner.bound-details.left.padding = #0.5
   \once \override TextSpanner.bound-details.right.padding = #-0.25
   #starting_note
   \startTextSpan
  #})


```

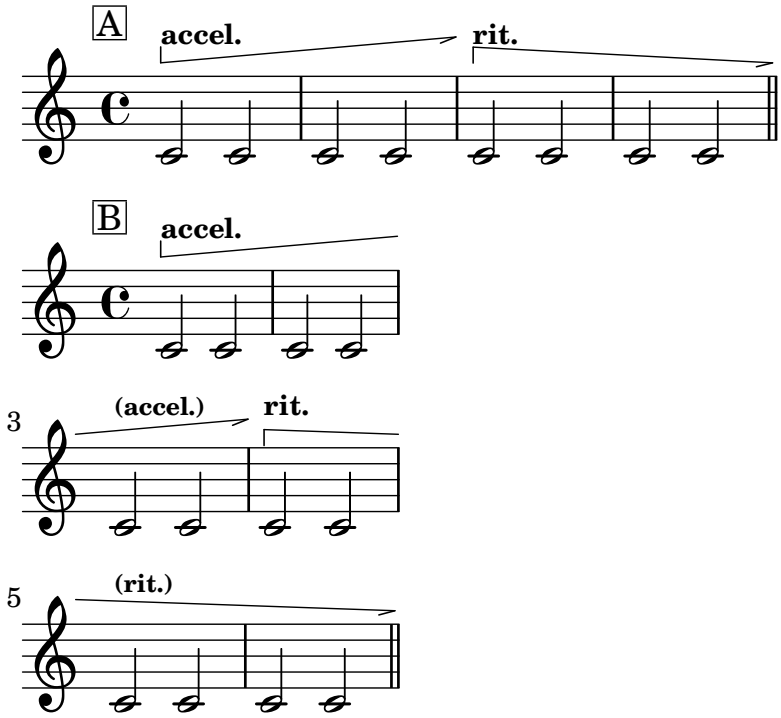
```
graceNoteAfterBeatOff =
#(define-music-function (ending_note) (ly:music?)
  #{
   #ending_note
   \stopTextSpan
  #})

%%%%%%%%%%%%%%%%%%LSR SNIPPET START%%%%%%%%%%%%%%%%%%

#(define (degrees->radians deg)
  (* PI (/ deg 180.0)))

slash =
#(define-music-function (ang stem-fraction protrusion)
  (number? number? number?)
  (remove-grace-property 'Voice 'Stem 'direction)
  #{
   \once \override Stem.stencil =
   #(lambda (grob)
     (let* ((x-parent (ly:grob-parent grob X))
            (is-rest? (ly:grob?
                       (ly:grob-object x-parent 'rest)))
            (beam (ly:grob-object grob 'beam))
            (stil (ly:stem::print grob)))
      (cond
       (is-rest? empty-stencil)
       ((ly:grob? beam)
        (let* ((refp (ly:grob-system grob))
               (stem-y-ext (ly:grob-extent grob grob Y))
               (stem-length
                (- (cdr stem-y-ext) (car stem-y-ext)))
               (beam-X-pos (ly:grob-property beam 'X-positions))
               (beam-Y-pos (ly:grob-property beam 'positions))
               (beam-slope (/ (- (cdr beam-Y-pos) (car beam-Y-pos))
                              (- (cdr beam-X-pos) (car beam-X-pos))))
               (beam-angle (atan beam-slope))
               (dir (ly:grob-property grob 'direction))
               (line-dy (* stem-length stem-fraction))
               (line-dy-with-protrusions (if (= dir 1)
                                             (+ (* 4 protrusion) beam-angle)
                                             (- (* 4 protrusion) beam-angle)))
```

```
129                 (ang (if (> beam-slope 0)
130                          (if (= dir 1)
131                              (+ (degrees->radians ang) (* beam-angle 0.7))
132                              (degrees->radians ang))
133                          (if (= dir 1)
134                              (degrees->radians ang)
135                              (- (degrees->radians ang) (* beam-angle 0.7)))))
136                 (line-dx (/ line-dy-with-protrusions (tan ang)))
137                 (protrusion-dx (/ protrusion (tan ang)))
138                 (corr (if (= dir 1) (car stem-y-ext) (cdr stem-y-ext))))
139            (ly:stencil-add
140             stil
141             (grob-interpret-markup grob
142                                    (markup
143                                     #:translate
144                                     (cons (- protrusion-dx)
145                                           (+ corr
146                                              (* dir
147                                                 (- stem-length
148                                                    (+ stem-fraction protrusion)))))
149                                     #:override '(thickness . 1.7)
150                                     #:draw-line
151                                     (cons line-dx
152                                           (* dir line-dy-with-protrusions)))))))
153           (else stil))))
154    #})
155
156  startSlashedGraceMusic = {
157   \slash 40 1 0.5
158   \override Flag.stroke-style = #"grace"
159  }
160  stopSlashedGraceMusic = {
161   \revert Flag.stroke-style
162  }
163
164  startAcciaccaturaMusic = {
165   \slash 40 1 0.5
166   s1*0(
167   \override Flag.stroke-style = #"grace"
168  }
169  stopAcciaccaturaMusic = {
```

```
  \revert Flag.stroke-style
  s1*0)
 }
 %%%%%%%%%%%%%%%%%% LSR SNIPPET END %%%%%%%%%%%%%%%%%%

 {
  \grace {
   \startSlashedGraceMusic
   \graceNoteBeforeBeatOn #'(4 -2 -1) d'8^\markup{\box A} e f g e' c'
  }
  \graceNoteBeforeBeatOff d''2
  \grace {
   \startSlashedGraceMusic
   \graceNoteAfterBeatOn #'(6 -1 -2) d''8^\markup{\box B} e' f'' g' e'' c'
  }
  \graceNoteAfterBeatOff d'2
 }
```

**Table of Contents**

## 8.3 Tempo Arrows




### 8.3.1 Description

Replication of accelerando and rallentando arrows chiefly seen in scores by Tōru Takemitsu.[3] The snippets also handle line break.

### 8.3.2 Grammar

```
\accelArrow #Line_angle ... \stopTextSpan
\rallArrow #Line_angle ... \stopTextSpan
```

**NB**

1. `#Line_angle` sets how angled the horizontal line should be. `#5` should be more than sufficient for a short line. When it goes over a line break or it extends for a long time, a smaller number may be recommended, such as `#2`.

3. Examples abound, but see: Tōru Takemitsu, *Fantasma/cantos : for clarinet and orchestra* (Schott ; Schott Japan, 1993) and Tōru Takemitsu, *Les yeux clos II : for piano* (Schott ; Schott Japan, 1990) Other composers from the same publishing company, e.g. Toshio Hosokawa, have also adopted variants of the arrows in their music.

2. These commands only set the tempo arrows; as such, indications such as `accel.` and `rall.` need to be added separately.

### 8.3.3 Code

```
\version "2.24.4"

% freely modified from: https://lsr.di.unimi.it/LSR/Item?id=1168
% as well as http://lsr.di.unimi.it/LSR/Item?id=1023


accelArrow =
#(define-music-function (line_angle) (number?)

   (define x_value (cos (* (/ 3.14159265358979 180) (- 90 line_angle))))
   (define y_value (sin (* (/ 3.14159265358979 180) (- 90 line_angle))))
   #{
     \tweak direction #up
     \tweak style  #'line
     \tweak thickness  #1
     \tweak to-barline ##t
     \tweak rotation #(list line_angle -1 0 )
     \tweak bound-details.left.stencil #ly:text-interface::print
     \tweak bound-details.left.text \markup \postscript
     #(string-append
       "gsave newpath
0 0 moveto "
       (number->string x_value) " "
       (number->string y_value)
       " rlineto
stroke
grestore")
     \tweak bound-details.left-broken.stencil #ly:text-interface::print
     \tweak bound-details.left-broken.text ##f

     \tweak bound-details.right.stencil #ly:text-interface::print
     \tweak bound-details.right.text \markup \postscript
     "newpath
0 0 moveto
-1 -0.3 rlineto
stroke"
```

```
      \tweak bound-details.right-broken.stencil #ly:text-interface::print
      \tweak bound-details.right-broken.text ##f
      \tweak font-shape  #'upright
      \tweak bound-details.left.padding  #0
      \tweak bound-details.right.padding  #0
      \tweak breakable ##t
      \tweak after-line-breaking ##t

      \startTextSpan
   #})

rallArrow =
#(define-music-function (line_angle) (number?)

   (define x_value (cos (* (/ 3.14159265358979 180) (- 90 line_angle))))
   (define y_value (sin (* (/ 3.14159265358979 180) (- 90 line_angle))))
   #{
     \tweak direction #up
     \tweak style  #'line
     \tweak thickness  #1
     \tweak to-barline ##t
     \tweak rotation #(list (* -1 line_angle) 1 0 )
     \tweak bound-details.left.stencil #ly:text-interface::print
     \tweak bound-details.left.text \markup \postscript
     #(string-append
       "gsave
newpath
0 0 moveto "
       (number->string x_value) " "
       (number->string (* -1 y_value))
       " rlineto
stroke
grestore")
     \tweak bound-details.left-broken.stencil #ly:text-interface::print
     \tweak bound-details.left-broken.text ##f

     \tweak bound-details.right.stencil #ly:text-interface::print
     \tweak bound-details.right.text \markup \postscript
     "newpath
0 0 moveto
-1 -0.3 rlineto
```

```
78  stroke"
79       \tweak bound-details.right-broken.stencil #ly:text-interface::print
80       \tweak bound-details.right-broken.text ##f
81       \tweak font-shape  #'upright
82       \tweak bound-details.left.padding  #0
83       \tweak bound-details.right.padding  #0
84       \tweak breakable ##t
85       \tweak after-line-breaking ##t
86
87       \startTextSpan
88     #})
89
90  \score {
91    \layout {
92      indent = 0
93    }
94    {
95      c'2^\markup{\translate #'(-4 . 2) \box "A"}
96      ^\markup {\translate #'(0 . 1.5) \tiny \bold "accel."}
97              \accelArrow #5   c'2
98      c'2 \after 2 \stopTextSpan c'2
99      c'2 ^\markup {\translate #'(0 . 1.5) \tiny \bold "rit."}
100             \rallArrow #3   c'2
101     c'2 \after 2 \stopTextSpan c'2 \bar "||"
102   }
103 }
104
105 \score {
106   \layout {
107     indent = 0
108     line-width = 40
109   }
110   {
111     c'2^\markup{\translate #'(-4 . 2) \box "B"}
112     ^\markup {\translate #'(0 . 1.5) \tiny \bold "accel."}
113             \accelArrow #5 c'2
114     c'2 c'2
115     c'2^\markup {\translate #'(0 . 1.5) \teeny \bold "(accel.)"}
116             \after 2 \stopTextSpan c'2
117     c'2 ^\markup {\translate #'(0 . 1.5) \tiny \bold "rit."}
118             \rallArrow #2 c'2 \break
```

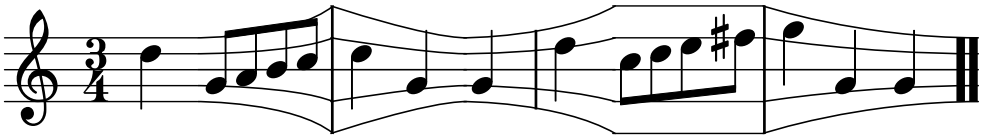
```
119       c'2^\markup {\translate #'(0 . 1.5) \teeny \bold "(rit.)"} c'2
120       c'2 \after 2 \stopTextSpan c'2 \bar "||"
121     }
122   }
```

**Table of Contents**

# Chapter 9

# Staff Lines

## 9.1 Expanding, Shrinking and Bloated Staff Lines



### 9.1.1 Description

I made this code as a proof of concept after having read some excellent snippets on LSR.[1]

### 9.1.2 Grammar

```
\expandingStaff #X-length
\shrinkingStaff #X-length
\bloatedStaff
\normalStaff
```

### 9.1.3 Code

```
1
2  shrinkingStaff =
```

1. See: https://lsr.di.unimi.it/LSR/Item?id=878, https://lsr.di.unimi.it/LSR/Item?id=1005, and https://lsr.di.unimi.it/LSR/Item?id=1007.

```
#(define-music-function
  (staffDist)
  (number?)

  #{
    \stopStaff
    \once  \override Staff.StaffSymbol.stencil = #ly:text-interface::print
    \once  \override Staff.StaffSymbol.text = \markup {
      \postscript #(string-append
          "newpath
          0 4 moveto
          0 4 6 2 " (number->string staffDist) " 2 curveto
          0 2 moveto
          0 2 6 1 " (number->string staffDist) " 1 curveto
          0 0 moveto "
          (number->string staffDist) " 0 lineto
          0 -2 moveto
          0 -2 6 -1 " (number->string staffDist) " -1 curveto
          0 -4 moveto
          0 -4 6 -2 " (number->string staffDist) " -2 curveto
          stroke")


    }
    \override Staff.StaffSymbol.line-positions = #'(-4 -2 0 2 4 )
    \startStaff
  #})

normalStaff = {
  \stopStaff
  \revert Staff.StaffSymbol.line-positions
  \revert Staff.StaffSymbol.stencil
  \startStaff
}

expandingStaff =
#(define-music-function
  (staffDist)
  (number?)

  #{
```

```
44
45      \stopStaff
46      \once  \override Staff.StaffSymbol.stencil = #ly:text-interface::print
47      \once  \override Staff.StaffSymbol.text = \markup {
48        \postscript #(string-append
49            "newpath
50            0 2 moveto
51            0 2 6 2 " (number->string staffDist) " 4 curveto
52            0 1 moveto
53            0 1 6 1 " (number->string staffDist) " 2 curveto
54            0 0 moveto "
55            (number->string staffDist) " 0 lineto
56            0 -1 moveto
57            0 -1 6 -1 " (number->string staffDist) " -2 curveto
58            0 -2 moveto
59            0 -2 6 -2 " (number->string staffDist) " -4 curveto
60            stroke ")
61      }
62
63      \startStaff
64      \override Staff.StaffSymbol.line-positions = #'(-8 -4 0 4 8 )
65    #})
66
67  bloatedStaff = {
68    \stopStaff
69    \override Staff.StaffSymbol.line-positions = #'(-8 -4 0 4 8 )
70    \override Staff.LedgerLineSpanner.stencil = ##f
71    \startStaff}
72
73
74
75  % to adjust the length of the individual barlines, see:
76  % https://lilypond.org/doc/v2.24/Documentation/internals/barline
77
78  {
79
80    \override Staff.LedgerLineSpanner.transparent = ##t
81    \numericTimeSignature
82    \time 3/4
83    \once \override Staff.BarLine.bar-extent = #'(-2 . 2)
84    d''4 \expandingStaff #8.5
```

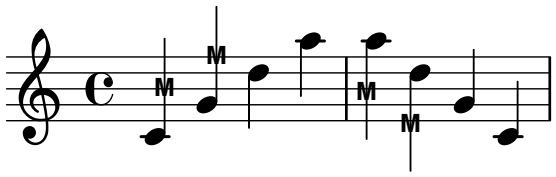
```
85
86    g'8 a' b' c''
87    \once \override Staff.BarLine.bar-extent = #'(-4 . 4)
88    \shrinkingStaff #8.5
89    d''4 g' \expandingStaff #9.5 g'
90    \once \override Staff.BarLine.bar-extent = #'(-2.5 . 2.5)
91
92
93    e''4 \bloatedStaff c''8 d'' e'' fs''
94    \once \override Staff.BarLine.bar-extent = #'(-4 . 4)
95
96    \shrinkingStaff #13.5
97
98    g''4 g' g'
99    \bar ".."
100
101 }
102
103 \layout {
104   \context{
105     \Score   proportionalNotationDuration = #(ly:make-moment 1/6)
106   }
107 }
108
109
```

Table of Contents

# Chapter 10

# Stems

## 10.1 ”M” on Stem




### 10.1.1 Description

This function attaches ”M” to the stem. I have used this to indicate **M**ultiphonics on woodwind instruments in my pieces. This function lengthens the stem in order to give a balanced look, especially combined with stems/flags.

### 10.1.2 Grammar

```
\MOnStemOn NOTE ...
\MOnStemOff
```

**NB** `\MOnStemOn` toggles the feature on, while `\MOnStemOff` toggles it off.

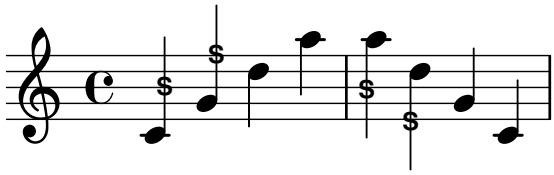
### 10.1.3 Code

```
1  MOnStemOn = {
2    \override Stem.length = #12
3    \override Stem.details.beamed-lengths = #'(5.5)
4    \override Stem.stencil =
5    #(lambda (grob)
```

```
      (let* ((x-parent (ly:grob-parent grob X))
             (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
        (if is-rest?
            empty-stencil
            (ly:stencil-combine-at-edge
             (ly:stem::print grob)
             Y
             (- (ly:grob-property grob 'direction))
             (grob-interpret-markup grob
                                    (markup
                                     #:center-align
                                     #:teeny #:sans #:bold "M"))
             -3.5))))
}

MOnStemOff = {
  \revert Stem.length
  \revert Stem.details.beamed-lengths
  \revert Stem.stencil
  \revert Flag.stencil
}

{
  \MOnStemOn c'4 g' \MOnStemOff d'' a''
  \MOnStemOn a'' d'' \MOnStemOff g' c'
}
```

Table of Contents

## 10.2 ”S” on Stem



### 10.2.1 Description

This function attaches ”S” to the stem. I have used this to indicate **S**plit tone on clarinet/bass clarinet in my pieces. This function lengthens the stem in order to give a balanced look, especially combined with stems/flags.

### 10.2.2 Grammar

```
\SOnStemOn NOTE ...
\SOnStemOff
```

**NB** `\SOnStemOn` toggles the feature on, while `\SOnStemOff` toggles it off.

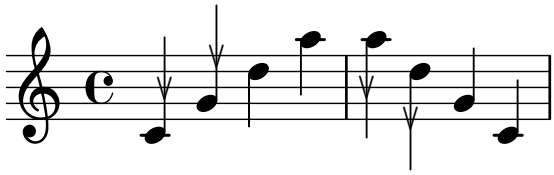
### 10.2.3 Code

```
SOnStemOn = {
  \override Stem.length = #12
  \override Stem.details.beamed-lengths = #'(5.5)
  \override Stem.stencil =
  #(lambda (grob)
     (let* ((x-parent (ly:grob-parent grob X))
            (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
       (if is-rest?
           empty-stencil
           (ly:stencil-combine-at-edge
            (ly:stem::print grob)
            Y
            (- (ly:grob-property grob 'direction))
            (grob-interpret-markup grob
                                   (markup
                                    #:center-align
                                    #:teeny #:sans #:bold "S"))
            -3.5))))
}

```

```
21 SOnStemOff = {
22   \revert Stem.length
23   \revert Stem.details.beamed-lengths
24   \revert Stem.stencil
25   \revert Flag.stencil
26 }
27
28 {
29   \SOnStemOn c'4 g' \SOnStemOff d'' a''
30   \SOnStemOn a'' d'' \SOnStemOff g' c'
31 }
```

Table of Contents

## 10.3 ”V” on Stem



### 10.3.1 Description

This function attaches ”V” to the stem. I have used this to designate a note with a differentiated timbre from others, for example ”brassy tone” for bassoon in my *Gz III* (2019-21) for bass clarinet and bassoon. This function lengthens the stem in order to give a balanced look, especially combined with stems/flags.

### 10.3.2 Grammar

```
\VOnStemOn NOTE ...
\VOnStemOff
```

**NB** `\VOnStemOn` toggles the feature on, while `\VOnStemOff` toggles it off.

### 10.3.3 Code

```
VOnStemOn = {
  \override Stem.no-stem-extend = ##f
  \override Stem.length = #12
  \override Stem.details.beamed-lengths = #'(5.5)
  \override Stem.stencil =
  #(lambda (grob)
     (let* ((x-parent (ly:grob-parent grob X))
            (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
       (if is-rest?
           empty-stencil
           (ly:stencil-combine-at-edge
            (ly:stem::print grob)
            Y
            (- (ly:grob-property grob 'direction))
            (grob-interpret-markup grob
                                   (markup
                                    #:center-align
                                    #:teeny #:sans #:musicglyph "scripts.upbow"))
            -3.5))))
```

```
20  }
21
22  VOnStemOff = {
23    \revert Stem.length
24    \revert Stem.stencil
25    \revert Flag.stencil
26  }
27
28
29  {
30    \VOnStemOn c'4 g' \VOnStemOff d'' a''
31    \VOnStemOn a'' d'' \VOnStemOff g' c'
32  }
```

Table of Contents

# Chapter 11

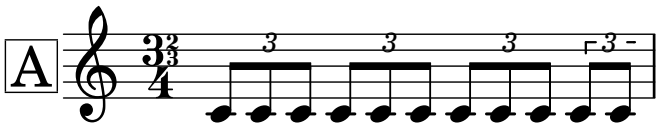
# Time Signatures

First nine entries of this chapter discuss fractional time signatures (variants of the irrational time signatures) and their compound forms. I have been inspired to create these implementations after chancing upon the email exchanges on `lilypond-user` dated from 2014.[1]

While Gould discourages the use of time signatures with numerators as fractions,[2] there are cases where the use of such time signatures seems justified, particularly when the fractions deal with some form of tuplets. This is a form of time signature notation widely seen in works by Chaya Czernowin, Stefan Beyer, myself, and so many others.

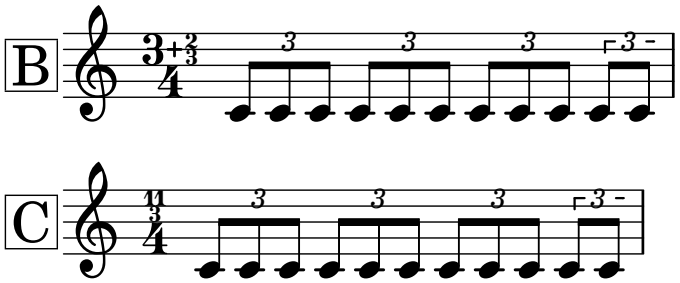
I present the implementation of fractional time signatures in three different styles, A, B, and C. There are implementations for compound meters for each of the styles, in two and three time signatures.

## 11.1 Fractional Time Signatures, Style A




1. See: https://lists.gnu.org/archive/html/lilypond-user/2014-06/msg00209.html. However, in the process of writing this documentation I have come across another email thread on the same mailing list: https://mail.gnu.org/archive/html/lilypond-user/2020-04/msg00423.html

2. Gould, *Behind bars : the definitive guide to music notation*, 180.




### 11.1.1 Description

This particular style of fractional time signatures[3] can be seen in scores by Stefan Beyer, for example *Marsch* (2013-14),[4] *Mittel und Zwecke (Boulevard)* (2014),[5] *Bleib hier. Schau zu. Mach kein Geräusch.* (2017),[6] and *Most of My Clients Come Back* (2012-13).[7] In the case of *Mass und Gewicht* (2021), Beyer uses fractions on the denominator of the time signatures.[8]

Because the size the fractions is a half of the ordinary time signatures, it may be difficult to see from afar.[9]

### 11.1.2 Grammar

```
\fractionalTimeSignatureA
       #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureA
       #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureAPlus
       #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureAPlus
       #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
```

**NB**

3. After having come up with this code, there were other implementations that could be seen on this email thread: https://mail.gnu.org/archive/html/lilypond-user/2020-04/msg00423.html

4. Stefan Beyer, *Marsch* (Manuscript, 2013-14).

5. Stefan Beyer, *Mittel und Zwecke (Boulevard)* (Manuscript, 2014).

6. Stefan Beyer, *Bleib hier. Schau zu. Mach kein Geräusch.* (Manuscript, 2017).

7. Stefan Beyer, *Most of My Clients Come Back* (Manuscript, 2012-13).

8. It would be relatively easy to modify the Scheme code so that the fraction appears next to the denominator of the time signature, instead.

9. It should be noted that in other works such as *Lotte Reiniger's The Sleeping Beauty* (2020-21), Beyer also uses the irrational time signatures as seen in the *Incomplete Tuplet Bracket for Irrational Time Signatures* section of this document.

1. `\fractionalTimeSignatureA` lists time signatures *without* the use of the `+` (plus) sign.
2. `\fractionalTimeSignatureAPlus` lists time signatures with the `+` (plus) sign, when the list with four `NUM`s are given.
3. `NUM1`, `NUM2`, `NUM3`, and `NUM4` can be understood as follows:

$$\frac{1+\frac{2}{3}}{4}$$

   where `NUM1` is optional. The code has `cond` clause, which adjusts the appearance of the time signature according to the length of the list, either having 3 or 4 numbers.
4. `MEASURE_SPAN` denotes how the measure may be written using an *irrational time signature*. In the example snippet, this would be:

$$\frac{3}{4}+\frac{2}{12}=\frac{\mathbf{11}}{\mathbf{12}}$$

5. `BEAT_STRUCT` indicates beat structure, by which the beaming of the measure abides.

### 11.1.3 Code

```
% Inspired by:
% https://lists.gnu.org/archive/html/lilypond-user/2014-06/msg00209.html

\version "2.24.4"
\language "english"

suppressWarning =
#(define-void-function (amount message)(number? string?)
  (for-each
   (lambda (warning)
    (ly:expect-warning message))
   (iota amount 1 1)))

\suppressWarning 3 "strange time signature found"

incompleteTupletBracket =  {
 \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
 \once \override Voice.TupletBracket.bracket-visibility = ##t

```

```
}
incompleteSmallTupletBracket =  {
 \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
 \once \override Voice.TupletBracket.bracket-visibility = ##t
 \once \override Voice.TupletNumber.X-offset =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       2.2
       1.2))

 \once \override Voice.TupletBracket.shorten-pair =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       '(-0.7 . 0.15)
       '(-0.3 . 0.8)))
 \once \override Voice.TupletBracket.X-positions =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       '(1.8 . 3)
       '(0.3 . 2.7)))
}

fractionalTimeSignatureA =
#(define-music-function
  (timeSignatureToShow underlyingMeter beatStructure)
  (list? fraction? number-list?)
  #{
   \time $underlyingMeter
   \set beatStructure = $beatStructure

   \override Staff.TimeSignature.stencil =
   #ly:text-interface::print

   \override Staff.TimeSignature.text =
   #(if (= (length timeSignatureToShow) 4)

        (markup
         #:override
         (cons 'baseline-skip 0)
         (#:center-column
          (#:number
```

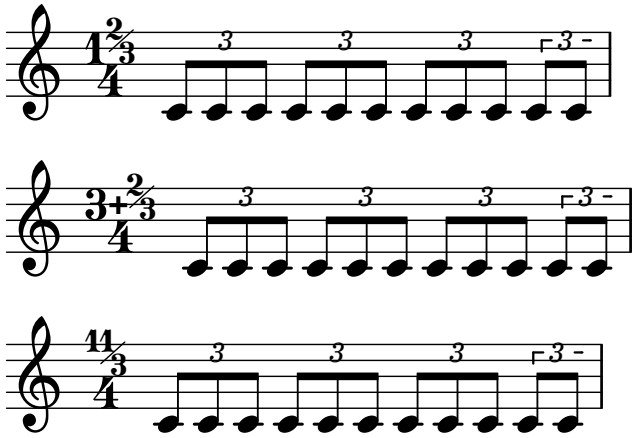
```
61              (#:concat
62               (#:simple
63                (number->string (car timeSignatureToShow))
64                #:halign -1.5
65                (#:center-column
66                 ((#:translate
67                   (cons 0 1)
68                   (#:fontsize -6
69                               (number->string
70                                (cadr timeSignatureToShow))))
71                  (#:translate
72                   (cons 0 0)
73                   (#:fontsize -6
74                               (number->string
75                                (caddr timeSignatureToShow))))))))
76              #:number
77              (number->string (cadddr timeSignatureToShow)))))
78
79           (markup
80            #:override
81            (cons 'baseline-skip 0)
82            (#:center-column
83             (#:number
84              (#:translate
85               (cons 0 1)
86               (#:fontsize -6 (number->string
87                               (car timeSignatureToShow))))
88              #:number
89              (#:translate
90               (cons 0 0)
91               (#:fontsize -6 (number->string
92                               (cadr timeSignatureToShow))))
93              #:number
94              (number->string (caddr timeSignatureToShow)))))
95
96           )
97     #})
98
99   fractionalTimeSignatureAPlus =
100  #(define-music-function
101    (timeSignatureToShow underlyingMeter beatStructure)
```

```
102    (list? fraction? number-list?)
103    #{
104     \time $underlyingMeter
105     \set beatStructure = $beatStructure
106
107     \override Staff.TimeSignature.stencil =
108     #ly:text-interface::print
109
110     \override Staff.TimeSignature.text =
111     #(if (= (length timeSignatureToShow) 4)
112
113          (markup
114           #:override
115           (cons 'baseline-skip 0)
116           (#:center-column
117            (#:number
118             (#:concat
119              (#:simple
120               (number->string (car timeSignatureToShow))
121
122               (#:fontsize -12 (string-append " "))
123               (string-append "+")
124               (#:fontsize -12 (string-append " "))
125
126               #:center-column
127               ((#:translate
128                 (cons 0 1)
129                 (#:fontsize -6
130                             (number->string
131                              (cadr timeSignatureToShow))))
132                (#:translate
133                 (cons 0 0)
134                 (#:fontsize -6
135                             (number->string
136                              (caddr timeSignatureToShow)))))))
137              #:number
138              (number->string (cadddr timeSignatureToShow)))))
139
140          (markup
141           #:override
142           (cons 'baseline-skip 0)
```

```
          (#:center-column
           (#:number
            (#:translate
             (cons 0 1)
             (#:fontsize -6 (number->string
                             (car timeSignatureToShow))))
            #:number
            (#:translate
             (cons 0 0)
             (#:fontsize -6 (number->string
                             (cadr timeSignatureToShow))))
            #:number
            (number->string (caddr timeSignatureToShow)))))

        )
  #})
\new Staff \with { instrumentName = \markup {\fontsize #4 \box "A"}} {
 \fractionalTimeSignatureA #'(3 2 3 4) 11/12 3,3,3,2
 \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
 \tuplet 3/2 {c' c' c'}
 \incompleteTupletBracket \tuplet 3/2 {c' c'}
}
\new Staff \with { instrumentName = \markup {\fontsize #4 \box "B"}} {
 \fractionalTimeSignatureAPlus #'(3 2 3 4) 11/12 3,3,3,2
 \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
 \tuplet 3/2 {c' c' c'}
 \incompleteTupletBracket \tuplet 3/2 {c' c'}
}
\new Staff \with { instrumentName = \markup {\fontsize #4 \box "C"}} {
 \fractionalTimeSignatureA #'(11 3 4) 11/12 3,3,3,2
 \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
 \tuplet 3/2 {c' c' c'}
 \incompleteTupletBracket \tuplet 3/2 {c' c'}
}

```

Table of Contents

## 11.2 Fractional Time Signatures, Style B



### 11.2.1 Description

Style B differs from Style A, as the fraction has a bigger font size. This is similar to the design of fractional time signatures I have used in works such as *Gz II* (2017-22).[10]

### 11.2.2 Grammar

```
\fractionalTimeSignatureB
       #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureB
       #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureBPlus
       #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureBPlus
       #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. `\fractionalTimeSignatureB` lists time signatures *without* the use of the `+` (plus) sign.
2. `\fractionalTimeSignatureBPlus` lists time signatures with the `+` (plus) sign, when the list with four `NUM`s are given.
3. See *Grammar* of Fractional Time Signatures, Style A for the explanation on the arguments.

10. Yoshiaki Onishi, *Gz II : for two accordions* (Brühl and Berlin: Edition Gravis, 2024).

### 11.2.3 Code

```
\version "2.24.4"
\language "english"

% Revised Jan 2 2025 for improving the appearance of fractions

suppressWarning =
#(define-void-function (amount message)(number? string?)
      (for-each
       (lambda (warning)
            (ly:expect-warning message))
       (iota amount 1 1)))

\suppressWarning 3 "strange time signature found"

incompleteTupletBracket =  {
     \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
     \once \override Voice.TupletBracket.bracket-visibility = ##t

}
incompleteSmallTupletBracket =  {
     \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
     \once \override Voice.TupletBracket.bracket-visibility = ##t
     \once \override Voice.TupletNumber.X-offset =
     #(lambda (grob)
           (if (= UP (ly:grob-property grob 'direction))
               2.2
               1.2))

     \once \override Voice.TupletBracket.shorten-pair =
     #(lambda (grob)
           (if (= UP (ly:grob-property grob 'direction))
               '(-0.7 . 0.15)
               '(-0.3 . 0.8)))
     \once \override Voice.TupletBracket.X-positions =
     #(lambda (grob)
           (if (= UP (ly:grob-property grob 'direction))
               '(1.8 . 3)
               '(0.3 . 2.7)))
}
```

```
40
41  fractionalTimeSignatureB =
42  #(define-music-function
43    (timeSignatureToShow underlyingMeter beatStructure)
44    (list? fraction? number-list?)
45    #{
46
47          \time $underlyingMeter
48          \set beatStructure = $beatStructure
49
50          \override Staff.TimeSignature.stencil =
51          #ly:text-interface::print
52          \override Staff.TimeSignature.text =
53
54          #(if (= (length timeSignatureToShow) 4)
55
56               (markup
57                (make-override-markup
58                 (cons 'baseline-skip 0)
59                 (make-center-column-markup
60                  (list
61                   (make-line-markup
62                    (list
63                     (make-number-markup
64                      (number->string (car timeSignatureToShow)))
65
66
67                     (make-hspace-markup -0.5)
68                     (make-right-align-markup
69                      (make-number-markup
70                       (make-translate-markup
71                        (cons 0 1.5)
72                        (make-fontsize-markup
73                         -3
74                         (number->string (cadr timeSignatureToShow))))))
75
76                     (make-hspace-markup -1.5)
77
78                     (make-override-markup
79                      (cons 'alignment 0)
80                      (make-translate-markup
```

```
81                      (cons 0 0.8)
82                      (make-draw-line-markup (cons 1.5 1.35))))
83
84                    (make-hspace-markup -1.5)
85
86                    (make-number-markup
87                     (make-left-align-markup
88                      (make-fontsize-markup
89                       -3
90                       (number->string (caddr timeSignatureToShow)))))))
91
92                  (make-number-markup
93                   (number->string (cadddr timeSignatureToShow))))))
94
95
96                (markup
97                 (make-override-markup
98                  (cons 'baseline-skip 0)
99                  (make-center-column-markup
100                  (list
101                   (make-line-markup
102                    (list
103                     (make-number-markup
104                      (make-right-align-markup
105                       (make-translate-markup
106                        (cons 0 1.5)
107                        (make-fontsize-markup
108                         -3
109                         (number->string (car timeSignatureToShow))))))
110
111                     (make-hspace-markup -1.5)
112
113                     (make-override-markup
114                      (cons 'alignment 0)
115                      (make-translate-markup
116                       (cons 0 0.8)
117                       (make-draw-line-markup (cons 1.5 1.35))))
118
119                     (make-hspace-markup -1.5)
120
121                     (make-translate-markup
```

```
122                     (cons 0 0)
123                     (make-fontsize-markup
124                      -3
125                      (make-number-markup
126                       (number->string (cadr timeSignatureToShow)))))))
127
128                 (make-number-markup
129                  (number->string (caddr timeSignatureToShow)))))))
130
131             )
132   #})
133
134 fractionalTimeSignatureBPlus =
135 #(define-music-function
136   (timeSignatureToShow underlyingMeter beatStructure)
137   (list? fraction? number-list?)
138   #{
139
140         \time $underlyingMeter
141         \set beatStructure = $beatStructure
142
143         \override Staff.TimeSignature.stencil =
144         #ly:text-interface::print
145         \override Staff.TimeSignature.text =
146
147         #(if (= (length timeSignatureToShow) 4)
148
149
150              (markup
151               (make-override-markup
152                (cons 'baseline-skip 0)
153                (make-center-column-markup
154                 (list
155                  (make-line-markup
156                   (list
157                    (make-number-markup
158                     (number->string (car timeSignatureToShow)))
159                    (make-fontsize-markup
160                     -12
161                     (make-simple-markup " "))
162
```

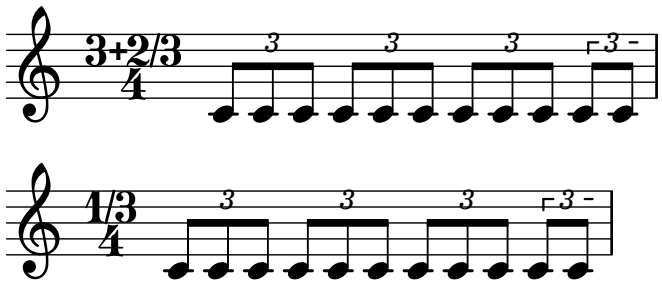
```
163
164                     (make-hspace-markup -1.25)
165                     (make-translate-markup
166                      (cons 0 0.4)
167                      (make-bold-markup
168                       (make-simple-markup "+")))
169
170                     (make-hspace-markup -0.25)
171
172                     (make-hspace-markup -0.5)
173                     (make-right-align-markup
174                      (make-number-markup
175                       (make-translate-markup
176                        (cons 0 1.5)
177                        (make-fontsize-markup
178                         -3
179                         (number->string (cadr timeSignatureToShow))))))
180
181
182
183
184                     (make-hspace-markup -1.5)
185
186                     (make-override-markup
187                      (cons 'alignment 0)
188                      (make-translate-markup
189                       (cons 0 0.8)
190                       (make-draw-line-markup (cons 1.5 1.35))))
191
192                     (make-hspace-markup -1.5)
193
194                     (make-number-markup
195                      (make-left-align-markup
196                       (make-fontsize-markup
197                        -3
198                        (number->string (caddr timeSignatureToShow)))))))
199
200                   (make-number-markup
201                    (number->string (cadddr timeSignatureToShow)))))))
202
203
```

```
204               (markup
205                (make-override-markup
206                 (cons 'baseline-skip 0)
207                 (make-center-column-markup
208                  (list
209                   (make-line-markup
210                    (list
211                     (make-number-markup
212                      (make-right-align-markup
213                       (make-translate-markup
214                        (cons 0 1.6)
215                        (make-fontsize-markup
216                         -3
217                         (number->string (car timeSignatureToShow))))))
218
219                     (make-hspace-markup -1.5)
220
221                     (make-override-markup
222                      (cons 'alignment 0)
223                      (make-translate-markup
224                       (cons 0 0.8)
225                       (make-draw-line-markup (cons 1.5 1.35))))
226
227                     (make-hspace-markup -1.5)
228
229                     (make-translate-markup
230                      (cons 0 0)
231                      (make-fontsize-markup
232                       -3
233                       (make-number-markup
234                        (number->string (cadr timeSignatureToShow)))))))
235
236                   (make-number-markup
237                    (number->string (caddr timeSignatureToShow))))))
238               )
239    #})
240
241
242  {
243   \fractionalTimeSignatureB #'(1 2 3 4) 11/12 3,3,3,2
244   \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
```

```
 \tuplet 3/2 {c' c' c'}
 \incompleteTupletBracket \tuplet 3/2 {c' c'}
}

{
 \fractionalTimeSignatureBPlus #'(3 2 3 4) 11/12 3,3,3,2
 \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
 \tuplet 3/2 {c' c' c'}
 \incompleteTupletBracket \tuplet 3/2 {c' c'}
}

{
 \fractionalTimeSignatureB #'(11 3 4) 11/12 3,3,3,2
 \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
 \tuplet 3/2 {c' c' c'}
 \incompleteTupletBracket \tuplet 3/2 {c' c'}
}
```

**Table of Contents**

## 11.3 Fractional Time Signatures, Style C




### 11.3.1 Description

Style C of the fractional time signatures offers the largest font size for displaying the fractions. This design is commonly seen in scores by Chaya Czernowin, in such works as String Quartet (1995),[11] *Lovesong* (2010),[12] *Streams (Slow Summer Stay I)* (2012),[13] and *At the fringe of our gaze* (2012/13).[14]

### 11.3.2 Grammar

```
\fractionalTimeSignatureC
       #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureC
       #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. By default, `\fractionalTimeSignatureC` shows `+` (plus) sign when four `NUM`s are given. As the font size for the ordinary numerator and the fractions is the same, without `+` it becomes very confusing to read the time signature. Thus, contrary to Styles A and B, there is no separate function for the time signature with the `+` sign given.
2. See *Grammar* of Fractional Time Signatures, Style A for the explanation on the arguments.

11. Chaya Czernowin, *String Quartet* (Schott, 1995).
12. Chaya Czernowin, *Lovesong : for mixed ensemble* (Schott, 2010).
13. Chaya Czernowin, *Streams (Slow Summer Stay I) : for 8 players* (Schott, 2012).
14. Chaya Czernowin, *At the fringe of our gaze : for Orchestra and Concertino Group* (Schott, 2012/13).

### 11.3.3 Code

```
\version "2.24.4"
\language "english"

% Revised Jan 2 2025 for improving the appearance of fractions

suppressWarning =
#(define-void-function (amount message)(number? string?)
      (for-each
       (lambda (warning)
            (ly:expect-warning message))
       (iota amount 1 1)))

\suppressWarning 2 "strange time signature found"

incompleteTupletBracket =  {
     \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
     \once \override Voice.TupletBracket.bracket-visibility = ##t

}
incompleteSmallTupletBracket =  {
     \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
     \once \override Voice.TupletBracket.bracket-visibility = ##t
     \once \override Voice.TupletNumber.X-offset =
     #(lambda (grob)
           (if (= UP (ly:grob-property grob 'direction))
               2.2
               1.2))

     \once \override Voice.TupletBracket.shorten-pair =
     #(lambda (grob)
           (if (= UP (ly:grob-property grob 'direction))
               '(-0.7 . 0.15)
               '(-0.3 . 0.8)))
     \once \override Voice.TupletBracket.X-positions =
     #(lambda (grob)
           (if (= UP (ly:grob-property grob 'direction))
               '(1.8 . 3)
               '(0.3 . 2.7)))
}
```

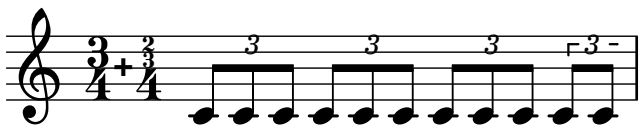
```
40
41  fractionalTimeSignatureC =
42  #(define-music-function
43    (timeSignatureToShow underlyingMeter beatStructure)
44    (list? fraction? number-list?)
45    #{
46
47          \time $underlyingMeter
48          \set beatStructure = $beatStructure
49
50          \override Staff.TimeSignature.stencil =
51          #ly:text-interface::print
52          \override Staff.TimeSignature.text =
53
54          #(if (= (length timeSignatureToShow) 4)
55
56               (markup
57                (make-override-markup
58                 (cons 'baseline-skip 0)
59                 (make-center-column-markup
60                  (list
61                   (make-line-markup
62                    (list
63                     (make-number-markup
64                      (number->string
65                       (car timeSignatureToShow)))
66                     (make-fontsize-markup
67                      -12
68                      (make-simple-markup " "))
69
70
71                     (make-hspace-markup -1.25)
72                     (make-translate-markup
73                      (cons 0 0.4)
74                      (make-bold-markup
75                       (make-simple-markup "+")))
76
77                     (make-hspace-markup -0.25)
78
79                     (make-hspace-markup -0.5)
80                     (make-right-align-markup
```

```
                    (make-number-markup
                     (number->string
                      (cadr timeSignatureToShow))))

                   (make-hspace-markup -0.6)

                   (make-override-markup
                    (list (cons 'alignment 0)
                          (cons 'thickness 2))
                    (make-draw-line-markup (cons 0.5 2)))

                   (make-hspace-markup -0.6)

                   (make-number-markup
                    (make-left-align-markup
                     (number->string
                      (caddr timeSignatureToShow))))))

                (make-number-markup
                 (number->string
                  (cadddr timeSignatureToShow)))))))


              (markup
               (make-override-markup
                (cons 'baseline-skip 0)
                (make-center-column-markup
                 (list
                  (make-line-markup
                   (list

                    (make-right-align-markup
                     (make-number-markup
                      (number->string
                       (car timeSignatureToShow))))

                    (make-hspace-markup -0.6)

                    (make-override-markup
                     (list (cons 'alignment 0)
                           (cons 'thickness 2))
```

```
122                      (make-draw-line-markup
123                       (cons 0.5 2)))
124
125                     (make-hspace-markup -0.6)
126
127                     (make-number-markup
128                      (make-left-align-markup
129                       (number->string
130                        (cadr timeSignatureToShow))))))
131
132                   (make-number-markup
133                    (number->string
134                     (caddr timeSignatureToShow))))))
135                ))
136    #})
137
138
139  {
140       \fractionalTimeSignatureC #'(3 2 3 4) 11/12 3,3,3,2
141       \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
142       \tuplet 3/2 {c' c' c'}
143       \incompleteTupletBracket \tuplet 3/2 {c' c'}
144  }
145
146
147  {
148       \fractionalTimeSignatureC #'(1 3 4) 11/12 3,3,3,2
149       \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
150       \tuplet 3/2 {c' c' c'}
151       \incompleteTupletBracket \tuplet 3/2 {c' c'}
152  }
```

**Table of Contents**

## 11.4 Compound Meter with Two Fractional Time Signatures, Style A



### 11.4.1 Description

This is an implementation of a compound meter with two fractional time signatures with Style A.

### 11.4.2 Grammar

```
\compoundFractionalTimeSignatureATwo
       #'((TIME_SIG1)(TIME_SIG2)) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. Following the convention of `\compoundMeter` to enter the two time signatures, you will create a list of lists. Each `TIME_SIG` accepts:
   - an ordinary time signature (list with two numbers);
   - a time signature with a fraction (list with three numbers), or;
   - a time signature with an ordinary numerator and a fraction.

   See *Grammar* of Fractional Time Signatures, Style A for the explanation on the arguments for the order of arguments to specify time signatures.
2. `MEASURE_SPAN` and `BEAT_STRUCT` follow the same convention as before.

### 11.4.3 Code

```
\version "2.24.4"
\language "english"

suppressWarning =
#(define-void-function (amount message)(number? string?)
  (for-each
   (lambda (warning)
    (ly:expect-warning message))
```

```
    (iota amount 1 1)))

\suppressWarning 1 "strange time signature found"

incompleteTupletBracket =  {
 \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
 \once \override Voice.TupletBracket.bracket-visibility = ##t

}
incompleteSmallTupletBracket =  {
 \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
 \once \override Voice.TupletBracket.bracket-visibility = ##t
 \once \override Voice.TupletNumber.X-offset =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       2.2
       1.2))

 \once \override Voice.TupletBracket.shorten-pair =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       '(-0.7 . 0.15)
       '(-0.3 . 0.8)))
 \once \override Voice.TupletBracket.X-positions =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       '(1.8 . 3)
       '(0.3 . 2.7)))
}

compoundFractionalTimeSignatureATwo =
#(define-music-function
  (timeSignatureToShow underlyingMeter beatStructure)
  (list? fraction? number-list?)
  (define mkup
   (markup
    #:concat
    (
     #:override
     (cons 'baseline-skip 0)
     (cond ((= (length (car timeSignatureToShow)) 2)
```

```
50              (make-center-column-markup
51               (list (make-number-markup
52                      (number->string
53                       (car (car timeSignatureToShow))))
54                     (make-number-markup
55                      (number->string
56                       (cadr (car timeSignatureToShow)))))))
57
58             ((= (length (car timeSignatureToShow)) 3)
59              (make-override-markup
60               (cons 'baseline-skip 0)
61               (make-center-column-markup
62                (list
63
64
65                 (make-center-column-markup
66                  (list
67                   (make-translate-markup
68                    (cons 0 1)
69                    (make-fontsize-markup
70                     -6
71                     (make-number-markup
72                      (number->string
73                       (car (car timeSignatureToShow))))))
74                   (make-translate-markup
75                    (cons 0 0)
76                    (make-fontsize-markup
77                     -6
78                     (make-number-markup
79                      (number->string
80                       (cadr (car timeSignatureToShow)))))))))
81                 (make-number-markup
82                  (number->string
83                   (caddr (car timeSignatureToShow))))))
84                ))
85
86
87             ((= (length (car timeSignatureToShow)) 4)
88
89              (make-override-markup
90               (cons 'baseline-skip 0)
```

```
91              (make-center-column-markup
92               (list
93
94                (make-concat-markup
95                 (list (make-number-markup
96                        (number->string
97                         (car (car timeSignatureToShow))))
98                       (make-halign-markup
99                        -1.5
100                       (make-center-column-markup
101                        (list
102                         (make-translate-markup
103                          (cons 0 1)
104                          (make-fontsize-markup
105                           -6
106                           (make-number-markup
107                            (number->string
108                             (cadr (car timeSignatureToShow))))))
109                         (make-translate-markup
110                          (cons 0 0)
111                          (make-fontsize-markup
112                           -6
113                           (make-number-markup
114                            (number->string
115                             (caddr (car timeSignatureToShow))))))))))
116               (make-number-markup
117                (number->string
118                 (cadddr (car timeSignatureToShow))))))
119             ))
120           )
121
122
123     #:translate
124     (cons 0 -0.5)
125     (#:fontsize -12 " ")
126     #:translate
127     (cons 0 -0.5)
128     (#:bold "+")
129     #:translate
130     (cons 0 -0.5)
131     (#:fontsize -12 " ")
```

```
132
133       #:override
134       (cons 'baseline-skip 0)
135       (cond ((= (length (cadr timeSignatureToShow)) 2)
136              (make-center-column-markup
137               (list (make-number-markup
138                      (number->string
139                       (car (cadr timeSignatureToShow))))
140                     (make-number-markup
141                      (number->string
142                       (cadr (cadr timeSignatureToShow)))))))
143
144             ((= (length (cadr timeSignatureToShow)) 3)
145
146              (make-override-markup
147               (cons 'baseline-skip 0)
148               (make-center-column-markup
149                (list
150
151
152                  (make-center-column-markup
153                   (list
154                    (make-translate-markup
155                     (cons 0 1)
156                     (make-fontsize-markup
157                      -6
158                      (make-number-markup
159                       (number->string
160                        (car (cadr timeSignatureToShow))))))
161                    (make-translate-markup
162                     (cons 0 0)
163                     (make-fontsize-markup
164                      -6
165                      (make-number-markup
166                       (number->string
167                        (cadr (cadr timeSignatureToShow)))))))))
168                  (make-number-markup
169                   (number->string
170                    (caddr (cadr timeSignatureToShow))))))
171                ))
172
```

```
173
174             ((= (length (cadr timeSignatureToShow)) 4)
175
176              (make-override-markup
177               (cons 'baseline-skip 0)
178               (make-center-column-markup
179                (list
180
181                 (make-concat-markup
182                  (list (make-number-markup
183                         (number->string
184                          (car (cadr timeSignatureToShow))))
185                        (make-halign-markup
186                         -1.5
187                         (make-center-column-markup
188                          (list
189                           (make-translate-markup
190                            (cons 0 1)
191                            (make-fontsize-markup
192                             -6
193                             (make-number-markup
194                              (number->string
195                               (cadr (cadr timeSignatureToShow))))))
196                           (make-translate-markup
197                            (cons 0 0)
198                            (make-fontsize-markup
199                             -6
200                             (make-number-markup
201                              (number->string
202                               (caddr (cadr timeSignatureToShow))))))))))
203                 (make-number-markup
204                  (number->string
205                   (cadddr (cadr timeSignatureToShow))))))
206               ))
207              )
208       )))
209
210     #{
211      \time $underlyingMeter
212      \set beatStructure = $beatStructure
213
```

```
    \override Timing.TimeSignature.stencil =
    #ly:text-interface::print
    \override Timing.TimeSignature.text =
    #mkup
   #})


{
 \compoundFractionalTimeSignatureATwo #'((3 4)(2 3 4)) 11/12 3,3,3,2
 \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
 \tuplet 3/2 {c' c' c'}
 \incompleteTupletBracket \tuplet 3/2 {c' c'}
}
```

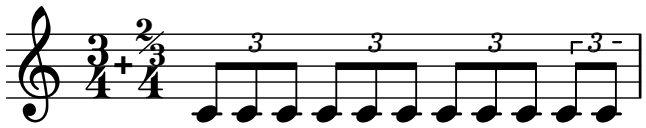
### 11.4.4 Discussion

1. This was a tricky one to make, as I had to resort to building the Scheme code without using the syntactic sugars, i.e. `#:`.[15] If any modification are to be made to this code, it is recommended to carefully examine where the corresponding parenthesis of a starting parenthesis is located. It is also helpful to watch LilyPond Log for any errors, as it seems to give hints for how many argument(s) a function is looking for.
2. I am hoping to find ways to simplify the code, as the same bits (with variations in variables that are called upon) of the codes are used to streamline the formatting of the time signature appearances.

**Table of Contents**

15. See *Known issues and warnings* at: https://lilypond.org/doc/v2.24/Documentation/extending/markup-construction-in-scheme

## 11.5 Compound Meter with Two Fractional Time Signatures, Style B



### 11.5.1 Description

This is an implementation of a compound meter with two fractional time signatures with Style B.

### 11.5.2 Grammar

```
\compoundFractionalTimeSignatureBTwo
       #'((TIME_SIG1)(TIME_SIG2)) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. Following the convention of `\compoundMeter` to enter the two time signatures, you will create a list of lists. Each `TIME_SIG` accepts: an ordinary time signature (list with two numbers), a time signature with a fraction (list with three numbers), or a time signature with an ordinary numerator and a fraction. See *Grammar* of Fractional Time Signatures, Style A for the explanation on the arguments for the order of arguments to specify time signatures.
2. `MEASURE_SPAN` and `BEAT_STRUCT` follow the same convention as before.

### 11.5.3 Code

```
\version "2.24.4"
\language "english"

% Revised Jan 2 2025 for improving the appearance of fractions

suppressWarning =
#(define-void-function (amount message)(number? string?)
       (for-each
        (lambda (warning)
             (ly:expect-warning message))
        (iota amount 1 1)))
```

```
12
13 \suppressWarning 1 "strange time signature found"
14
15 incompleteTupletBracket =  {
16      \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
17      \once \override Voice.TupletBracket.bracket-visibility = ##t
18
19 }
20 incompleteSmallTupletBracket =  {
21      \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
22      \once \override Voice.TupletBracket.bracket-visibility = ##t
23      \once \override Voice.TupletNumber.X-offset =
24      #(lambda (grob)
25            (if (= UP (ly:grob-property grob 'direction))
26                2.2
27                1.2))
28
29      \once \override Voice.TupletBracket.shorten-pair =
30      #(lambda (grob)
31            (if (= UP (ly:grob-property grob 'direction))
32                '(-0.7 . 0.15)
33                '(-0.3 . 0.8)))
34      \once \override Voice.TupletBracket.X-positions =
35      #(lambda (grob)
36            (if (= UP (ly:grob-property grob 'direction))
37                '(1.8 . 3)
38                '(0.3 . 2.7)))
39 }
40
41 compoundFractionalTimeSignatureBTwo =
42 #(define-music-function
43   (timeSignatureToShow underlyingMeter beatStructure)
44   (list? fraction? number-list?)
45   (define mkup
46        (markup
47         #:concat
48         (
49
50              #:override
51              (cons 'baseline-skip 0)
52              (cond ((= (length (car timeSignatureToShow)) 2)
```

```
53                      (make-center-column-markup
54                       (list (make-number-markup
55                              (number->string
56                               (car (car timeSignatureToShow))))
57                             (make-number-markup
58                              (number->string
59                               (cadr (car timeSignatureToShow)))))))
60
61                     ((= (length (car timeSignatureToShow)) 3)
62                      (make-override-markup
63                       (cons 'baseline-skip 0)
64                       (make-center-column-markup
65                        (list
66                         (make-line-markup
67                          (list
68                           (make-number-markup
69                            (make-right-align-markup
70                             (make-translate-markup
71                              (cons 0 1.6)
72                              (make-fontsize-markup
73                               -3
74                               (number->string
75                                (car (car timeSignatureToShow)))))))
76
77                           (make-hspace-markup -1.5)
78
79                           (make-override-markup
80                            (cons 'alignment 0)
81                            (make-translate-markup
82                             (cons 0 0.8)
83                             (make-draw-line-markup (cons 1.5 1.35))))
84
85                           (make-hspace-markup -1.5)
86
87                           (make-translate-markup
88                            (cons 0 0)
89                            (make-fontsize-markup
90                             -3
91                             (make-number-markup
92                              (number->string
93                               (cadr (car timeSignatureToShow))))))))
```

```
94
95                          (make-number-markup
96                           (number->string
97                            (caddr (car timeSignatureToShow))))))))
98
99
100                       ((= (length (car timeSignatureToShow)) 4)
101
102                        (make-override-markup
103                         (cons 'baseline-skip 0)
104                         (make-center-column-markup
105                          (list
106                           (make-line-markup
107                            (list
108                             (make-number-markup
109                              (number->string
110                               (car (car timeSignatureToShow))))
111                             (make-fontsize-markup
112                              -12
113                              (make-simple-markup " "))
114
115                             (make-hspace-markup -1.25)
116                             (make-translate-markup
117                              (cons 0 0.4)
118                              (make-bold-markup
119                               (make-simple-markup "+")))
120
121                             (make-hspace-markup -0.25)
122
123                             (make-hspace-markup -0.5)
124                             (make-right-align-markup
125                              (make-number-markup
126                               (make-translate-markup
127                                (cons 0 1.5)
128                                (make-fontsize-markup
129                                 -3
130                                 (number->string
131                                  (cadr (car timeSignatureToShow)))))))
132
133                             (make-hspace-markup -1.5)
134
```

```
                        (make-override-markup
                         (cons 'alignment 0)
                         (make-translate-markup
                          (cons 0 0.8)
                          (make-draw-line-markup
                           (cons 1.5 1.35))))

                        (make-hspace-markup -1.5)

                        (make-number-markup
                         (make-left-align-markup
                          (make-fontsize-markup
                           -3
                           (number->string
                            (caddr (car timeSignatureToShow))))))))

                       (make-number-markup
                        (number->string
                         (cadddr (car timeSignatureToShow)))))))))


                #:translate
                (cons 0 -0.5)
                (#:fontsize -12 " ")
                #:translate
                (cons 0 -0.5)
                (#:bold "+")
                #:translate
                (cons 0 -0.5)
                (#:fontsize -12 " ")

                #:override
                (cons 'baseline-skip 0)

                (cond ((= (length (cadr timeSignatureToShow)) 2)
                       (make-center-column-markup
                        (list (make-number-markup
                               (number->string
                                (car (cadr timeSignatureToShow))))
                              (make-number-markup
                               (number->string
```

```
                    (cadr (cadr timeSignatureToShow)))))))

                ((= (length (cadr timeSignatureToShow)) 3)
                 (make-override-markup
                  (cons 'baseline-skip 0)
                  (make-center-column-markup
                   (list
                    (make-line-markup
                     (list
                      (make-number-markup
                       (make-right-align-markup
                        (make-translate-markup
                         (cons 0 1.6)
                         (make-fontsize-markup
                          -3
                          (number->string
                           (car (cadr timeSignatureToShow)))))))

                      (make-hspace-markup -1.5)

                      (make-override-markup
                       (cons 'alignment 0)
                       (make-translate-markup
                        (cons 0 0.8)
                        (make-draw-line-markup (cons 1.5 1.35))))

                      (make-hspace-markup -1.5)

                      (make-translate-markup
                       (cons 0 0)
                       (make-fontsize-markup
                        -3
                        (make-number-markup
                         (number->string
                          (cadr (cadr timeSignatureToShow))))))))

                    (make-number-markup
                     (number->string
                      (caddr (cadr timeSignatureToShow))))))))
```

```
                ((= (length (cadr timeSignatureToShow)) 4)

                 (make-override-markup
                  (cons 'baseline-skip 0)
                  (make-center-column-markup
                   (list
                    (make-line-markup
                     (list
                      (make-number-markup
                       (number->string
                        (car (cadr timeSignatureToShow))))
                      (make-fontsize-markup
                       -12
                       (make-simple-markup " "))

                      (make-hspace-markup -1.25)
                      (make-translate-markup
                       (cons 0 0.4)
                       (make-bold-markup
                        (make-simple-markup "+")))

                      (make-hspace-markup -0.25)

                      (make-hspace-markup -0.5)
                      (make-right-align-markup
                       (make-number-markup
                        (make-translate-markup
                         (cons 0 1.5)
                         (make-fontsize-markup
                          -3
                          (number->string
                           (cadr (cadr timeSignatureToShow)))))))

                      (make-hspace-markup -1.5)

                      (make-override-markup
                       (cons 'alignment 0)
                       (make-translate-markup
                        (cons 0 0.8)
                        (make-draw-line-markup
                         (cons 1.5 1.35))))
```

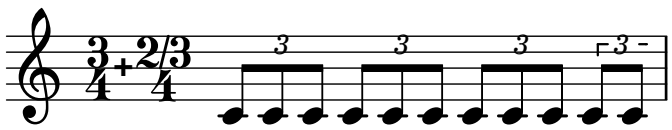
```
258
259                             (make-hspace-markup -1.5)
260
261                             (make-number-markup
262                              (make-left-align-markup
263                               (make-fontsize-markup
264                                -3
265                                (number->string
266                                 (caddr (cadr timeSignatureToShow))))))))
267
268                           (make-number-markup
269                            (number->string
270                             (cadddr (cadr timeSignatureToShow)))))))))
271                 )))
272
273     #{
274           \time $underlyingMeter
275           \set beatStructure = $beatStructure
276
277           \override Timing.TimeSignature.stencil =
278           #ly:text-interface::print
279           \override Timing.TimeSignature.text =
280           #mkup
281     #})
282
283
284   {
285
286         \compoundFractionalTimeSignatureBTwo #'((3 4)(2 3 4)) 11/12 3,3,3,2
287         \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
288         \tuplet 3/2 {c' c' c'}
289         \incompleteTupletBracket \tuplet 3/2 {c' c'}
290   }
```

### 11.5.4 Discussion

See **Discussion** of the entry *Compound Meter with Two Fractional Time Signatures, Style A.*

**Table of Contents**

## 11.6 Compound Meter with Two Fractional Time Signatures, Style C




### 11.6.1 Description

This is an implementation of a compound meter with two fractional time signatures with Style C.

### 11.6.2 Grammar

```
\compoundFractionalTimeSignatureCTwo
       #'((TIME_SIG1)(TIME_SIG2)) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. Following the convention of `\compoundMeter` to enter the two time signatures, you will create a list of lists. Each `TIME_SIG` accepts: an ordinary time signature (list with two numbers), a time signature with a fraction (list with three numbers), or a time signature with an ordinary numerator and a fraction. See *Grammar* of Fractional Time Signatures, Style A for the explanation on the arguments for the order of arguments to specify time signatures.
2. `MEASURE_SPAN` and `BEAT_STRUCT` follow the same convention as before.

### 11.6.3 Code

```
\version "2.24.4"
\language "english"

% Revised Jan 2 2025 for improving the appearance of fractions

suppressWarning =
#(define-void-function (amount message)(number? string?)
       (for-each
        (lambda (warning)
             (ly:expect-warning message))
        (iota amount 1 1)))
```

```
12
13 \suppressWarning 1 "strange time signature found"
14
15 incompleteTupletBracket =  {
16      \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
17      \once \override Voice.TupletBracket.bracket-visibility = ##t
18
19 }
20 incompleteSmallTupletBracket =  {
21      \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
22      \once \override Voice.TupletBracket.bracket-visibility = ##t
23      \once \override Voice.TupletNumber.X-offset =
24      #(lambda (grob)
25            (if (= UP (ly:grob-property grob 'direction))
26                2.2
27                1.2))
28
29      \once \override Voice.TupletBracket.shorten-pair =
30      #(lambda (grob)
31            (if (= UP (ly:grob-property grob 'direction))
32                '(-0.7 . 0.15)
33                '(-0.3 . 0.8)))
34      \once \override Voice.TupletBracket.X-positions =
35      #(lambda (grob)
36            (if (= UP (ly:grob-property grob 'direction))
37                '(1.8 . 3)
38                '(0.3 . 2.7)))
39 }
40
41 compoundFractionalTimeSignatureCTwo =
42 #(define-music-function
43   (timeSignatureToShow underlyingMeter beatStructure)
44   (list? fraction? number-list?)
45   (define mkup
46        (markup
47         #:concat
48         (
49
50              #:override
51              (cons 'baseline-skip 0)
52              (cond ((= (length (car timeSignatureToShow)) 2)
```

```
53                      (make-center-column-markup
54                       (list (make-number-markup
55                              (number->string
56                               (car (car timeSignatureToShow))))
57                             (make-number-markup
58                              (number->string
59                               (cadr (car timeSignatureToShow)))))))
60
61                     ((= (length (car timeSignatureToShow)) 3)
62                      (make-override-markup
63                       (cons 'baseline-skip 0)
64                       (make-center-column-markup
65                        (list
66                         (make-line-markup
67                          (list
68
69                           (make-right-align-markup
70                            (make-number-markup
71                             (number->string
72                              (car (car timeSignatureToShow)))))
73
74                           (make-hspace-markup -0.6)
75
76                           (make-override-markup
77                            (list (cons 'alignment 0)
78                                  (cons 'thickness 2))
79                            (make-draw-line-markup
80                             (cons 0.5 2)))
81
82                           (make-hspace-markup -0.6)
83
84                           (make-number-markup
85                            (make-left-align-markup
86                             (number->string
87                              (cadr (car timeSignatureToShow)))))))
88
89                         (make-number-markup
90                          (number->string
91                           (caddr (car timeSignatureToShow))))))))
92
93
```

```
94                    ((= (length (car timeSignatureToShow)) 4)
95
96                     (make-override-markup
97                      (cons 'baseline-skip 0)
98                      (make-center-column-markup
99                       (list
100                       (make-line-markup
101                        (list
102                         (make-number-markup
103                          (number->string
104                           (car (car timeSignatureToShow))))
105                         (make-fontsize-markup
106                          -12
107                          (make-simple-markup " "))
108
109
110                         (make-hspace-markup -1.25)
111                         (make-translate-markup
112                          (cons 0 0.4)
113                          (make-bold-markup
114                           (make-simple-markup "+")))
115
116                         (make-hspace-markup -0.25)
117
118                         (make-hspace-markup -0.5)
119                         (make-right-align-markup
120                          (make-number-markup
121                           (number->string
122                            (cadr (car timeSignatureToShow)))))
123
124                         (make-hspace-markup -0.6)
125
126                         (make-override-markup
127                          (list (cons 'alignment 0)
128                                (cons 'thickness 2))
129                          (make-draw-line-markup (cons 0.5 2)))
130
131                         (make-hspace-markup -0.6)
132
133                         (make-number-markup
134                          (make-left-align-markup
```

```
                        (number->string
                         (caddr (car timeSignatureToShow)))))))

                    (make-number-markup
                     (number->string
                      (cadddr (car timeSignatureToShow)))))))))


              #:translate
              (cons 0 -0.5)
              (#:fontsize -12 " ")
              #:translate
              (cons 0 -0.5)
              (#:bold "+")
              #:translate
              (cons 0 -0.5)
              (#:fontsize -12 " ")

              #:override
              (cons 'baseline-skip 0)

              (cond ((= (length (cadr timeSignatureToShow)) 2)
                     (make-center-column-markup
                      (list (make-number-markup
                             (number->string
                              (car (cadr timeSignatureToShow))))
                            (make-number-markup
                             (number->string
                              (cadr (cadr timeSignatureToShow)))))))

                    ((= (length (cadr timeSignatureToShow)) 3)
                     (make-override-markup
                      (cons 'baseline-skip 0)
                      (make-center-column-markup
                       (list
                        (make-line-markup
                         (list

                          (make-right-align-markup
                           (make-number-markup
                            (number->string
```

```
176                          (car (cadr timeSignatureToShow)))))
177
178                        (make-hspace-markup -0.6)
179
180                        (make-override-markup
181                         (list (cons 'alignment 0)
182                               (cons 'thickness 2))
183                         (make-draw-line-markup
184                          (cons 0.5 2)))
185
186                        (make-hspace-markup -0.6)
187
188                        (make-number-markup
189                         (make-left-align-markup
190                          (number->string
191                           (cadr (cadr timeSignatureToShow)))))))
192
193                      (make-number-markup
194                       (number->string
195                        (caddr (cadr timeSignatureToShow))))))))
196
197
198                    ((= (length (cadr timeSignatureToShow)) 4)
199
200                     (make-override-markup
201                      (cons 'baseline-skip 0)
202                      (make-center-column-markup
203                       (list
204                        (make-line-markup
205                         (list
206                          (make-number-markup
207                           (number->string
208                            (car (cadr timeSignatureToShow))))
209                          (make-fontsize-markup
210                           -12
211                           (make-simple-markup " "))
212
213
214                          (make-hspace-markup -1.25)
215                          (make-translate-markup
216                           (cons 0 0.4)
```

```
                              (make-bold-markup
                               (make-simple-markup "+")))

                             (make-hspace-markup -0.25)

                             (make-hspace-markup -0.5)
                             (make-right-align-markup
                              (make-number-markup
                               (number->string
                                (cadr (cadr timeSignatureToShow)))))

                             (make-hspace-markup -0.6)

                             (make-override-markup
                              (list (cons 'alignment 0)
                                    (cons 'thickness 2))
                              (make-draw-line-markup (cons 0.5 2)))

                             (make-hspace-markup -0.6)

                             (make-number-markup
                              (make-left-align-markup
                               (number->string
                                (caddr (cadr timeSignatureToShow)))))))

                            (make-number-markup
                             (number->string
                              (cadddr (cadr timeSignatureToShow)))))))))
                )))

    #{
          \time $underlyingMeter
          \set beatStructure = $beatStructure
          \override Timing.TimeSignature.stencil =
          #ly:text-interface::print
          \override Timing.TimeSignature.text =
          #mkup
    #})


{
```

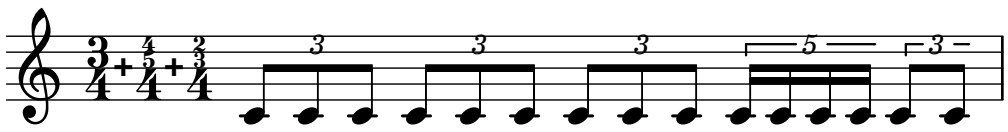
```
258
259       \compoundFractionalTimeSignatureCTwo
260               #'((3 4)(2 3 4)) 11/12 3,3,3,2
261       \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
262       \tuplet 3/2 {c' c' c'}
263       \incompleteTupletBracket \tuplet 3/2 {c' c'}
264   }
```

### 11.6.4 Discussion

See **Discussion** of the entry *Compound Meter with Two Fractional Time Signatures, Style A.*

**Table of Contents**

## 11.7 Compound Meter with Three Fractional Time Signatures, Style A



### 11.7.1 Description

This is an implementation of a compound meter with three fractional time signatures with Style A.

### 11.7.2 Grammar

```
\compoundFractionalTimeSignatureAThree
     #'((TIME_SIG1)(TIME_SIG2)(TIME_SIG3)) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. Following the convention of `\compoundMeter` to enter the two time signatures, you will create a list of lists. Each `TIME_SIG` accepts:

   - an ordinary time signature (list with two numbers);
   - a time signature with a fraction (list with three numbers), or;
   - a time signature with an ordinary numerator and a fraction.

   See *Grammar* of Fractional Time Signatures, Style A for the explanation on the arguments for the order of arguments to specify time signatures.

2. In the code of the given snippet, the value for `MEASURE_SPAN` may appear absurd. However, this results from following the same convention as before, i.e. adding the constituent time signatures to give a general irrational time signature for the entire bar. Thus:

$$\frac{3}{4} + \frac{4}{20} + \frac{2}{12} = \mathbf{\frac{67}{60}}$$

3. `BEAT_STRUCT` follows the same convention as before; however, as the given code shows, it may be necessary to still use `[` and `]` to explicitly specify the beaming.

### 11.7.3 Code

```
\version "2.24.4"
\language "english"

suppressWarning =
#(define-void-function (amount message)(number? string?)
  (for-each
   (lambda (warning)
    (ly:expect-warning message))
   (iota amount 1 1)))

\suppressWarning 3 "strange time signature found"

incompleteTupletBracket =  {
 \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
 \once \override Voice.TupletBracket.bracket-visibility = ##t

}
incompleteSmallTupletBracket =  {
 \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
 \once \override Voice.TupletBracket.bracket-visibility = ##t
 \once \override Voice.TupletNumber.X-offset =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       2.2
       1.2))

 \once \override Voice.TupletBracket.shorten-pair =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       '(-0.7 . 0.15)
       '(-0.3 . 0.8)))
 \once \override Voice.TupletBracket.X-positions =
 #(lambda (grob)
   (if (= UP (ly:grob-property grob 'direction))
       '(1.8 . 3)
       '(0.3 . 2.7)))
}

compoundFractionalTimeSignatureAThree =
```

```
#(define-music-function
  (timeSignatureToShow underlyingMeter beatStructure)
  (list? fraction? number-list?)
  (define mkup
   (markup
    #:concat
    (
     #:override
     (cons 'baseline-skip 0)
     (cond ((= (length (car timeSignatureToShow)) 2)
            (make-center-column-markup
             (list (make-number-markup
                    (number->string
                     (car (car timeSignatureToShow))))
                   (make-number-markup
                    (number->string
                     (cadr (car timeSignatureToShow)))))))

           ((= (length (car timeSignatureToShow)) 3)
            (make-override-markup
             (cons 'baseline-skip 0)
             (make-center-column-markup
              (list
               (make-center-column-markup
                (list
                 (make-translate-markup
                  (cons 0 1)
                  (make-fontsize-markup
                   -6
                   (make-number-markup
                    (number->string
                     (car (car timeSignatureToShow))))))
                 (make-translate-markup
                  (cons 0 0)
                  (make-fontsize-markup
                   -6
                   (make-number-markup
                    (number->string
                     (cadr (car timeSignatureToShow)))))))))
               (make-number-markup
                (number->string
```

```
81                  (caddr (car timeSignatureToShow))))))
82               ))
83
84
85            ((= (length (car timeSignatureToShow)) 4)
86
87             (make-override-markup
88              (cons 'baseline-skip 0)
89              (make-center-column-markup
90               (list
91
92                (make-concat-markup
93                 (list (make-number-markup
94                        (number->string
95                         (car (car timeSignatureToShow))))
96                       (make-halign-markup
97                        -1.5
98                        (make-center-column-markup
99                         (list
100                         (make-translate-markup
101                          (cons 0 1)
102                          (make-fontsize-markup
103                           -6
104                           (make-number-markup
105                            (number->string
106                             (cadr (car timeSignatureToShow))))))
107                         (make-translate-markup
108                          (cons 0 0)
109                          (make-fontsize-markup
110                           -6
111                           (make-number-markup
112                            (number->string
113                             (caddr (car timeSignatureToShow)))))))))))
114               (make-number-markup
115                (number->string
116                 (cadddr (car timeSignatureToShow))))))
117             ))
118            )
119
120
121       #:translate
```

```
122       (cons 0 -0.5)
123       (#:fontsize -12 " ")
124       #:translate
125       (cons 0 -0.5)
126       (#:bold "+")
127       #:translate
128       (cons 0 -0.5)
129       (#:fontsize -12 " ")
130
131       #:override
132       (cons 'baseline-skip 0)
133       (cond ((= (length (cadr timeSignatureToShow)) 2)
134              (make-center-column-markup
135               (list (make-number-markup
136                      (number->string
137                       (car (cadr timeSignatureToShow))))
138                     (make-number-markup
139                      (number->string
140                       (cadr (cadr timeSignatureToShow)))))))
141
142             ((= (length (cadr timeSignatureToShow)) 3)
143
144              (make-override-markup
145               (cons 'baseline-skip 0)
146               (make-center-column-markup
147                (list
148
149
150                 (make-center-column-markup
151                  (list
152                   (make-translate-markup
153                    (cons 0 1)
154                    (make-fontsize-markup
155                     -6
156                     (make-number-markup
157                      (number->string
158                       (car (cadr timeSignatureToShow))))))
159                   (make-translate-markup
160                    (cons 0 0)
161                    (make-fontsize-markup
162                     -6
```

```
163                     (make-number-markup
164                      (number->string
165                       (cadr (cadr timeSignatureToShow))))))))
166                   (make-number-markup
167                    (number->string
168                     (caddr (cadr timeSignatureToShow))))))
169                 ))
170
171
172               ((= (length (cadr timeSignatureToShow)) 4)
173
174                (make-override-markup
175                 (cons 'baseline-skip 0)
176                 (make-center-column-markup
177                  (list
178
179                   (make-concat-markup
180                    (list (make-number-markup
181                           (number->string
182                            (car (cadr timeSignatureToShow))))
183                          (make-halign-markup
184                           -1.5
185                           (make-center-column-markup
186                            (list
187                             (make-translate-markup
188                              (cons 0 1)
189                              (make-fontsize-markup
190                               -6
191                               (make-number-markup
192                                (number->string
193                                 (cadr (cadr timeSignatureToShow))))))
194                             (make-translate-markup
195                              (cons 0 0)
196                              (make-fontsize-markup
197                               -6
198                               (make-number-markup
199                                (number->string
200                                 (caddr (cadr timeSignatureToShow))))))))))
201                   (make-number-markup
202                    (number->string
203                     (cadddr (cadr timeSignatureToShow))))))
```

```
204               ))
205             )
206       #:translate
207       (cons 0 -0.5)
208       (#:fontsize -12 " ")
209       #:translate
210       (cons 0 -0.5)
211       (#:bold "+")
212       #:translate
213       (cons 0 -0.5)
214       (#:fontsize -12 " ")
215
216       #:override
217       (cons 'baseline-skip 0)
218       (cond ((= (length (caddr timeSignatureToShow)) 2)
219              (make-center-column-markup
220               (list (make-number-markup
221                      (number->string
222                       (car (caddr timeSignatureToShow))))
223                     (make-number-markup
224                      (number->string
225                       (cadr (caddr timeSignatureToShow)))))))
226
227             ((= (length (caddr timeSignatureToShow)) 3)
228
229              (make-override-markup
230               (cons 'baseline-skip 0)
231               (make-center-column-markup
232                (list
233
234
235                 (make-center-column-markup
236                  (list
237                   (make-translate-markup
238                    (cons 0 1)
239                    (make-fontsize-markup
240                     -6
241                     (make-number-markup
242                      (number->string
243                       (car (caddr timeSignatureToShow))))))
244                   (make-translate-markup
```

```
245                  (cons 0 0)
246                  (make-fontsize-markup
247                   -6
248                   (make-number-markup
249                    (number->string
250                     (cadr (caddr timeSignatureToShow))))))))
251                (make-number-markup
252                 (number->string
253                  (caddr (caddr timeSignatureToShow))))))
254               ))
255
256
257              ((= (length (caddr timeSignatureToShow)) 4)
258
259               (make-override-markup
260                (cons 'baseline-skip 0)
261                (make-center-column-markup
262                 (list
263
264                  (make-concat-markup
265                   (list (make-number-markup
266                          (number->string
267                           (car (caddr timeSignatureToShow))))
268                         (make-halign-markup
269                          -1.5
270                          (make-center-column-markup
271                           (list
272                            (make-translate-markup
273                             (cons 0 1)
274                             (make-fontsize-markup
275                              -6
276                              (make-number-markup
277                               (number->string
278                                (cadr (caddr timeSignatureToShow))))))
279                            (make-translate-markup
280                             (cons 0 0)
281                             (make-fontsize-markup
282                              -6
283                              (make-number-markup
284                               (number->string
285                                (caddr (caddr timeSignatureToShow)))))))))))
```

```
286                 (make-number-markup
287                  (number->string
288                   (cadddr (caddr timeSignatureToShow))))))
289               ))
290             )
291
292
293       )))
294
295   #{
296    \time $underlyingMeter
297    \set beatStructure = $beatStructure
298
299    \override Timing.TimeSignature.stencil =
300    #ly:text-interface::print
301    \override Timing.TimeSignature.text =
302    #mkup
303   #})
304
305
306 {
307  \compoundFractionalTimeSignatureAThree
308          #'((3 4)(4 5 4)(2 3 4)) 67/60 3,3,3,4,2
309  \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
310  \tuplet 3/2 {c'[ c' c']}
311  \incompleteTupletBracket \tuplet 5/4 {c'16[ c' c' c']}
312  \incompleteTupletBracket \tuplet 3/2 {c'8 c'}
313 }
```

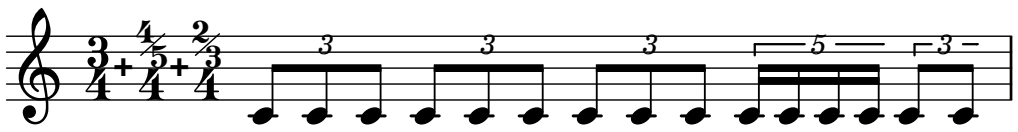
### 11.7.4 Discussion

1. As mentioned in the **Grammar** section, it appears that specifying the beaming in the LilyPond code is still necessary. This is probably because of the unusual value of the fraction that needs to be given in the second argument of the function, `MEASURE_SPAN`.

2. As in the case of the other compound meters introduced in this document, I am hoping to find ways to simplify the code.

**Table of Contents**

## 11.8 Compound Meter with Three Fractional Time Signatures, Style B




### 11.8.1 Description

This is an implementation of a compound meter with three fractional time signatures with Style B.

### 11.8.2 Grammar

```
\compoundFractionalTimeSignatureBThree
      #'((TIME_SIG1)(TIME_SIG2)(TIME_SIG3)) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. See **Grammar** of the entry *Compound Meter with Three Fractional Time Signatures, Style A.*

### 11.8.3 Code

```
1  \version "2.24.4"
2  \language "english"
3
4  % Revised Jan 2 2025 for improving the appearance of fractions
5
6
7  suppressWarning =
8  #(define-void-function (amount message)(number? string?)
9        (for-each
10        (lambda (warning)
11             (ly:expect-warning message))
12        (iota amount 1 1)))
13
14 \suppressWarning 1 "strange time signature found"
15
16 incompleteTupletBracket =  {
17      \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
```

```
18      \once \override Voice.TupletBracket.bracket-visibility = ##t
19
20  }
21  incompleteSmallTupletBracket =  {
22      \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
23      \once \override Voice.TupletBracket.bracket-visibility = ##t
24      \once \override Voice.TupletNumber.X-offset =
25      #(lambda (grob)
26            (if (= UP (ly:grob-property grob 'direction))
27                2.2
28                1.2))
29
30      \once \override Voice.TupletBracket.shorten-pair =
31      #(lambda (grob)
32            (if (= UP (ly:grob-property grob 'direction))
33                '(-0.7 . 0.15)
34                '(-0.3 . 0.8)))
35      \once \override Voice.TupletBracket.X-positions =
36      #(lambda (grob)
37            (if (= UP (ly:grob-property grob 'direction))
38                '(1.8 . 3)
39                '(0.3 . 2.7)))
40  }
41
42  compoundFractionalTimeSignatureBThree =
43  #(define-music-function
44    (timeSignatureToShow underlyingMeter beatStructure)
45    (list? fraction? number-list?)
46    (define mkup
47         (markup
48          #:concat
49
50          (
51
52               #:override
53               (cons 'baseline-skip 0)
54               (cond ((= (length (car timeSignatureToShow)) 2)
55                      (make-center-column-markup
56                       (list (make-number-markup
57                              (number->string
58                               (car (car timeSignatureToShow))))
```

```
                        (make-number-markup
                         (number->string
                          (cadr (car timeSignatureToShow)))))))

                   ((= (length (car timeSignatureToShow)) 3)
                    (make-override-markup
                     (cons 'baseline-skip 0)
                     (make-center-column-markup
                      (list
                       (make-line-markup
                        (list
                         (make-number-markup
                          (make-right-align-markup
                           (make-translate-markup
                            (cons 0 1.6)
                            (make-fontsize-markup
                             -3
                             (number->string
                              (car (car timeSignatureToShow)))))))

                         (make-hspace-markup -1.5)

                         (make-override-markup
                          (cons 'alignment 0)
                          (make-translate-markup
                           (cons 0 0.8)
                           (make-draw-line-markup (cons 1.5 1.35))))

                         (make-hspace-markup -1.5)

                         (make-translate-markup
                          (cons 0 0)
                          (make-fontsize-markup
                           -3
                           (make-number-markup
                            (number->string
                             (cadr (car timeSignatureToShow))))))))

                        (make-number-markup
                         (number->string
                          (caddr (car timeSignatureToShow))))))))
```

```
100
101
102                     ((= (length (car timeSignatureToShow)) 4)
103
104                      (make-override-markup
105                       (cons 'baseline-skip 0)
106                       (make-center-column-markup
107                        (list
108                         (make-line-markup
109                          (list
110                           (make-number-markup
111                            (number->string
112                             (car (car timeSignatureToShow))))
113                           (make-fontsize-markup
114                            -12
115                            (make-simple-markup " "))
116
117                           (make-hspace-markup -1.25)
118                           (make-translate-markup
119                            (cons 0 0.4)
120                            (make-bold-markup
121                             (make-simple-markup "+")))
122
123                           (make-hspace-markup -0.25)
124
125                           (make-hspace-markup -0.5)
126                           (make-right-align-markup
127                            (make-number-markup
128                             (make-translate-markup
129                              (cons 0 1.5)
130                              (make-fontsize-markup
131                               -3
132                               (number->string
133                                (cadr (car timeSignatureToShow)))))))
134
135                           (make-hspace-markup -1.5)
136
137                           (make-override-markup
138                            (cons 'alignment 0)
139                            (make-translate-markup
140                             (cons 0 0.8)
```

```
                          (make-draw-line-markup
                           (cons 1.5 1.35))))

                        (make-hspace-markup -1.5)

                        (make-number-markup
                         (make-left-align-markup
                          (make-fontsize-markup
                           -3
                           (number->string
                            (caddr (car timeSignatureToShow)))))))

                       (make-number-markup
                        (number->string
                         (cadddr (car timeSignatureToShow)))))))))


                #:translate
                (cons 0 -0.5)
                (#:fontsize -12 " ")
                #:translate
                (cons 0 -0.5)
                (#:bold "+")
                #:translate
                (cons 0 -0.5)
                (#:fontsize -12 " ")

                #:override
                (cons 'baseline-skip 0)

                (cond ((= (length (cadr timeSignatureToShow)) 2)
                       (make-center-column-markup
                        (list (make-number-markup
                               (number->string
                                (car (cadr timeSignatureToShow))))
                              (make-number-markup
                               (number->string
                                (cadr (cadr timeSignatureToShow)))))))

                      ((= (length (cadr timeSignatureToShow)) 3)
                       (make-override-markup
```

```
                    (cons 'baseline-skip 0)
                    (make-center-column-markup
                     (list
                      (make-line-markup
                       (list
                        (make-number-markup
                         (make-right-align-markup
                          (make-translate-markup
                           (cons 0 1.6)
                           (make-fontsize-markup
                            -3
                            (number->string
                             (car (cadr timeSignatureToShow)))))))

                        (make-hspace-markup -1.5)

                        (make-override-markup
                         (cons 'alignment 0)
                         (make-translate-markup
                          (cons 0 0.8)
                          (make-draw-line-markup (cons 1.5 1.35))))

                        (make-hspace-markup -1.5)

                        (make-translate-markup
                         (cons 0 0)
                         (make-fontsize-markup
                          -3
                          (make-number-markup
                           (number->string
                            (cadr (cadr timeSignatureToShow)))))))

                       (make-number-markup
                        (number->string
                         (caddr (cadr timeSignatureToShow)))))))


                   ((= (length (cadr timeSignatureToShow)) 4)

                    (make-override-markup
                     (cons 'baseline-skip 0)
```

```
223                     (make-center-column-markup
224                      (list
225                       (make-line-markup
226                        (list
227                         (make-number-markup
228                          (number->string
229                           (car (cadr timeSignatureToShow))))
230                         (make-fontsize-markup
231                          -12
232                          (make-simple-markup " "))
233
234                         (make-hspace-markup -1.25)
235                         (make-translate-markup
236                          (cons 0 0.4)
237                          (make-bold-markup
238                           (make-simple-markup "+")))
239
240                         (make-hspace-markup -0.25)
241
242                         (make-hspace-markup -0.5)
243                         (make-right-align-markup
244                          (make-number-markup
245                           (make-translate-markup
246                            (cons 0 1.5)
247                            (make-fontsize-markup
248                             -3
249                             (number->string
250                              (cadr (cadr timeSignatureToShow)))))))
251
252                         (make-hspace-markup -1.5)
253
254                         (make-override-markup
255                          (cons 'alignment 0)
256                          (make-translate-markup
257                           (cons 0 0.8)
258                           (make-draw-line-markup
259                            (cons 1.5 1.35))))
260
261                         (make-hspace-markup -1.5)
262
263                         (make-number-markup
```

```
                        (make-left-align-markup
                         (make-fontsize-markup
                          -3
                          (number->string
                           (caddr (cadr timeSignatureToShow))))))))

                      (make-number-markup
                       (number->string
                        (cadddr (cadr timeSignatureToShow)))))))))

                #:translate
                (cons 0 -0.5)
                (#:fontsize -12 " ")
                #:translate
                (cons 0 -0.5)
                (#:bold "+")
                #:translate
                (cons 0 -0.5)
                (#:fontsize -12 " ")

                #:override
                (cons 'baseline-skip 0)

                (cond ((= (length (caddr timeSignatureToShow)) 2)
                       (make-center-column-markup
                        (list (make-number-markup
                               (number->string
                                (car (caddr timeSignatureToShow))))
                              (make-number-markup
                               (number->string
                                (cadr (caddr timeSignatureToShow)))))))

                      ((= (length (caddr timeSignatureToShow)) 3)
                       (make-override-markup
                        (cons 'baseline-skip 0)
                        (make-center-column-markup
                         (list
                          (make-line-markup
                           (list
                            (make-number-markup
                             (make-right-align-markup
```

```
                        (make-translate-markup
                         (cons 0 1.6)
                         (make-fontsize-markup
                          -3
                          (number->string
                           (car (caddr timeSignatureToShow)))))))

                       (make-hspace-markup -1.5)

                       (make-override-markup
                        (cons 'alignment 0)
                        (make-translate-markup
                         (cons 0 0.8)
                         (make-draw-line-markup (cons 1.5 1.35))))

                       (make-hspace-markup -1.5)

                       (make-translate-markup
                        (cons 0 0)
                        (make-fontsize-markup
                         -3
                         (make-number-markup
                          (number->string
                           (cadr (caddr timeSignatureToShow)))))))

                     (make-number-markup
                      (number->string
                       (caddr (caddr timeSignatureToShow)))))))


                   ((= (length (caddr timeSignatureToShow)) 4)

                    (make-override-markup
                     (cons 'baseline-skip 0)
                     (make-center-column-markup
                      (list
                       (make-line-markup
                        (list
                         (make-number-markup
                          (number->string
                           (car (caddr timeSignatureToShow))))
```

```
346                          (make-fontsize-markup
347                           -12
348                           (make-simple-markup " "))
349
350                          (make-hspace-markup -1.25)
351                          (make-translate-markup
352                           (cons 0 0.4)
353                           (make-bold-markup
354                            (make-simple-markup "+")))
355
356                          (make-hspace-markup -0.25)
357
358                          (make-hspace-markup -0.5)
359                          (make-right-align-markup
360                           (make-number-markup
361                            (make-translate-markup
362                             (cons 0 1.5)
363                             (make-fontsize-markup
364                              -3
365                              (number->string
366                               (cadr (caddr timeSignatureToShow)))))))
367
368                          (make-hspace-markup -1.5)
369
370                          (make-override-markup
371                           (cons 'alignment 0)
372                           (make-translate-markup
373                            (cons 0 0.8)
374                            (make-draw-line-markup
375                             (cons 1.5 1.35))))
376
377                          (make-hspace-markup -1.5)
378
379                          (make-number-markup
380                           (make-left-align-markup
381                            (make-fontsize-markup
382                             -3
383                             (number->string
384                              (caddr (caddr timeSignatureToShow))))))))
385
386                        (make-number-markup
```

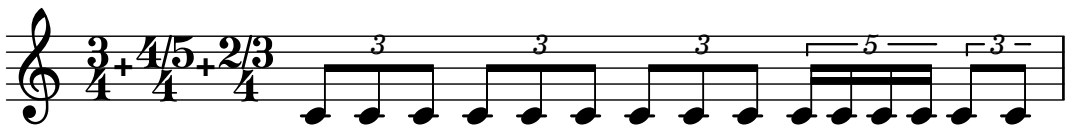
```
                          (number->string
                           (cadddr (caddr timeSignatureToShow)))))))))
               )

         ))

   #{
        \time $underlyingMeter
        \set beatStructure = $beatStructure

        \override Timing.TimeSignature.stencil =
        #ly:text-interface::print
        \override Timing.TimeSignature.text =
        #mkup
   #})

{

      \compoundFractionalTimeSignatureBThree
              #'((3 4)(4 5 4)(2 3 4)) 67/60 3,3,3,4,2
      \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
      \tuplet 3/2 {c'[ c' c']}
      \incompleteTupletBracket \tuplet 5/4 {c'16[ c' c' c']}
      \incompleteTupletBracket \tuplet 3/2 {c'8 c'}
}
```

### 11.8.4 Discussion

See **Discussion** of the entry *Compound Meter with Three Fractional Time Signatures, Style A.*

**Table of Contents**

## 11.9 Compound Meter with Three Fractional Time Signatures, Style C



### 11.9.1 Description

This is an implementation of a compound meter with three fractional time signatures with Style C.

### 11.9.2 Grammar

```
\compoundFractionalTimeSignatureCThree
       #'((TIME_SIG1)(TIME_SIG2)(TIME_SIG3)) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. See **Grammar** of the entry *Compound Meter with Three Fractional Time Signatures, Style A.*

### 11.9.3 Code

```
\version "2.24.4"
\language "english"

% Revised Jan 2 2025 for improving the appearance of fractions

suppressWarning =
#(define-void-function (amount message)(number? string?)
      (for-each
       (lambda (warning)
            (ly:expect-warning message))
       (iota amount 1 1)))

\suppressWarning 1 "strange time signature found"

incompleteTupletBracket =  {
     \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
     \once \override Voice.TupletBracket.bracket-visibility = ##t
```

```
18
19  }
20  incompleteSmallTupletBracket =  {
21       \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
22       \once \override Voice.TupletBracket.bracket-visibility = ##t
23       \once \override Voice.TupletNumber.X-offset =
24       #(lambda (grob)
25             (if (= UP (ly:grob-property grob 'direction))
26                 2.2
27                 1.2))
28
29       \once \override Voice.TupletBracket.shorten-pair =
30       #(lambda (grob)
31             (if (= UP (ly:grob-property grob 'direction))
32                 '(-0.7 . 0.15)
33                 '(-0.3 . 0.8)))
34       \once \override Voice.TupletBracket.X-positions =
35       #(lambda (grob)
36             (if (= UP (ly:grob-property grob 'direction))
37                 '(1.8 . 3)
38                 '(0.3 . 2.7)))
39  }
40
41  compoundFractionalTimeSignatureCThree =
42  #(define-music-function
43    (timeSignatureToShow underlyingMeter beatStructure)
44    (list? fraction? number-list?)
45    (define mkup
46         (markup
47          #:concat
48          (
49
50               #:override
51               (cons 'baseline-skip 0)
52               (cond ((= (length (car timeSignatureToShow)) 2)
53                      (make-center-column-markup
54                       (list (make-number-markup
55                              (number->string
56                               (car (car timeSignatureToShow))))
57                             (make-number-markup
58                              (number->string
```

```
59                          (cadr (car timeSignatureToShow)))))))
60
61                      ((= (length (car timeSignatureToShow)) 3)
62                       (make-override-markup
63                        (cons 'baseline-skip 0)
64                        (make-center-column-markup
65                         (list
66                          (make-line-markup
67                           (list
68
69                            (make-right-align-markup
70                             (make-number-markup
71                              (number->string
72                               (car (car timeSignatureToShow)))))
73
74                            (make-hspace-markup -0.6)
75
76                            (make-override-markup
77                             (list (cons 'alignment 0)
78                                   (cons 'thickness 2))
79                             (make-draw-line-markup
80                              (cons 0.5 2)))
81
82                            (make-hspace-markup -0.6)
83
84                            (make-number-markup
85                             (make-left-align-markup
86                              (number->string
87                               (cadr (car timeSignatureToShow)))))))
88
89                          (make-number-markup
90                           (number->string
91                            (caddr (car timeSignatureToShow)))))))))
92
93
94                      ((= (length (car timeSignatureToShow)) 4)
95
96                       (make-override-markup
97                        (cons 'baseline-skip 0)
98                        (make-center-column-markup
99                         (list
```

```
100                     (make-line-markup
101                      (list
102                       (make-number-markup
103                        (number->string
104                         (car (car timeSignatureToShow))))
105                       (make-fontsize-markup
106                        -12
107                        (make-simple-markup " "))
108
109
110                       (make-hspace-markup -1.25)
111                       (make-translate-markup
112                        (cons 0 0.4)
113                        (make-bold-markup
114                         (make-simple-markup "+")))
115
116                       (make-hspace-markup -0.25)
117
118                       (make-hspace-markup -0.5)
119                       (make-right-align-markup
120                        (make-number-markup
121                         (number->string
122                          (cadr (car timeSignatureToShow)))))
123
124                       (make-hspace-markup -0.6)
125
126                       (make-override-markup
127                        (list (cons 'alignment 0)
128                              (cons 'thickness 2))
129                        (make-draw-line-markup (cons 0.5 2)))
130
131                       (make-hspace-markup -0.6)
132
133                       (make-number-markup
134                        (make-left-align-markup
135                         (number->string
136                          (caddr (car timeSignatureToShow)))))))
137
138                     (make-number-markup
139                      (number->string
140                       (cadddr (car timeSignatureToShow))))))))
```

```
141
142
143             #:translate
144             (cons 0 -0.5)
145             (#:fontsize -12 " ")
146             #:translate
147             (cons 0 -0.5)
148             (#:bold "+")
149             #:translate
150             (cons 0 -0.5)
151             (#:fontsize -12 " ")
152
153             #:override
154             (cons 'baseline-skip 0)
155
156             (cond ((= (length (cadr timeSignatureToShow)) 2)
157                    (make-center-column-markup
158                     (list (make-number-markup
159                            (number->string
160                             (car (cadr timeSignatureToShow))))
161                           (make-number-markup
162                            (number->string
163                             (cadr (cadr timeSignatureToShow)))))))
164
165                   ((= (length (cadr timeSignatureToShow)) 3)
166                    (make-override-markup
167                     (cons 'baseline-skip 0)
168                     (make-center-column-markup
169                      (list
170                       (make-line-markup
171                        (list
172
173                         (make-right-align-markup
174                          (make-number-markup
175                           (number->string
176                            (car (cadr timeSignatureToShow)))))
177
178                         (make-hspace-markup -0.6)
179
180                         (make-override-markup
181                          (list (cons 'alignment 0)
```

```
182                          (cons 'thickness 2))
183                         (make-draw-line-markup
184                          (cons 0.5 2)))
185
186                        (make-hspace-markup -0.6)
187
188                        (make-number-markup
189                         (make-left-align-markup
190                          (number->string
191                           (cadr (cadr timeSignatureToShow)))))))
192
193                       (make-number-markup
194                        (number->string
195                         (caddr (cadr timeSignatureToShow))))))))
196
197
198                     ((= (length (cadr timeSignatureToShow)) 4)
199
200                      (make-override-markup
201                       (cons 'baseline-skip 0)
202                       (make-center-column-markup
203                        (list
204                         (make-line-markup
205                          (list
206                           (make-number-markup
207                            (number->string
208                             (car (cadr timeSignatureToShow))))
209                           (make-fontsize-markup
210                            -12
211                            (make-simple-markup " "))
212
213
214                           (make-hspace-markup -1.25)
215                           (make-translate-markup
216                            (cons 0 0.4)
217                            (make-bold-markup
218                             (make-simple-markup "+")))
219
220                           (make-hspace-markup -0.25)
221
222                           (make-hspace-markup -0.5)
```

```
223                         (make-right-align-markup
224                          (make-number-markup
225                           (number->string
226                            (cadr (cadr timeSignatureToShow)))))
227
228                         (make-hspace-markup -0.6)
229
230                         (make-override-markup
231                          (list (cons 'alignment 0)
232                                (cons 'thickness 2))
233                          (make-draw-line-markup (cons 0.5 2)))
234
235                         (make-hspace-markup -0.6)
236
237                         (make-number-markup
238                          (make-left-align-markup
239                           (number->string
240                            (caddr (cadr timeSignatureToShow)))))))
241
242                       (make-number-markup
243                        (number->string
244                         (cadddr (cadr timeSignatureToShow)))))))))
245
246                 #:translate
247                 (cons 0 -0.5)
248                 (#:fontsize -12 " ")
249                 #:translate
250                 (cons 0 -0.5)
251                 (#:bold "+")
252                 #:translate
253                 (cons 0 -0.5)
254                 (#:fontsize -12 " ")
255
256                 #:override
257                 (cons 'baseline-skip 0)
258
259                 (cond ((= (length (caddr timeSignatureToShow)) 2)
260                        (make-center-column-markup
261                         (list (make-number-markup
262                                (number->string
263                                 (car (caddr timeSignatureToShow))))
```

```
                    (make-number-markup
                     (number->string
                      (cadr (caddr timeSignatureToShow)))))))

                ((= (length (caddr timeSignatureToShow)) 3)
                 (make-override-markup
                  (cons 'baseline-skip 0)
                  (make-center-column-markup
                   (list
                    (make-line-markup
                     (list

                      (make-right-align-markup
                       (make-number-markup
                        (number->string
                         (car (caddr timeSignatureToShow)))))

                      (make-hspace-markup -0.6)

                      (make-override-markup
                       (list (cons 'alignment 0)
                             (cons 'thickness 2))
                       (make-draw-line-markup
                        (cons 0.5 2)))

                      (make-hspace-markup -0.6)

                      (make-number-markup
                       (make-left-align-markup
                        (number->string
                         (cadr (caddr timeSignatureToShow)))))))

                    (make-number-markup
                     (number->string
                      (caddr (caddr timeSignatureToShow)))))))))

                ((= (length (caddr timeSignatureToShow)) 4)

                 (make-override-markup
                  (cons 'baseline-skip 0)
```

```
305                     (make-center-column-markup
306                      (list
307                       (make-line-markup
308                        (list
309                         (make-number-markup
310                          (number->string
311                           (car (caddr timeSignatureToShow))))
312                         (make-fontsize-markup
313                          -12
314                          (make-simple-markup " "))
315
316
317                         (make-hspace-markup -1.25)
318                         (make-translate-markup
319                          (cons 0 0.4)
320                          (make-bold-markup
321                           (make-simple-markup "+")))
322
323                         (make-hspace-markup -0.25)
324
325                         (make-hspace-markup -0.5)
326                         (make-right-align-markup
327                          (make-number-markup
328                           (number->string
329                            (cadr (caddr timeSignatureToShow)))))
330
331                         (make-hspace-markup -0.6)
332
333                         (make-override-markup
334                          (list (cons 'alignment 0)
335                                (cons 'thickness 2))
336                          (make-draw-line-markup (cons 0.5 2)))
337
338                         (make-hspace-markup -0.6)
339
340                         (make-number-markup
341                          (make-left-align-markup
342                           (number->string
343                            (caddr (caddr timeSignatureToShow)))))))
344
345                       (make-number-markup
```

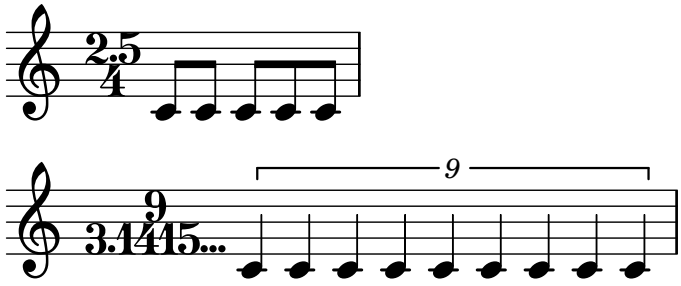
```
                          (number->string
                           (cadddr (caddr timeSignatureToShow)))))))))
                )
         ))

   #{
        \time $underlyingMeter
        \set beatStructure = $beatStructure

        \override Timing.TimeSignature.stencil =
        #ly:text-interface::print
        \override Timing.TimeSignature.text =
        #mkup
   #})


{

      \compoundFractionalTimeSignatureCThree
              #'((3 4)(4 5 4)(2 3 4)) 67/60 3,3,3,4,2
      \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
      \tuplet 3/2 {c'[ c' c']}
      \incompleteTupletBracket \tuplet 5/4 {c'16[ c' c' c']}
      \incompleteTupletBracket \tuplet 3/2 {c'8 c'}
}
```

### 11.9.4 Discussion

See **Discussion** of the entry *Compound Meter with Three Fractional Time Signatures, Style A.*

**Table of Contents**

## 11.10 Time Signature with Decimals




### 11.10.1 Description

This is an implementation of a time signature with decimals. This function allows the user to use decimals for both numerator and denominator values of the time signature. While there are many examples where the numerator value contains decimals, notable examples for denominators that contain decimals include works by Mark Andre, such as *Un-fini I* (1995) for harp,[16] *Un-fini II* (1994-95) for harpsichord,[17] and *Contrapunctus* (1998/99) for piano..[18] This function also has the option of showing ellipsis, as discussed in **Grammar**.

### 11.10.2 Grammar

```
\decimalPointTimeSignature
       #'((NUMERATOR)(DENOMINATOR)) MEASURE_SPAN BEAT_STRUCT
```

**NB**

1. The first argument takes a list of two lists. For both `NUMERATOR` and `DENOMINATOR`, one or two numbers can be placed.
   (a) If only one number is placed, it is treated as an integer. For example, `#'((3)(4))` would print: $\frac{3}{4}$.
   (b) If two numbers are placed, the first number is the integer portion of the number, and the second number is the decimals. For example, `#'((3 5)(4 232))` would print: $\frac{3.5}{4.232}$.

16. Mark Andre, *Un-fini I : 1995, für eine Harfenistin/einen Harfenisten (Harfe, Tam-tam, grosse Trommel)*, Neue Musik bei Carus (Stuttgart: Carus, 1997).
17. Mark Andre, *Un-fini II : pour clavecin (1996)* (Paris: Editions Durand, 1998).
18. Mark Andre, *Contrapunctus : pour piano* (Paris: Durand, 2006).

(c) Placing a dot `.` at the end of the second number will print the ellipsis `...` at the end. This is useful for notating infinite decimal representations. For example, `#'((3 14159.)(4))` would print: $\frac{3.14159...}{4}$.

2. `MEASURE_SPAN` denotes how the measure may be written without the use of "decimal point" time signature.

3. `BEAT_STRUCT` indicates beat structure, by which the beaming of the measure abides.

### 11.10.3 Code

```
\version "2.24.4"
\language "english"


decimalPointTimeSignature =
#(define-music-function
  (timeSignatureToShow underlyingMeter beatStructure)
  (list? fraction? number-list?)

  (define (is-float? x)
    (and (number? x) (inexact? x)))

  #{
    \time $underlyingMeter
    \set beatStructure = $beatStructure
    \override Staff.TimeSignature.stencil =
    #ly:text-interface::print
    \override Staff.TimeSignature.text =
    #(markup
      (make-override-markup
       (cons 'baseline-skip 0)
       (make-center-column-markup
        (list
         (if (= (length (car  timeSignatureToShow)) 1)
             (make-number-markup
              (number->string

               (car (car timeSignatureToShow))))
             (make-line-markup
              (list
               (make-number-markup
```

```
32                  (number->string
33                   (car (car timeSignatureToShow))))
34                 (make-hspace-markup -0.5)
35                 (make-translate-markup
36                  '(0 . 0.15)
37                  (make-musicglyph-markup "period"))
38                 (make-hspace-markup -0.5)
39                 (if (not (is-float? (cadr (car timeSignatureToShow))))
40                     (make-number-markup
41                      (number->string
42                       (inexact->exact (cadr (car timeSignatureToShow)))
43                       ))
44                     (make-line-markup
45                      (list (make-number-markup
46                             (number->string
47                              (inexact->exact
48                               (cadr (car timeSignatureToShow)))
49                              ))
50                            (make-hspace-markup -0.5)
51                            (make-translate-markup
52                             '(0 . 0.15)
53                             (make-musicglyph-markup "period"))
54                            (make-hspace-markup -0.5)
55                            (make-translate-markup
56                             '(0 . 0.15)
57                             (make-musicglyph-markup "period"))
58                            (make-hspace-markup -0.5)
59                            (make-translate-markup
60                             '(0 . 0.15)
61                             (make-musicglyph-markup "period"))))
62                     )
63                  )
64                 )
65                )
66           (if (= (length (cadr  timeSignatureToShow)) 1)
67               (make-number-markup
68                (number->string
69                 (car (cadr timeSignatureToShow))))
70               (make-line-markup
71                (list
72                 (make-number-markup
```

```
                 (number->string
                  (car (cadr timeSignatureToShow))))
                (make-hspace-markup -0.5)
                (make-translate-markup
                 '(0 . 0.15)
                 (make-musicglyph-markup "period"))
                (make-hspace-markup -0.5)
                (if (not (is-float? (cadr (cadr timeSignatureToShow))))
                    (make-number-markup
                     (number->string
                      (inexact->exact (cadr (cadr timeSignatureToShow)))
                      ))
                    (make-line-markup
                     (list (make-number-markup
                            (number->string
                             (inexact->exact
                              (cadr (cadr timeSignatureToShow)))
                             ))
                           (make-hspace-markup -0.5)
                           (make-translate-markup
                            '(0 . 0.15)
                            (make-musicglyph-markup "period"))
                           (make-hspace-markup -0.5)
                           (make-translate-markup
                            '(0 . 0.15)
                            (make-musicglyph-markup "period"))
                           (make-hspace-markup -0.5)
                           (make-translate-markup
                            '(0 . 0.15)
                            (make-musicglyph-markup "period"))))
                    )
                 )
                )
               )
              )
             ))
        (make-hspace-markup -1))
  #})


{
```

```
    \decimalPointTimeSignature #'((2 5)(4)) 5/8 2,3
    c'8 c' c' c' c'
}


{
    \decimalPointTimeSignature #'((9)(3 1415.)) 9/4 3,3,3
    \tuplet 9/9 {c'4 4 4 4 4 4 4 4 4}
}
```

### 11.10.4 Discussion

The structure of the code where the user specifies integer and decimal portions of either numerator, denominator, or both, resulted from the fact that the period "." by default appeared too close to the staff line and the denominator, possibly rendering the time signature difficult to read. In the code I made these periods appear via `\translate` feature, where I offset the period upward by the value of 0.15, allowing the period sign to be separated from the staff line and the denominator.

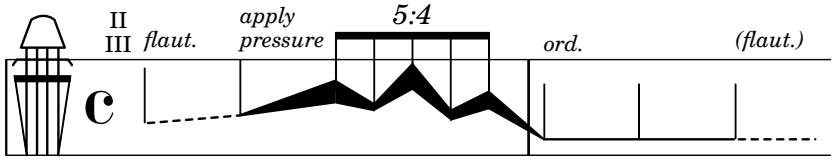
**Table of Contents**

# Chapter 12

# Combinations

This chapter presents examples that combine several snippets from the previous chapters. **Variables Used** provides a comprehensive list of all the variables required to generate the snippet. Among these, indented variables indicate "variables of a variable," i.e., dependent variables necessary for the main variables to function. The **Code** section only lists the score portion of the LilyPond code.

## 12.1 Prescriptive Notation for String Instruments




### 12.1.1 Description

An example of a prescriptive notation for a string instrument. Vertical placement of the notehead corresponds to the position at which bowing takes place. Horizontally it shows the change of the bow pressure against the string(s).

### 12.1.2 Variables Used

```
\strPosClef
  \strPosClefDesign
  \strPosClefSize
```

```
\dashedLineNotehead
\modularLineNotehead
\noteheadless
```

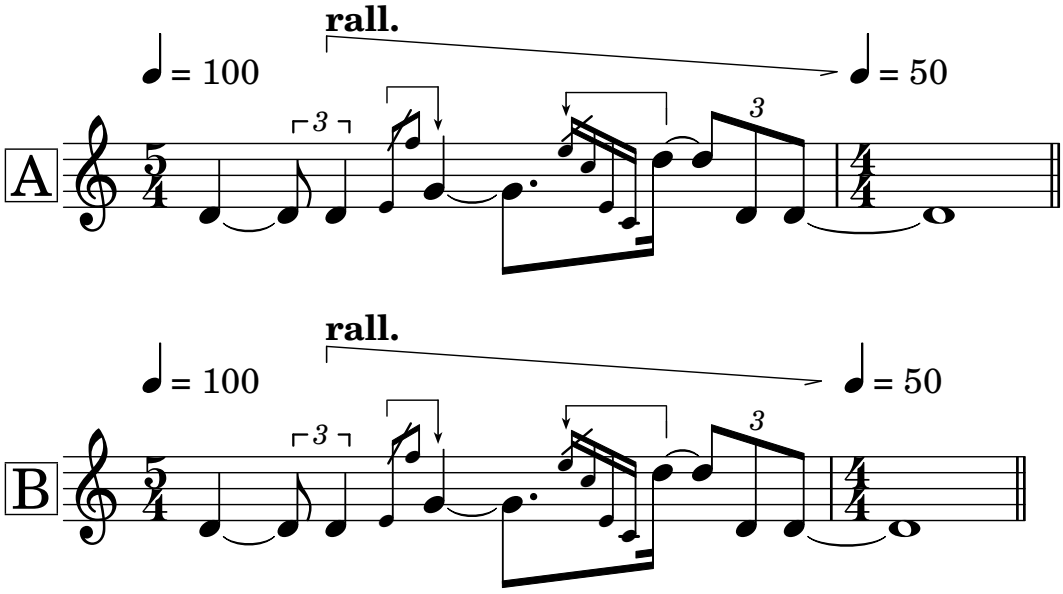
### 12.1.3 Code

```
1
2  \score {
3    {
4      \override Staff.StaffSymbol.line-positions = #'(6  -6)
5      \strPosClef
6      \dashedLineNotehead g'4
7              ^\markup {\fontsize #-4 \italic flaut.}
8              ^\markup \translate #'(-2.5 . -0) \center-column
9                          {\translate #'(0 . -1.5) \fontsize #-4 II
10                         \fontsize #-4 III}
11                 a' #6
12     \modularLineNotehead a'
13             ^\markup \column {\translate #'(0 . -1.5)
14                         \fontsize #-4 \italic apply \fontsize #-4
15                         \italic pressure}
16                 d'' #15 #150 #6
17     \override TupletNumber.text = #tuplet-number::calc-fraction-text
18     \stemUp  \tuplet 5/4 {
19       \modularLineNotehead d''8 b' #150 #50 #2.5
20       \modularLineNotehead b' f'' #50 #175 #2.5
21       \modularLineNotehead f'' a' #175 #70 #2.5
22       \modularLineNotehead a' c'' #70 #120 #2.5
23       \modularLineNotehead c'' e' #120 #15 #3.5
24     }
25     |
26     \modularLineNotehead e'4
27             ^\markup {\fontsize #-4 \italic ord.}
28                 e' #15 #15 #12
29     \noteheadless e'
30     \dashedLineNotehead e'
31             ^\markup {\fontsize #-4 \italic (flaut.)}
32                 e' #5
33   }
34
35   \layout {
```

```
36      \context {
37        \Score proportionalNotationDuration = #(ly:make-moment 1/10)
38                        \override SpacingSpanner.uniform-stretching = ##t
39      }
40    }
41  }
42
```

Table of Contents

## 12.2 Multiple Instances Of Spanners At Once




### 12.2.1 Description

Invoking two or more Text Spanners (that require `\stopTextSpan` for them to finish their processes) all on one single layer could cause the spanners to behave unexpectedly. This entry is an attempt to avoid such unexpected behaviors by invoking a spanner per layer (A), or per staff line (B).

### 12.2.2 Variables Used

```
\startSlashedGraceMusic
\stopSlashedGraceMusic
\graceNoteBeforeBeatOn
\graceNoteBeforeBeatOff
\graceNoteAfterBeatOn
\graceNoteAfterBeatOff
\rallArrow
```

### 12.2.3 Code

```
1
2  %%%%%%%%%%%%%%%%% A %%%%%%%%%%%%%%%%%
3  \score {
4    \new Staff = "allElementsCombined"
5    \with {instrumentName = \markup {\fontsize #4 \box "A"}}  {
6      \numericTimeSignature
7      \override Score.MetronomeMark.Y-offset = #5.75
8      \tempo 4 = 100
9      \time 5/4
10     <<
11       {
12         \tieNeutral \stemNeutral d'4~
13         \tuplet 3/2 {d'8 d'4}
14         \stemUp \grace  {
15           \startSlashedGraceMusic \graceNoteBeforeBeatOn e'8 f''
16           \stopSlashedGraceMusic
17         } \graceNoteBeforeBeatOff g'4~
18         \stemNeutral g'8.[ \grace {
19         \startSlashedGraceMusic \graceNoteAfterBeatOn
20           e''16 c'' e' c' \stopSlashedGraceMusic
21         }
22         \graceNoteAfterBeatOff d''16]~
23         \tuplet 3/2 {d''8 d'8 d'8~} |
24         \time 4/4
25         d'1 \bar"||"
26       }
27       \\
28       {
29         s4  \tuplet 3/2 {
30           s8 \override Voice.TextSpanner.Y-offset = #6.5
31           s4^\markup {\translate #'(0 . 6.5) \bold "rall."}
32           \rallArrow #4
33         } s2. \tempo 4 = 50 s4*4 \stopTextSpan
34       }
35     >>
36   }
37 }
38
39
```

```
40
41
42
43  %%%%%%%%%%%%%%%%%% B %%%%%%%%%%%%%%%%%%
44  \score {
45    <<
46      \new Staff = "tempoLine" \with {
47        \remove Clef_engraver
48        \remove Staff_symbol_engraver
49        \remove Time_signature_engraver
50      }
51      {
52        \numericTimeSignature
53        \override Score.MetronomeMark.Y-offset = #6
54        \tempo 4 = 100
55        \time 5/4
56        s4  \tuplet 3/2 {
57          s8 \override Voice.TextSpanner.Y-offset = #-2.25
58          s4^\markup {\translate #'(0 . 0) \bold "rall."}
59          \rallArrow #4} s2 \after 64*15 \stopTextSpan s8*2 |
60        \tempo 4 = 50  s4*4
61      }
62      \new Staff = "music"
63      \with { instrumentName = \markup {\fontsize #4 \box "B"}}
64      {
65        \tieNeutral \stemNeutral d'4~
66        \tuplet 3/2 {d'8 d'4}
67        \grace {
68          \startSlashedGraceMusic \graceNoteBeforeBeatOn e'8 f''
69          \stopSlashedGraceMusic
70        } \graceNoteBeforeBeatOff g'4~
71        g'8.[ \grace { \startSlashedGraceMusic \graceNoteAfterBeatOn
72          e''16 c'' e' c' \stopSlashedGraceMusic
73        }
74        \graceNoteAfterBeatOff d''16]~
75        \tuplet 3/2 {d''8 d'8 d'8~} |
76        \time 4/4
77        d'1 \bar"||"
78      }
79    >>
80  }
```

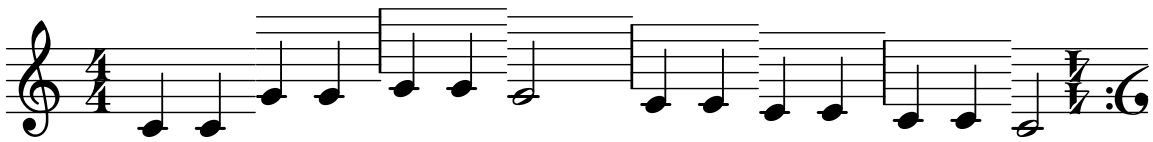
**Table of Contents**

# Chapter 13

# Miscellanies

This chapter presents snippets that do not really belong to any of the other preceding chapters but I learned tremendously from making. Quite often I have made these snippets as a diversion.

## 13.1 Shifting Staffs, Rotated Clef and Time Signature



### 13.1.1 Description

Staff lines that are shifted so that, when the note moves away from the middle C, the staff lines move accordingly. The excerpt ends with a time signature and a clef that are rotated 180 degrees.

### 13.1.2 Code

```
\version "2.24.4"
\language "english"

staone = {
  \stopStaff
  \override Staff.StaffSymbol.line-positions =
  #'(0 2 4 6 8)
```

```
  \startStaff
}
statwo = {
  \stopStaff
  \override Staff.StaffSymbol.line-positions =
  #'(1 3 5 7 9)
  \startStaff
}
stathree = {
  \stopStaff
  \override Staff.StaffSymbol.line-positions =
  #'(-1 1 3 5 7)
  \startStaff
}
stafour = {
  \stopStaff
  \override Staff.StaffSymbol.line-positions =
  #'(-2 0 2 4 6)
  \startStaff
}
stafive = {
  \stopStaff
  \override Staff.StaffSymbol.line-positions =
  #'(-3 -1 1 3 5)
  \startStaff
}
stanorm = {
  \stopStaff
  \revert Staff.StaffSymbol.line-positions
  \startStaff
}
{
  \numericTimeSignature
  \time 4/4

  c'4 c' \staone g' g' \statwo a' a' \staone g'2
  \stathree f'4 f' \stafour e' e' \stafive d' d' \stanorm
  \override TextScript.outside-staff-priority = ##f
  \once \override TextScript.extra-offset = #'(0 . -4.5)
  c'2 ^\markup \concat {
    {
```

```
49        \hspace #3 \rotate #180
50        {\compound-meter #'(4 4)}
51      }
52      {
53        \translate-scaled #'(1 . 0.5)
54        \rotate #180 \musicglyph "clefs.F"
55      }
56    }
57    \bar ""
58
59  }
60
61  \layout {
62    \context{
63      \Score   proportionalNotationDuration = #(ly:make-moment 1/7)
64    }
65  }
```

**Table of Contents**

# Chapter 14

# Exploring Scheme

## 14.1 Introduction

Scheme, one of the dialects of the Lisp family of programming languages, is used in LilyPond as its extension language. Scheme allows LilyPond users to explore the inner workings of the program, enabling significant customization. The snippets in this document would not exist without taking advantage of it.[1]

However, learning Scheme can be daunting. In his unfinished book on Scheme and LilyPond, Urs Liska refers to its "thorny path."[2] While I have experience with Common Lisp (another Lisp dialect) from my work with OpenMusic, adjusting to Scheme's grammatical nuances still took some time.

This chapter does not aim to be a comprehensive guide to using Scheme in LilyPond.[3] Instead, it offers suggestions for newcomers to familiarize themselves with Scheme.

### 14.1.1 Step 1a: Focus on the Scheme Language Itself

Scheme is a language distinct from LilyPond, and understanding this distinction is essential. For simpler LilyPond tasks, Scheme may not be necessary. However, when working with internal parameters, Scheme allows deeper customization. It is beneficial to first study Scheme independently, learning its syntax and concepts by writing simple code.

1. For newcomers: parts of LilyPond code written in Scheme are often enclosed in `#(` and `)`. Numerical values preceded by `#`, and number pairs such as `\#'(1 . -2)`, are also part of the Scheme language.

2. Urs Liska, *Understanding Scheme In LilyPond*, vol. 2024, December 19 (2020), Web Page, https://scheme-book.readthedocs.io/en/latest/.

3. For a deeper dive, refer to the resource by Liska, as well as Jean Abou Samra, *Extending LilyPond*, vol. 2024, December 19 (2021), Web Page, https://extending-lilypond.gitlab.io/en/index.html. LilyPond also provides its own Extending Manual: https://lilypond.org/doc/v2.24/Documentation/extending/index

### 14.1.2 Step 1b: Get Used to Prefix Notation

Scheme, like its Lisp relatives, uses prefix notation (Cambridge Polish Notation). Here are examples:

```
(+ 12 34)
>> This expression results in the value of 46.

(+ 4 (* 3 9))
>> This expression first resolves the multiplication: (+ 4 27), which is 31.
```

If you are new to this, I recommend starting with Daniel P. Friedman and Matthias Felleisen, *The little Schemer (4th ed.)* (Cambridge, MA, USA: MIT Press, 1996), ISBN: 0262560992. While you might be eager to dive into using Scheme in LilyPond, learning Scheme as a programming language will make the process smoother.[4]

### 14.1.3 Step 2: Study Lots of Snippets

Once familiar with Scheme, study how it integrates with LilyPond by reviewing snippets from LSR. Start with shorter examples and analyze their structure. Here is an example snippet for adding the *Schleifer* ornament:[5] The corresponding code:[6]

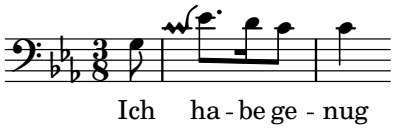



**Figure 14.1:** LSR No. 1185: *Schleifer* Ornament.

```
1  % Implementation by Martin Straeten of the Schleifer ornament
2  % as used by Johann Sebastian Bach, contributed to the user
3  % mailing list. In this case, it functions like a set of (always?)
4  % two grace notes, hence using a modified grace note to represent
5  % it in LilyPond makes sense.
6  %
7  % Code styling and user interface by Simon Albrecht 2024.
8
9  schleiferMarkup = \markup {
10   \large \halign #.2 \raise #0.0
```

4. Liska and Samra's resources serve as excellent refreshers later on.

5. https://lsr.di.unimi.it/LSR/Item?id=1185

6. The mailing list thread referenced in the preamble is available at https://lists.gnu.org/archive/html/lilypond-user/2021-09/msg00352.html

```
11    \combine
12    \halign #.8 \musicglyph "scripts.prall"
13    \rotate #140 \normalsize \raise #2.4 \musicglyph "flags.u3"
14  }
15  schleiferGrace =
16  #(define-music-function (note) (ly:music?)
17     #{
18       \grace {
19         \once\override NoteHead.stencil = #ly:text-interface::print
20         \once\override NoteHead.X-extent = #'(-2 . -0)
21         \once\override NoteHead.text = \schleiferMarkup
22         \once\omit Stem
23         \once\omit Flag
24         $note
25       }
26     #})
27
28  \relative {
29    \time 3/8
30    \partial 8
31    \clef bass
32    \key c \minor
33    g8
34    \schleiferGrace c es8. d16 c8
35    c4
36  }
37  \addlyrics {
38    Ich ha -- be ge -- nug
39  }
```

The `\schleiferGrace` variable creates a customized ornament using Scheme's `define-music-function` macro. For a deeper understanding of the macro syntax, refer to the *LilyPond – Internals Reference*.[7]

Taking the variable `\schleiferGrace`, we see that invoking it creates an instance of activating a Scheme function that starts at Line 16. `define-music-function` is a macro that allows you to create a function that operates on LilyPond.

According to *LilyPond – Internals Reference*, the syntax for `define-music-function` is:

7. https://lilypond.org/doc/v2.24/Documentation/internals/scheme-functions

```
(define-music-function (arg1 arg2 …)
                       (type1? type2? …)
    function-body)
```

In the code, the argument's name is `note`, and it is tested according to the type specified in `type1?`, which in this case is `ly:music?`. According to the *Internal Reference*, `ly:music?` is a function that checks whether the object—in this case, `note`—is a `Music` object. Thus, it becomes clear that this function will not work unless it is followed by a musical note.

From Line 17 to Line 26, we see that a LilyPond code snippet has been inserted, as `#{` and `#}` signify the boundary of the LilyPond code within the Scheme code. This means that as part of invoking the variable `\schleiferGrace`, it passes through this LilyPond fragment, which is responsible for creating a grace note. Here, the notehead of the grace note is replaced with `\schleiferMarkup`, which is defined in Lines 9 to 14 of the code.[8]

Lines 22 and 23 show that the stem and flag are omitted from the grace note, while Line 24's `$note` signifies that the original argument `note` is called upon.[9] In this way, the *Schleifer* ornament is created from a note that follows the variable `\schleiferGrace`. This note is transformed into a grace note with a customized stencil setting, all done within the Scheme code.

### 14.1.4 Step 3: Hack the Codes

Once you study a code and become familiar with how it operates, experimenting with the code by hacking is a good way to deepen your understanding. Below, I give one example using the preceding *Schleifer* ornament snippet.

The *LilyPond – Internal Reference* reveals that the object `NoteHead` has its own standard settings, as well as support for about a dozen other interfaces.[10] One of them is the `grob-interface`, which makes it possible to change the color of a graphical object, or *Grob*.[11] Further reading in the *LilyPond – Notation Reference* shows that it is possible to override the color of an object.[12] Let us now tweak the *Schleifer* ornament code to allow us to change the ornament's color.

Following the reference, add the following line underneath `\once\override NoteHead.X-extent`:

8. The technique of sequential overrides, invoking the Scheme command `#ly:text-interface::print`, sets the `.stencil` of the notehead to use whatever is defined in the `.text` parameter. This technique is frequently used and is very useful in customizing notation. See also: https://lilypond.org/doc/v2.24/Documentation/notation/modifying-stencils.

9. Refer to this page for the difference between `#` and `$`: https://lilypond.org/doc/v2.24/Documentation/extending/lilypond-scheme-syntax

10. https://lilypond.org/doc/v2.24/Documentation/internals/notehead

11. https://lilypond.org/doc/v2.24/Documentation/internals/grob_002dinterface

12. https://lilypond.org/doc/v2.24/Documentation/notation/inside-the-staff#coloring-objects

```
\once\override NoteHead.color = #red
```

Running LilyPond now should produce the following result: Hard-coding a change like this

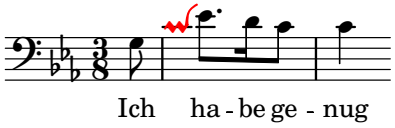



**Figure 14.2:** LSR No. 1185: *Schleifer* Ornament in red.

may be good for testing the waters, but we may want the *Schleifer* ornament in more than just one color. The beauty of extending LilyPond is that we can customize the Scheme code to allow for this flexibility.

Let us move on. We should now let the `define-music-function` know that we are adding an additional argument to specify the color. The first part of the code will look like this:

```
#(define-music-function (note schleiferColor) (ly:music? color?)
```

This adds the argument `schleiferColor`, which only accepts color, as indicated by the corresponding test function `color?`.

Then, implement this argument in the sequence of `\once\override` processes. The line `NoteHead.color` can now be changed to:

```
\once\override NoteHead.color = #schleiferColor
```

Now, the variable `\schleiferGrace` requires one more argument to specify the ornament's color. The entire code should look like this:

```
schleiferMarkup = \markup {
  \large \halign #.2 \raise #0.0
  \combine
  \halign #.8 \musicglyph "scripts.prall"
  \rotate #140 \normalsize \raise #2.4 \musicglyph "flags.u3"
}

schleiferGrace =
#(define-music-function (note schleiferColor) (ly:music? color?)
   #{
     \grace {
       \once\override NoteHead.stencil = #ly:text-interface::print
       \once\override NoteHead.X-extent = #'(-2 . 0)
       \once\override NoteHead.color = #schleiferColor
```

```
15          \once\override NoteHead.text = \schleiferMarkup
16          \once\omit Stem
17          \once\omit Flag
18          $note
19        }
20      #})
21  \relative {
22    \time 3/8
23    \partial 8
24    \clef bass
25    \key c \minor
26    g8
27    \schleiferGrace c #green es8. d16 c8
28    c4
29  }
30  \addlyrics {
31    Ich ha -- be ge -- nug
32  }
```

This produces the following output:

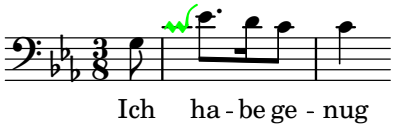



**Figure 14.3:** LSR No. 1185: *Schleifer* Ornament in green.

Notice that on Line 27, `#green` has been added. You can change this to any of the colors listed under "Normal Colors" in the *Notation Reference*,[13] such as `#'"lightsalmon"`, `#(x11-color "medium turquoise")`, or even `#'"#5e45ad"`.

As an exercise, try replicating the following excerpt:[14]

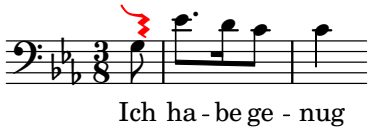



**Figure 14.4:** Can you replicate this?

**Table of Contents**

13. https://lilypond.org/doc/v2.24/Documentation/notation/list-of-colors
14. See LSR1185e3.ly for the answer.

## 14.2 Example 1: Creating a Time Signature with Its Compound Meter Form

On January 1, 2025, I came across a post by an anonymous user on Facebook.[15] The post asked if it would be possible to create a time signature that had its beat structure expressed in the form of a compound meter. Something like this:

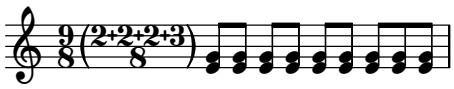


**Figure 14.5:** What the anonymous user wanted to achieve.

I responded to the post with relevant email threads on `lilypond-user` mailing list. I commented that it would be possible to override `TimeSignature.stencil` with custom-made time signatures. Incidentally, I was making a series of Fractional Time Signatures, which used this method.

The code for the aforementioned example is as follows:

```
\version "2.24.4"

{
  \clef "G"
  \time 9/8
  \set beatStructure = #'(2 2 2 3)
  \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
  \once \override Timing.TimeSignature.text = \markup
  {
    \override #'(baseline-skip . 0)
    \center-column \number {9 8}
    \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
    \hspace #-0.75
    \override #'(baseline-skip . 0)
    \center-column \number {{2+2+2+3} 8}
    \hspace #-0.75
    \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
    \hspace #-1
  }
  \repeat unfold 9 {<e' g'>8 }
}
```

15. https://www.facebook.com/groups/gnulilypond/posts/10162467719483529/

I realized that, while this might be an acceptable method if such time signatures appeared only once or twice in a piece, it may become problematic if I had to copy and paste this code every time I have such a time signature. Normally this could easily be resolved by making a variable out of `\override` clauses; however, a piece of music may use time signatures of this form in different configurations, just as the following example:

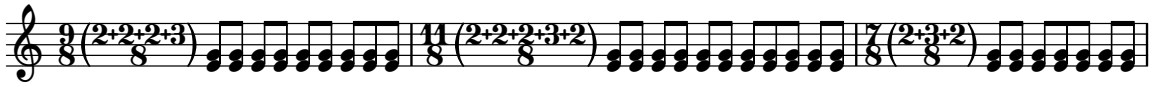


**Figure 14.6:** More compound meters.

The code:

```
\version "2.24.4"

{
  \clef "G"
  \time 9/8
  \set beatStructure = #'(2 2 2 3)
  \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
  \once \override Timing.TimeSignature.text = \markup
  {
    \override #'(baseline-skip . 0)
    \center-column \number {9 8}
    \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
    \hspace #-0.75
    \override #'(baseline-skip . 0)
    \center-column \number {{2+2+2+3} 8}
    \hspace #-0.75
    \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
    \hspace #-1
  }
  \repeat unfold 9 {<e' g'>8}

  \time 11/8
  \set beatStructure = #'(2 2 2 3 2)
  \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
  \once \override Timing.TimeSignature.text = \markup
  {
    \override #'(baseline-skip . 0)
    \center-column \number {11 8}
```

```
      \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
      \hspace #-0.75
      \override #'(baseline-skip . 0)
      \center-column \number {{2+2+2+3+2} 8}
      \hspace #-0.75
      \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
      \hspace #-1
    }
    \repeat unfold 11 {<e' g'>8 }

    \time 7/8
    \set beatStructure = #'(2 3 2)
    \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
    \once \override Timing.TimeSignature.text = \markup
    {
      \override #'(baseline-skip . 0)
      \center-column \number {7 8}
      \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
      \hspace #-0.75
      \override #'(baseline-skip . 0)
      \center-column \number {{2+3+2} 8}
      \hspace #-0.75
      \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
      \hspace #-1
    }
    \repeat unfold 7 {<e' g'>8 }

}
```

Writing as long of a code as this (for just three measures!) would be cumbersome, indeed. What could help is to come up with a music function, using the Scheme.

### 14.2.1 Step 1: Analyze What Could Be Automatized

I quote the code for the first example of this section again. This time, however, I turn the variables that could change each time I create an instance of this kind of time signature, into red:

```
\version "2.24.4"

{
  \clef "G"
```

```
    \time 9/8
    \set beatStructure = #'(2 2 2 3)
    \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
    \once \override Timing.TimeSignature.text = \markup
    {
      \override #'(baseline-skip . 0)
      \center-column \number {9 8}
      \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
      \hspace #-0.75
      \override #'(baseline-skip . 0)
      \center-column \number {{2+2+2+3}{8}}
      \hspace #-0.75
      \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
      \hspace #-1
    }
    \repeat unfold 9 {<e' g'>8 }
}
```

### 14.2.2 Step 2: Write the Code

It would be good if this function could accept the following as arguments:

- Time signature of the measure as defined normally in the LilyPond function `\time`. For this, I will set `timesig` as the name of the argument, that tests its value with `fraction?`.
- The customized stencil of the time signature. I need to declare how it looks, namely:
  - Overall time signature;
  - Numerator portion of the compound meter, and;
  - Denominator portion of the compound meter.

  It should look similar to how the LilyPond function `\compoundMeter` that accepts a list of lists.For this, I will set `beatstruct` as the name of the argument, that tests its value with `list?`.

I will now build the rest of the function. Notice the way the Scheme code references various locations of a list, using `car`, `cadr`, and so on:

```
\version "2.24.4"

compoundTimeWithBeatStructure =
#(define-music-function (timesig beatstruct) (fraction? list?)
```

```
    #{
      \time #timesig
      \set beatStructure = #(cadr beatstruct)
      \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
      \once \override Timing.TimeSignature.text = \markup
      {
        \override #'(baseline-skip . 0)
        \center-column \number
        {
          #(number->string (car (car beatstruct)))
          #(number->string (cadr (car beatstruct)))
        }
        \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
        \hspace #-0.75
        \override #'(baseline-skip . 0)
        \center-column \number
        {
          {#(string-join (map number->string (cadr beatstruct)) "+")}
          #(number->string (car (caddr beatstruct)))
        }
        \hspace #-0.75
        \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
        \hspace #-1
      }
    #}

    )

{
  \compoundTimeWithBeatStructure 9/8 #'((9 8)(2 2 2 3)(8))
  \repeat unfold 9 {<e' g'>8}
  \compoundTimeWithBeatStructure 11/8 #'((11 8)(2 2 2 3 2)(8))
  \repeat unfold 11 {<e' g'>8}
  \compoundTimeWithBeatStructure 7/8 #'((7 8)(2 3 2)(8))
  \repeat unfold 7 {<e' g'>8}
}
```

Thus, there is now a function called `\compoundTimeWithBeatStructure`, whose grammar is:

```
\compoundTimeWithBeatStructure
        TIME_SIGNATURE #'((TIME_SIGNATURE)(BEAT_STRUCTURE)(DENOMINATOR))
```

Running the code will result in the identical snippet as the previous figure:

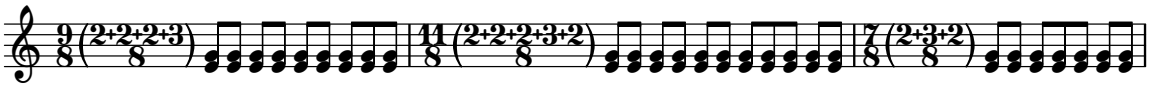


**Figure 14.7:** The same result as before with a shorter code.

**Table of Contents**

# Bibliography


Adès, Thomas. *Asyla : for large orchestra*. Faber Music, 1997.

Andre, Mark. *Contrapunctus : pour piano*. Paris: Durand, 2006.

———. *Un-fini I : 1995, für eine Harfenistin/einen Harfenisten (Harfe, Tam-tam, grosse Trommel)*. Neue Musik bei Carus. Stuttgart: Carus, 1997.

———. *Un-fini II : pour clavecin (1996)*. Paris: Editions Durand, 1998.

Beyer, Stefan. *Bleib hier. Schau zu. Mach kein Geräusch.* Manuscript, 2017.

———. *Marsch*. Manuscript, 2013-14.

———. *Mittel und Zwecke (Boulevard)*. Manuscript, 2014.

———. *Most of My Clients Come Back*. Manuscript, 2012-13.

Boulez, Pierre. *... explosante-fixe ... transitoire VII : (version 1991/93)*. Universal Edition, 1991.

———. *Sur incises : pour trois pianos, trois harpes et trois percussions-claviers (1996/1998)*. Universal Edition, 1998.

Czernowin, Chaya. *At the fringe of our gaze : for Orchestra and Concertino Group*. Schott, 2012/13.

———. *Lovesong : for mixed ensemble*. Schott, 2010.

———. *Streams (Slow Summer Stay I) : for 8 players*. Schott, 2012.

———. *String Quartet*. Schott, 1995.

Friedman, Daniel P., and Matthias Felleisen. *The little Schemer (4th ed.)* Cambridge, MA, USA: MIT Press, 1996. ISBN: 0262560992.

Gould, Elaine. *Behind bars : the definitive guide to music notation*. London: Faber Music, 2011. Book.

Levine, Carin, and Christina Mitropoulos-Bott. *The techniques of flute playing = Die Spieltechnik der Flöte*. Kassel ; New York: Bärenreiter, 2003.

Liska, Urs. *Understanding Scheme In LilyPond*. Vol. 2024. December 19. 2020. Web Page. https://scheme-book.readthedocs.io/en/latest/.

Onishi, Yoshiaki. *Gz II : for two accordions*. Brühl and Berlin: Edition Gravis, 2024.

Salzedo, Carlos. *L'étude moderne de la harpe... Modern study of the harp*. 3 p.l., 53 p. New York - Boston, G. Schirmer, 1921.

Samra, Jean Abou. *Extending LilyPond*. Vol. 2024. December 19. 2021. Web Page. https://extending-lilypond.gitlab.io/en/index.html.

Sparnaay, Harry. *The Bass Clarinet: A Personal History*. Periferia Sheet Music, 2012.

Takemitsu, Tōru. *Fantasma/cantos : for clarinet and orchestra*. Schott ; Schott Japan, 1993.

———. *Les yeux clos II : for piano*. Schott ; Schott Japan, 1990.

# Appendices

# Appendix A: Resources

As I taught LilyPond in a special topic course at the University of Delaware in Fall 2024, I compiled a list of links to useful websites and pages. It is in no way intended as a comprehensive list; instead, I list some essential pages that I have frequently looked up and found very useful. This page is subject to frequent revision.

**On LilyPond**

- Website: https://lilypond.org/
- Installing: https://lilypond.org/doc/v2.24/Documentation/learning/installing
- Manuals: https://lilypond.org/manuals.html

**Text Editor for LilyPond**

- Frescobaldi (Editor): https://frescobaldi.org/

**Coding LilyPond**

- Cheat Sheet: https://lilypond.org/doc/v2.24/Documentation/notation/cheat-sheet
- Snippets: https://lilypond.org/doc/v2.24/Documentation/web/snippets
- LilyPond Snippet Repository: https://lsr.di.unimi.it/

**Mailing List**

- Mailing list: https://lists.gnu.org/mailman/listinfo/lilypond-user
- Archives 1 https://lists.gnu.org/archive/html/lilypond-user/
- Archives 2 https://www.mail-archive.com/lilypond-user@gnu.org/

**Advanced Topic on LilyPond**

- LilyPond – Extending v2.24.4: https://lilypond.org/doc/v2.24/Documentation/extending/index#top

- Scheme (in LilyPond): https://scheme-book.readthedocs.io/en/latest/
- Extending LilyPond: https://extending-lilypond.gitlab.io/en/extending/index.html
- Scheme Resources https://www.gnu.org/software/guile/learn/#scheme-resources
- PostScript Manual: https://www.adobe.com/jp/print/postscript/pdfs/PLRM.pdf
- PostScript Tutorial: https://paulbourke.net/dataformats/postscript/

**Troubleshooting**

- The default text font for LilyPond doesn't seem to work (Mac)
- Frescobaldi freezes upon loading

**Miscellaneous Items**

- About Emmentaler font: https://lilypond.org/doc/v2.25/Documentation/notation/the-emmentaler-font

**Table of Contents**