

# SCORES LEVEL COMPOSITION BASED ON THE GUIDO MUSIC NOTATION

D. Fober, Y. Orlarey, S. Letz  
Grame - Centre National de Création Musicale  
{fober,orlarey,letz}@grame.fr

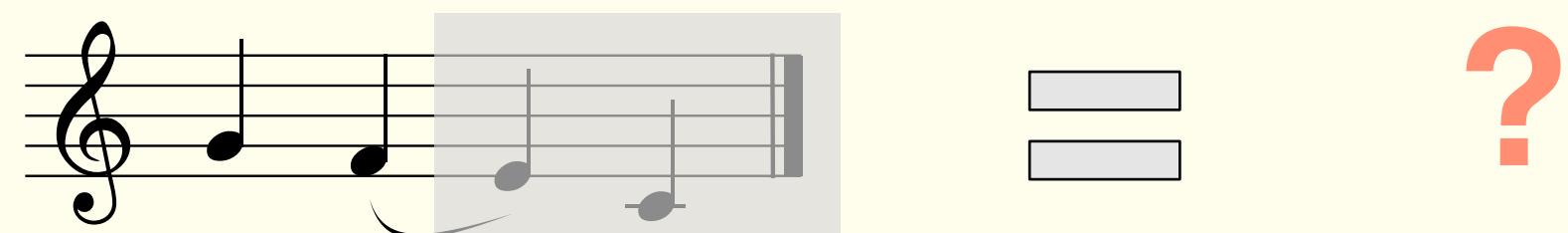
Based on the Guido Music Notation format, we have developed tools for music score "composition", i.e. operators that take scores both as target and arguments of high level transformations, applicable for example to the time domain (e.g. cutting the head or the tail of a score) or to the structural domains (e.g. putting scores in sequence or in parallel). Providing these operations at score level is particularly convenient to express music ideas and to compose these ideas in an homogeneous representation space.

## The problem

Scores level composition gives raise to a set of issues related to the music notation consistency. Applying operations at textual description level almost works syntactically but mostly produces undesired notations.

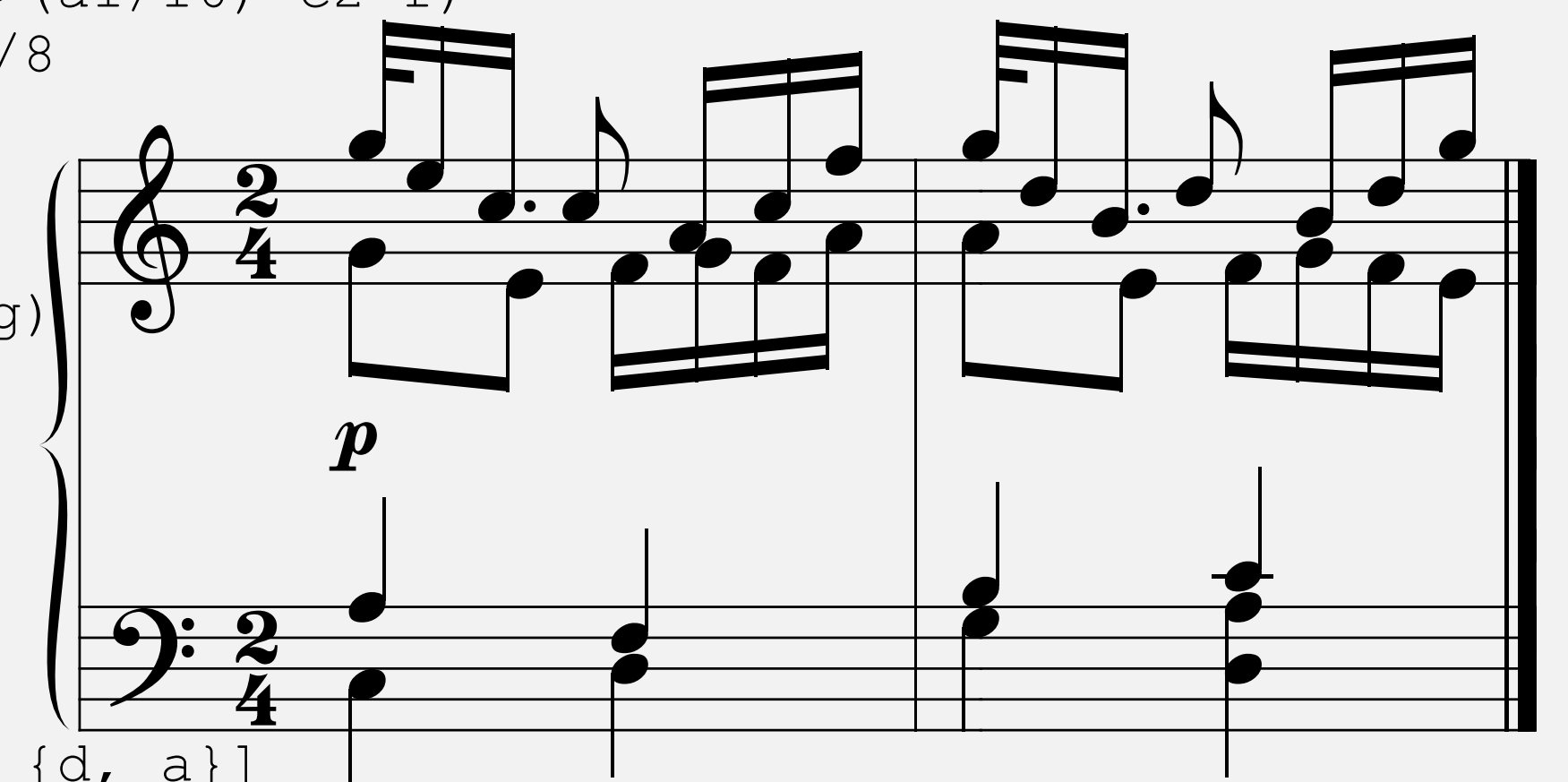
### Example 1: syntax error

Cut [g \slur(f e) c] = [g \slur(f ]



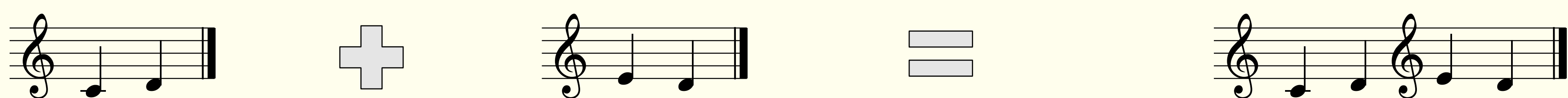
## The Guido Music Notation Format

```
{
[
  \pageFormat<w=21.5cm,h=15cm,tm=2mm,bm=2,lm=4,rm=2>
  \barFormat<"system"> \staff<1> \stemsUp \meter<"2/4">
  \intens<"p", dx=1hs,dy=-7hs>
  \beam(g2/32 e/16 c*3/32) c/8
  \beam(\noteFormat<dx=-0.9hs>(a1/16) c2 f)
  \beam(g/32 d/16 h1*3/32) d2/8
  \beam(h1/16 d2 g)],
[ \staff<1>\stemsDown g1/8 e
f/16 \noteFormat<dx=0.8hs>(g)
f a a/8 e
f/16 g f e],
[ \staff<2> \meter<"2/4">
\stemsUp a0 f h c],
[ \staff<2> \stemsDown c0 d g {d, a}]
}
```



### Example 2 : useless clef repetition

[ \clef<"g"> c d ] + [ \clef<"g"> e c ] = [ \clef<"g"> c d \clef<"g"> e c ]



## Proposed solution

### Based on elements time extend

time extent	description	sample
explicit	duration is explicit from the notation	slurs, cresc.
implicit	element lasts to the next similar element or to the end of the score	meter, dynamics, key
others	structure control	coda, da capo, repeats
-	formatting instructions	new line, new page
-	misc. notation elements	breath mark, bar

### Rule based operations

- **computing the beginning of a score:**
  - 1) the pending explicit time extent elements must be properly opened (i.e. opened-begin tags)
  - 2) the current implicit time extent elements must be recalled
- **computing the end of a score:**
  - 3) the explicit time extent elements must be properly closed (i.e. opened-end tags)
- **putting scores in sequence:**
  - 4) implicit time extent elements starting the second score must be skipped when they correspond to current existing elements
- **structure control:**
  - 5) computing the end of a score: every pending repeat end must be closed with a repeat end tag
  - 6) from successive unmatched repeat begin tags, only the first one must be retained

### Reversibility

- a new tag parameter : open = [begin | end | begin-end ]
- a new rule : *opened-end* followed by *opened-begin* are mutually cancelled

[ \atag<open="end">(f g) ] + [ \atag<open="begin">(f e) ]  
= [ \atag(f g f e) ]

### Terminology

- *opened-end* tags : [ g \slur (f ]
- *opened-begin* tags : [ e) c ]
- *pending repeat* : [ \repeatBegin a g ]

## Operations

### Composition

- putting scores in sequence
- putting scores in parallel using different alignments (left, right)

### Selection

- cutting parts in the time dimension (head, tail), using time based or event based specification
- cutting voices in the vertical dimension

### Transformation

- transposition
- time stretching
- pitch profile application
- rythm profile application

## Availability

As command line tools allowing series of transformations to be expressed as pipelining scores through operators e.g.

```
head s1 s2 | par s2 | transpose "[ c ]"
```

Integrated into GuidoCalculus, an application providing a graphic user interface and intuitive drag and drop features.

As source code: part of the guidolib project open source project (guidoar repository).

Links:

<http://guidolib.sf.net>  
<http://sourceforge.net/projects/guidolib/>

