brevity
A Compact Music Notation Format

James Tunnell

August 26, 2014

In a computer, music notation is usually represented by some file format. Music notation formats are often awkward or impossible to work with directly, and require a visualization layer that does not resemble the underlying representation. A simple file format for representation is presented that is straight-forward to both read and modify directly.

Contents

1 Introduction

Music notation formats generally have no natural visual resemblance to the physical musical notations they represent. This is unfortunate since for composers the visual arrangement of a composition conveys information about musical structure, since it shows parts together in parallel throughout time. So to allow computer-assisted composers to get the desired visual benefit, an extra visualization layer is provided, so that a digital score is presented in the end. But to gain the full benefit, a composer also needs to be able manipulate the digital score as if it is the very representation of the music. So in the end the extra layer must translate both directions between the extra presentation layer and the underlying representation layer.

The brevity format presented here strives to be as functional as a traditional file format (like MIDI or MusicXML) yet visually similar to a physical score. There is added complexity though, because in order to directly view and manipulate a file in this format one must be familiar with the particular syntax and symbols used to represent musical constructs. The following sections explain these particulars in detail.

2 Note and Rests

The basic units of a score are the notes and rests, because they define how time is broken up and what should happen during each time chunk. For a rest, of course, nothing should happen.

In brevity, notes and rests both define a duration, and a note additionally define a number of pitches to be played for the duration. So a rest is really just a note where no pitches are given to play. Of course, multiple pitches mean a note is really a chord. attached

2.1 Syntax Overview

The syntax for a single note or rest is shown below, where the number of pitches can be zero or more.

D: duration
P: pitch
L: link
A: accent

To make a sequence of notes/rests, simply string notes together, separated by spaces.

2.2 Durations

Notes and rests both need to declare some duration. Duration is represented by a rational number (i.e. a fraction), as in '3/4' or '2/3'. The numerator can be omitted, as in '/4', and then a 1 is assumed. This is convenient for quarter notes, eight notes, etc. Or, the denominator (and forward-slash) can be omitted, as in '1/' or '1' for a whole note.

2.3 Pitches

Every note can have zero or more pitches attached to it (of course, zero pitches would make the note a rest). Each pitch is given by two components: pitch class and octave. Pitch class is represented by a letter from `A` to `G` and an optional `#` (sharp) or `b` (flat). Following the pitch class, the octave is given, represented by a single decimal digit (`0` to `9`). For example: '`C4`', '`Eb2`', and '`F#5`'.

2.4 Accents

Accents control the duration of a note and the shape of note loudness (like the envelope of a signal). Accents are represented by a single character placed at the end of the note body, after all the pitches and links. Simply omit the accent character to place a note without any accent.

The supported accent types and the characters used to represent them are summarized in Table 1. A short description of the effect of each accent type follows:

- Staccato: separated from next note
- Staccatissimo: very separated from next note
- Accent (marcato): emphasize beginning and then taper off rather quickly
- Martellato (hammered): loud as an accent mark and short as a staccato
- Tenuto: Played at full length or longer. When under a slur, note is separated with a little space from surrounding notes

Accent type	**Marker**
Staccato	`.` period
Staccatissimo	`'` apostrophe
Accent (marcato)	`>` greater than
Martellato (hammered)	`^` caret
Tenuto (sustained)	`_` underscore

Table 1: Accent types and markers

As an example, some stocatto notes:

```
/4C5. /4D5. /4E5.
```

2.5 Relationships

Note relationships (or links) control the transition between sequential notes. A link is just between two notes, and won't have any other effect outside those notes. A link marker is placed just after a note pitch. Here is an example of a passage connected by slurs:

```
/4C4=C4 /16C4=E4 /16E4=F4 /4F4
```

Here is an example using slur/tie links:

```
/4C4=C4 /16C4=E4,F4=G4 /16E4,G4
```

Table 2 lists the supported link types and the characters used to represent them. Here is a description of each supported link type:

- Slur/tie: connects two pitches with no separation and no rearticulation. The same link marker is used for a slur or tie since they're effectively the same relationship.
- Legato: connects two pitches with no separation and with rearticulation.
- Glissando: connect two pitches by articulating all the pitches in between.
- Portamento: continuously glide between pitches, without rearticulation.

Link type	**Marker**
Slur/Tie	`=` equals
Legato	`-` minus
Glissando	`~` tilde
Portamento	`/` forward-slash

Table 2: Link types and markers

3 Sequences

A sequence is a series of notes/rests. A sequence is declared by placing notes/rests together and separating by spaces, as in

```
/4D6 /4A5 /16Ab6 3/16Bb6=Bb6 /4Bb6
```

A sequence can either be labeled for later reused, or modified to produce a new sequence.

3.1 Labeling Sequences

Sequences are assigned to a label by following the label with `:=` and then a sequence, as in

```
my_label:= /8C5 /8D5 /8E5 /8D5 /2C5
```

Once labeled, the label can be used in place of the sequence.

3.1.1 Continuing Labeled Sequences

A labeled sequence can be continued later, which will add more notes onto the end of the sequence. This might also be called concatenation.

A labeled sequence is continued by following the label with `:+` and then a sequence, as in

```
bass:= /8C4 /8C4 /8C4 /8C4
bass:+ /8E4 /8E4 /8E4 /8E4
```

This allows a sequence to span over multiple lines by starting each line with the same label. But continuing lines do not have to be consecutive, as long as the same label is used for each continuing line. So, multiple parts can be interleaved as follows:

```
bass:= /4C4 /4C4 /4C4 /4C4
lead:=  /2E6 /2E6

bass:+ /4E4 /4E4 /4E4 /4E4
lead:+  /2G6 /2G6
```

3.2 Modifying Sequences

Sequences can be modified, which results in a new sequence. If a sequence is not labeled, it needs to be grouped using parentheses before it can be modified, as in

```
( /4C5 /4D5 )
```

Sequences can be duplicated, transposed, stretched, and composed. Additionally, sequence modifications can be chained one right after the other.

3.2.1 Duplicating Sequences

A sequence can be replaced some number of duplicate sequences by following it with `:n`, where n indicates the number of duplicates.

For example, to replace a sequence with 2 duplicate sequences (i.e., repeat once),

`( /8C5 /8C#5 ):2` is equivalent to `/8C5 /8C#5 /8C5 /8C#5`

3.2.2 Transposing Sequences

A sequence can be transposed by appending a +/- and some number of semitones (half-steps) to transpose by. For example, to transpose up four semitones:

`( /4C5 /4D5 )+4` would yield `/4E5 /4F#5`

Or, to transpose down by four semitones:

`( /4C5 /4D5 )-4` would yield `/4Ab4 /4Bb4`

The result is the sequence transposed up/down some number of semitones.

3.2.3 Stretching Sequences

Sequences can be stretched in two ways: stretching to equal some given total duration, or stretching by multiplying all note durations by some factor.

Stretching to equal some duration

To stretch so that total note duration equals some given note duration. This is done by following a sequence with `:r`, where r is some note duration.

For example, to turn a sequence of two quarter notes into two half notes,

`( /4C5 /4D5 )=1` is equivalent to `/2C5 /2D5`

Or to make a half note triplet from three quarter notes,

`( /4A4 /4B4 /4C4 )=/2` is equivalent to `/6A4 /6B4 /6C4`

If the sequence contains notes of different durations, each note keeps the same ratio to total duration. For example,

`( /2Eb5 /4Bb5 3/4Bb5 )=1` is equivalent to `/3Eb5 /6Bb5 /2Bb5`

Stretching by multiplying note durations

A sequence can be stretched by multiplying the note durations by some multiplicative factor. This is done by following a sequence with `*r`, where r is some rational number, that can be formatted just like a note duration.

For example, to multiply durations by 2,

`( /4C5 /4D5 /2E5 )*2` is equivalet to `/2C5 /2D5 1E5`

3.2.4 Composing Sequences

Sequences can be composed together to create a new sequence. This is done implicitly just by following one sequence with more notes or sequences.

For example,

`(/4A5 /4C5):2 1Eb5` would be equivalent to `/4A5 /4C5 /4A5 /4C5 1Eb5`

Of course, labels can be used in places of sequences, as in

```
seq1:= /4F4 /4G4
seq2:= seq1:2 seq1+2
```

3.2.5 Chaining Modifications

Sequence modifications can be chained, so each modification is applied on the new sequence resulting from the previous modification. For example,

`(/4F4 /4G4)+2:2` is equivalent to `/4G4 /4A4 /4G4 /4A4`

4 Dynamics

Dynamics is concerned with absolute base loudness over an entire part. This is in contrast to note accents which affect the relative loudness shape of a single note. Dynamic levels are represented by markers which you might expect: `ppp`, `pp`, `p`, `mp`, `mf`, `f`, `ff`, and `fff`.

4.1 Immediate Changes

To immediately change dynamic level, simply place a dynamic marker by itself before a note. The change will take effect immediately at the start of the note. Here is an example of an immediate dynamic change:

```
p /8C6 /8C6 /4F6 f /8D6 /8D6 /4G6
```

4.2 Gradual Changes

Dynamic level can be gradually changed over time with a crescendo or decrescendo, the start of which is marked by a `<` or `>`, respectively. The end of a crescendo or decrescendo is marked by the target dynamic level. For example, `> ... p` marks a decrescendo from the current dynamic level down to piano over the course of whatver note/rests come between. Here is an example of a gradual dynamic change:

```
p< /8C6 /8C6 /4F6 f /8D6 /8D6 /4G6
```

5 Parts

Of course, a musical score wouldn't be complete without parts. If you've understood note sequences, then the good news is that a part is a note sequence! However, in the end, the final score needs have a name associated with each part. This is accomplished with a syntax like an associative array (hash table), as follows:

```
{"Bass": bass, "Lead Guitar":lead, "Rhythm Guitar":rhythm}
```