USING REGULAR EXPRESSIONS TO EXPRESS BOWING PATTERNS FOR STRING PLAYERS

Cordelia V. Hall, John T. O'Donnell

University of Glasgow
Department of Computing Science, Glasgow, UK
jtod@dcs.gla.ac.uk

ABSTRACT

The study of bowing is critically important for string players. Traditional bowing annotations are a valuable part of orchestral and individual documentation, but they do not help the performer to search a piece for other passages that should be bowed the same way, or to identify alternative bowing styles. We introduce a notation based on regular expressions that describes patterns of notes in the music, as well as the bowing to be applied to the pattern. These expressions support complex bowings, and not just single annotations without musical context. The notation is simpler than general tools for regular expressions used in some software, and is suitable for use by students and musicians. We have developed a music editor that implements the notation and edits documents in Lilypond. The approach has been evaluated by experimenting with the editor on six violin sonatas by Mozart. The experiments demonstrate that the regular expression notation is successful at finding passages and inserting the bowings; that the patterns occur a number of times; and the bowings can be inserted automatically and consistently.

1. INTRODUCTION

Creating bowings for string instruments takes hard work. It requires technical maturity and experience from the string player, so much so that when learning their instrument, many students are given bowings by their teachers and expected to work out their own later. This implies that there is a collection of bowings appearing in music commonly studied, but one that cannot be searched and consulted. It is likely that there is little in the way of archiving done for bowings developed by professionals either, yet their experience is worth a great deal as well.

In an information age, this is a sad state of affairs. It is not helped by the fact that many editors accept bowing notation, but then do not currently support search or other algorithms, so that the string player can represent but not really exploit bowing information.

We have been doing some experiments towards an editor that supports string players when learning chamber and orchestral music. The user can search for and possibly change bowings similar to a given one specified in one of the parts, and can look up what was done in well known editions.

To do this, we want to describe precisely a sequence or pattern of notes that will be bowed in a certain way. The entire piece (or even other pieces of music) can then be searched for occurrences of this pattern. There needs to be a precise notation for expressing such sequences, because computer algorithms will perform the search, and we also want it to be unambiguous and clear for a musician. The search will find a set of matches—places where the actual music fits the pattern.

We also want to insert a bowing into each occurrence of a match in a piece of music. Musicians often prefer to bow similar passages in a similar way, and it may also be necessary to edit the parts for different string sections in an ensemble so that they are using compatible bowings.

A further goal is to provide a notation that can be used to archive and search editions of music, including the bowings. The current situation is that bowings are inserted as graphic symbols in music notation, which is useful for the performer but does not support automated searches.

The bowing patterns must express the essential aspects of a passage (such as note durations) while omitting irrelevant aspects. A bowing pattern will apply to a sequence of notes, so the pattern must be able to express concatenation. Sometimes bowing patterns start or end in slightly different ways in various places; this means that the notation must be able to express a choice between alternatives. A regular bowing may occur several times within a passage, and it is even possible for the number of repetitions to vary in several places within the music. Therefore the notation must express a repetition of a smaller pattern.

The requirements discussed above are met by *regular expressions*, a mathematical technique developed to describe languages formally. Regular expressions can be used both to match music passages and to edit them.

2. REGULAR EXPRESSIONS AND BOWING PATTERNS

A regular expression takes a set of symbols and builds alternations, repetitions, and concatenations from them and other regular expressions.

The set of symbols we use for notes is be derived from a set of numbers giving note durations $\{1,2,4,8,16,32\}$. Dots can be added as needed to these durations to express dotted durations as well. We then add bowing directions from $\{u,d\}$ to elements of the set of note durations, forming the set of note representations $\{1u,1d,1.u,...,2u,2d,...\}$. We also need a set of articulation symbols. There are articulation symbols such as $\langle$ and $\rangle$, which denote articulation slurs, (and) which are ordinary slurs, '', which denotes a staccato mark, and '-', which denotes a legato mark. Finally, there is '*', which denotes an empty string, and is used to indicate a point at which a symbol will be inserted during an edit. The full set of symbols is:

$$\{1u,1d,2u,2d,4u,4d,8u,8d,16u,16d \\ 32u,32d,...,\langle,\rangle,(,),',-,*\}$$

All of the elements of this set, except for '*', match only themselves, while the '*' matches the empty string.

A *bowing pattern* is either

- a concatenation, written as $p1\ p2$ where $p1$ and $p2$ are symbols and/or bowing patterns,
- an alternation, written as $\{p1 \mid p2\}$, where $p1$ and $p2$ are symbols and/or bowing patterns,
- or a repetition, written as $[p]$, where p is either a symbol or a bowing pattern.

For example, the concatenation $16u\ 8d$ represents a semiquaver upbow followed by a quaver downbow; the alternation $\{16u \mid 8d\}$ represents either a semiquaver upbow or a quaver downbow, and the repetition $[8d]$ represents a sequence of quaver downbows.

3. USING BOWING PATTERNS TO EDIT MUSIC

Our software allows the user to see a Lilypond file, edit it, and then look at the typeset. Lilypond is a text-based music notation language, so it can provide some of the benefits of both text and graphics. It generates an excellent score, and because it is text-based, an editor can provide searching and other algorithms to a user. A hybrid of this form offers a great deal to a string player who wants to edit music in an appropriate form, but would also like to be able to consult bowings of other experts, or look at and edit other parts in chamber music.

Figure 1 shows the results of an experiment using the editor, applying Schradieck's bowings to several movements from Mozart's piano and violin sonatas. Notice that the edits in these cases occur at least twice and some cases as many as 7 times. That means that the editor saved time and made consistent edits on multiple instances of the bowings. In addition, these search and edit patterns may be saved, allowing the user to search for them in other editions of the same music and research their own bowing decisions.

As an example of how the bowing patterns work, we look at the second bowing inserted on the first line of Figure 2 (at the end of this paper). The annotated music has a dotted crochet (C) followed by a quaver (G) and four quavers following (B,A,G,F).

The bowing pattern used to insert this bowing is $\{2d \mid (4.d\ 8d)\} * 8u\ 8d\ 8u\ 8d*$. Because the pattern occurs elsewhere in the piece with a different initial note, it starts with an alternation, in which the second arm matches the dotted crochet (C) and quaver (G). The quavers afterwards, which match B,A,G,F, are surrounded by '*', indicating that new bowing symbols will be inserted there.

The bowing pattern used to edit this bowing is $\{2d \mid (4.d\ 8d)\}(8u\ 8u\ 8u\ 8u)$. Notice that while nothing is different about the initial alternation, the '*'s have been replaced with parentheses, indicating that they are slurred, and all of the last four quavers are upbows.

Like this pattern, most of the matching bowing patterns are simple concatenations with alternations. However, repetitions had an interesting role to play too. In KV 301, movement 2, the matching bowing pattern has a repetition, within which there is a concatenation followed by an alternation. When matching, it suggests a while loop in which the concatenation within the repeat is matched several times, followed by one of the alternatives of the alternation. When the concatenation is matched for the last time, the other alternative of the alternation is matched and the match of the bowing pattern succeeds. Similarly, in KV 304, movement 1, the matching pattern is a concatenation, followed by a repetition, followed by a concatenation. The repetition captures the repeated figure in the theme of the movement.

4. RESEARCH CONTEXT

Individual violin bow strokes have been studied from audio data and catalogued in databases [5] [7]. It has also been recognised that the annotations produced by musicians as they work are essential to their profession [6], and so bowing annotations have been supported in digital music stands such as muse [2] and eStand [1], although searching for bowing annotations appears not to be supported as yet.

Humdrum [3] supports accessing representations of music with regular expressions, although these systems are thought to be of use to music researchers only.

Our work also uses regular expressions to address a practical problem for for students and performing musicians. It enhances a music editor with the capability of searching for

search	edit	loc
{2*d* | (2.*d* 4*d*)} ∗(16*u* 8.*u*) (16*d* 8.*d*) {(16*u* 8.*u*)∗ | ∗}	{2*d* | (2.*d* 4*d*)} ⟨(16*u* 8.*u*) (16*d* 8.*d*) {(16*u* 8.*u*)⟩|⟩}	296,3 7
{2*d* | (4.*d* 8*d*)} ∗8*u* 8*d* 8*u* 8*d*∗	{2*d* | (4.*d* 8*d*)} (8*u* 8*u* 8*u* 8*u*)	296,3 5
8*d* ∗ ∗16*u* 16*d* 8*u* ∗ ∗16*d* 16*u* {4*d*∗ | 8*d*∗}	8*d*′ (16*u* 16*u* 8*u*)(16*d* 16*d* {4*d*) | 8*d*)}	301,1 6
∗4.*d* 16*u* 16*d*∗ ∗8*u* ∗ 8*d* ∗ ∗ ∗16*u* 16*d* 16*u* 16*d*∗ ∗4.*u* 16*d* 16*u*∗ 8*d* 8*u*	(4.*d* 16*d* 16*d*) (8*u*′ 8*u*′) (16*d* 16*d* 16*d* 16*d*) (4.*u* 16*u* 16*u*) 8*d* 8*u*	301,1 2
[8*d* ∗ 8*u* ∗ 8*d* ∗ ∗ (8*u* 8*u*) {∗16*d* 16*u*∗ | 8*d*}]	[8*d* (8*u*′ 8*u*′) (8*d* 8*d*) {(16*u* 16*u*) | 8*d*}]	301,2 2
16*u* 16*d* 16*u* 16*d* 16*u* 16*d* 16*u* (16*d* 16*d*) ∗16*u* ∗ 16*d* ∗ ∗ (16*u* 16*u*) ∗16*d* ∗ 16*u* ∗ ∗	16*u* 16*d* 16*u* 16*d* 16*u* 16*d* 16*u* (16*d* 16*d*) (16*u*′ 16*u*′) (16*d* 16*d*) (16*u*′ 16*u*′)	302,2 5
4*d* 8*u* ∗ 8*d*∗ (8*u* 8*u* 8*u* 8*u*)	4*d* 8*u* − 8*d* − (8*u* 8*u* 8*u* 8*u*)	302,2 4
{(4.*d* 16*d* 16*d*) ∗8*u* ∗ 8*d* ∗ ∗ | (4.*u* 16*u* 16*u*) ∗8*d* ∗ 8*u* ∗ ∗}	{(4.*d* 16*d* 16*d*) (8*u*′ 8*u*′) | (4.*d* 16*d* 16*d*) (8*u*′ 8*u*′)}	303,2 5
8*d* 8*u* 2*d* 2*u* 2.*d* 8*u* 8*d* [∗ ∗ 2*u* 4*d* ∗ 4*u*∗ ∗2.*d* 8*u* 8*d*] 4*u* 4*d* ∗4*u* ∗ 4*d* ∗ ∗	8*d* 8*u* 2*d* 2*u* 2.*d* 8*u* 8*d* [⟨(2*u* 4*u*) 4*u*′ ⟩2.*d* 8*u* 8*d*] 4*u* 4*d* (4*u*′ 4*u*′)	304,1 4
(8.*d* 32*d* 32*d*∗ ∗8.*d* 32*d* 32*d*∗ ∗8.*d* 32*d* 32*d*)	(8.*d* 32*d* 32*d*) (8.*u* 32*u* 32*u*) (8.*d* 32*d* 32*d*)	306,2 2

Figure 1. Bowing patterns inserted by the editor. We translated six movements from Mozart's piano and violin sonatas into Lilypond, and looked at an edition bowed by Schradieck for information on bowings to be changed and inserted. For each of the bowing patterns in the column labelled *search*, we used the editor to insert the corresponding *edit* pattern on the same line. The *loc* column gives the Köchel number, movement number, and, on the next line, the number of occurrences of an edit.

and modifying complex bowing patterns, not just individual annotations without context. We provide two levels of notation: one which is suited to the performer (the notation that appears in Figure 1), and a more complex underlying notation needed by the software to interact with Lilypond notation.

5. CONCLUSION

We have developed a music editor that implements simple regular expression notation and edits documents in Lilypond. The approach has been evaluated by experimenting with the editor on six violin sonatas by Mozart. The experiments demonstrate that the regular expression notation is successful at finding passages and inserting the bowings. They also show that the patterns occur a number of times, and the bowings can be inserted automatically and consistently (thus the use of patterns is justified).

Students may benefit particularly from this approach, as it supports the systematic study and comparison of bowings in various editions edited by well known musicians.

While Lilypond is well suited to the problem of expressing bowing patterns, being text based, there is no reason in principle that other text based or even WYSIWYG editors cannot make bowing patterns available to users.

6. REFERENCES

[1] J. Cross, "estand(tm) electronic music stand," in *Notes*, March 2004.

[2] C. Graefe, D. Wahila, J. Maguire, and O. Dasna, "muse: a digital music stand for symphony musicians," in *ACM interactions, Vol. 3, Issue 3*, May/June 1996, pp. 26–35.

[3] D. Huron, "The humdrum toolkit reference manual," in *Center for Computer Assisted Research in the Humanities*, 1994, pp. 1–551.

[4] W. Mozart, *Eighteen Sonatas For Piano and Violin*, H. Schradieck, Ed. G. Schirmer, 1934.

[5] N. Rasamimanana, E. Flety, and F. Bevilacqua, "Gesture analysis of violin bow strokes," in *6th International Gesture Workshop*, 2005, pp. 145–155.

[6] M. Winget, "Heroic frogs save the bow: Performing musician's annotation and interaction behavior with written music," in *7th International Conference on Music Information Retrieval*, 2006.

[7] D. Young and A. Deshmane, "Bowstroke database: A web-accessible archive of violin bowing data'," in *NIME*, 2007.

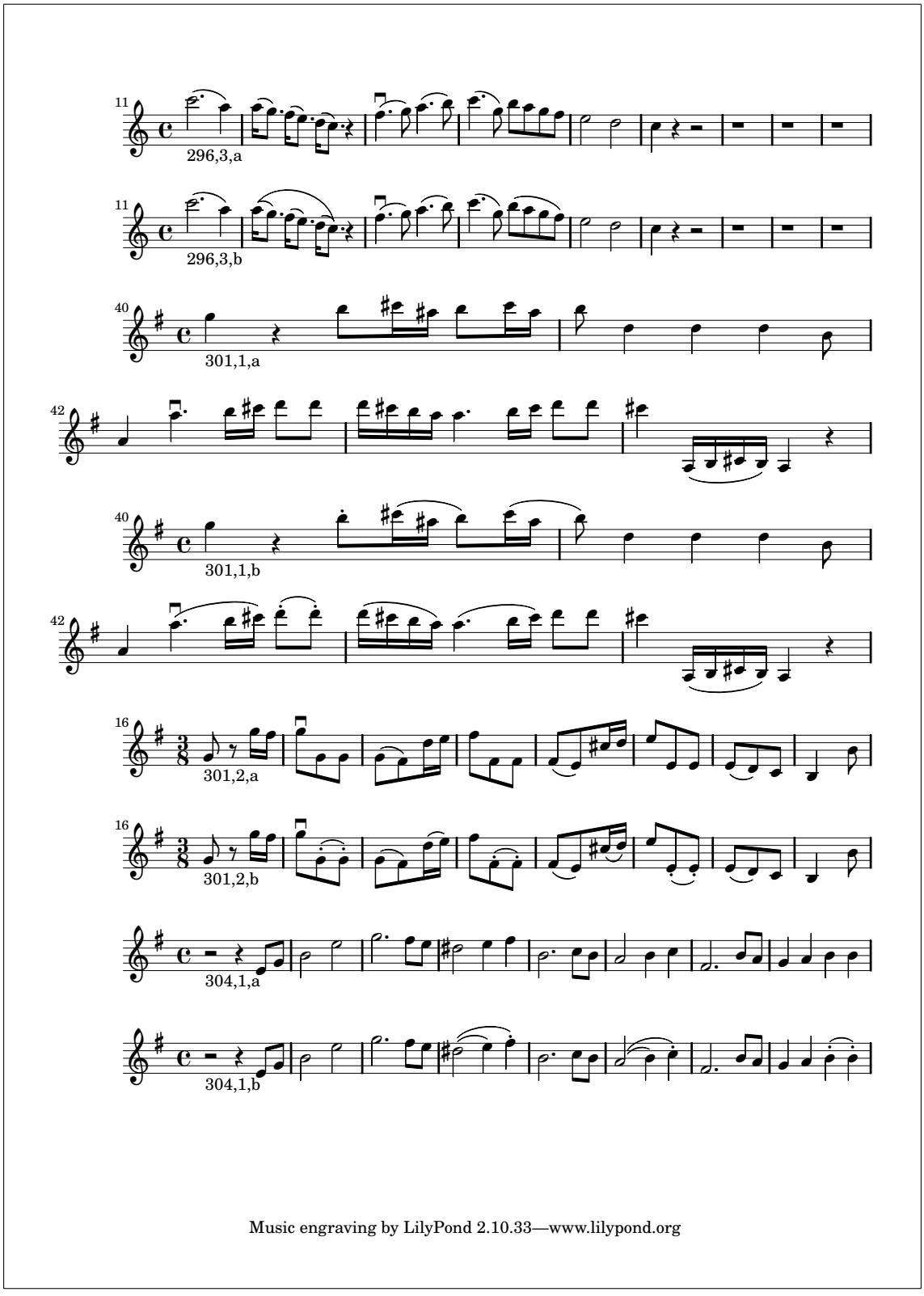

Figure 2. Part of the Lilypond output generated by the editor for the pieces referred to in Figure 1. The versions labelled 'a' are the original Mozart; the versions labelled 'b' are generated by the editor using the bowings of Schradieck [4].