Temporal Patterns in Polyphony

Mathieu Bergeron and Darrell Conklin

Department of Computing
City University London
{bergeron,conklin}@soi.city.ac.uk

Abstract. This paper formally characterizes the expressiveness of three approaches for polyphonic pattern representation and matching: $\mathcal{R}$ (relational patterns); $\mathcal{H}$ (Humdrum); and $\mathcal{SPP}$ (Structured Polyphonic Patterns). Relational networks have the highest expressiveness but $\mathcal{H}$ and $\mathcal{SPP}$ admit faster matching algorithms. It is shown how $\mathcal{H}$ and $\mathcal{SPP}$ can be cast as different restrictions of $\mathcal{R}$, both providing an expressive subset of full relational networks. In addition, the intersection of $\mathcal{H}$ and $\mathcal{SPP}$ yields yet another language: $\mathcal{SPP}_{\texttt{seq}}$, a restriction of $\mathcal{SPP}$ based on sequences of layered components. This new language is expressive enough to capture basic polyphonic patterns such as suspensions and parallel fifths and may be a new, more efficient approach to pattern extraction. The formal arguments contained in this paper are illustrated with musical examples extracted from J.S. Bach chorale harmonizations.

1 Motivation

Polyphony forms a large part of the western musical heritage and its essence — having multiple concurrent streams of musical events (with the temporal relations this implies) — is encountered in most kinds of modern music. However, there are few computational approaches for the expression and efficient matching of polyphonic patterns. This paper formally compares the expressiveness of three such languages and proposes a new one, establishing the hierarchy of Figure 1. To facilitate this presentation, arguments are restricted to patterns containing only two voices; results may however be generalized to denser polyphonic textures.

As a motivating example, consider the two-voice suspension of Figure 2. This typical polyphonic pattern is expressed in Figure 3 in the languages $\mathcal{R}$ (relational patterns); Humdrum; and $\mathcal{SPP}$ (Structured Polyphonic Patterns). As illustrated by the $\mathcal{R}$ expression (Figure 3i), even this simple pattern requires sophistication: variables to be instantiated by three events; inequality statements ensuring that the mapping from variables to events is injective; temporal relations between events (discussed below); and pitch relations such as consonance and dissonance.

This paper restricts its attention to the following binary temporal relations: i) $\texttt{m}(a, b)$ (a **m**eets b: a finishes when b begins), ii) the symmetric $\texttt{st}(a, b)$ (a and b **s**tart **t**ogether), iii) $\texttt{sw}(a, b)$ (a **s**tarts **w**hile b is sounding) and iv) the symmetric $\texttt{ov}(a, b)$ (a and b **ov**erlap: they sound together as some point in time). Figure 4 restates the relations in the notation of Allen [1] and Figure 2ii illustrates their musical relevance.

E. Chew, A. Childs, and C.-H. Chuan (Eds.): MCM 2009, CCIS 38, pp. 32–42, 2009.

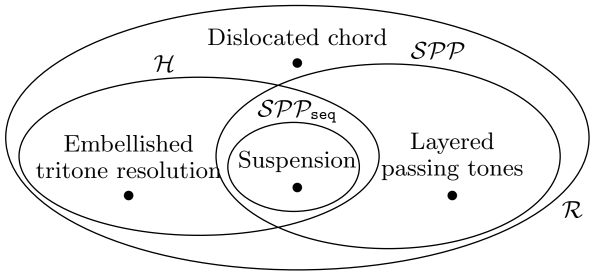


Fig. 1. Expressiveness of four polyphonic pattern languages: $\mathcal{R}$ (relational), $\mathcal{H}$ (Humdrum), $\mathcal{SPP}$ (Structured Polyphonic Patterns) and $\mathcal{SPP}_{\texttt{seq}}$ ($\mathcal{SPP}$ restricted to sequences of layered components)

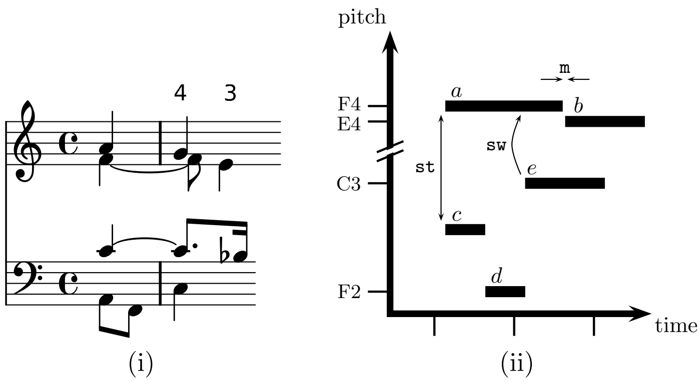


Fig. 2. (i) A 4-3 suspension between bass and alto voices in bars 16-17 of Bach's chorale BWV 283 and (ii) A piano-roll representation of the alto and bass voices

The Humdrum toolkit is widely-used for pattern matching in symbolic music data. Although Humdrum supports polyphony, it can be difficult to use for even simple patterns [5,2]. For example, Figure 3ii shows a Humdrum suspension pattern expressed with regular expressions. These typically do two things : i) match the beginning of events, the continuation of events or possibly no event at all (a "don't care" option) and ii) match features of those events by matching corresponding values in additional columns (this is the purpose of the "t" tokens at the end of the each lines, the first one matching a consonance feature and the second a dissonance feature).

In [4], the difficulties of Humdrum are circumvented by implementing a Prolog query that extracts all parallel fifths occurring in a corpus of J.S. Bach chorale harmonizations. The approach requires expert Prolog programming knowledge however, and even a slight reordering of Prolog clauses may have dramatic effects on pattern matching tractability. In general, the relational matching problem is

(i)

event(a)	voice(a, x)	m(a, b)
event(b)	voice(b, x)	sw(e, a)
event(e)	voice(e, y)	sw(b, e)
$a \neq b$	$x \neq y$	diss(a, e)
$a \neq e$		cons(b, e)
$b \neq e$		

(ii)

```
[a-g A-G]+[- # n]*[^)]*[(]   [a-g A-G]+[- # n]*.*.*t$
[a-g A-G]+[- # n]*[)]   [^(]*[a-g A-G]+[- # n]*.*t$
```

(iii)

$$\frac{-\{\}_x}{-\{\}_y} \; ; \; \frac{\{\texttt{sw\_cons}(y) : \texttt{t}\}_x}{-\{\texttt{sw\_diss}(x) : \texttt{t}\}_y}$$

Fig. 3. A suspension pattern using (i) relations; (ii) Humdrum and (iii) $\mathcal{SPP}$

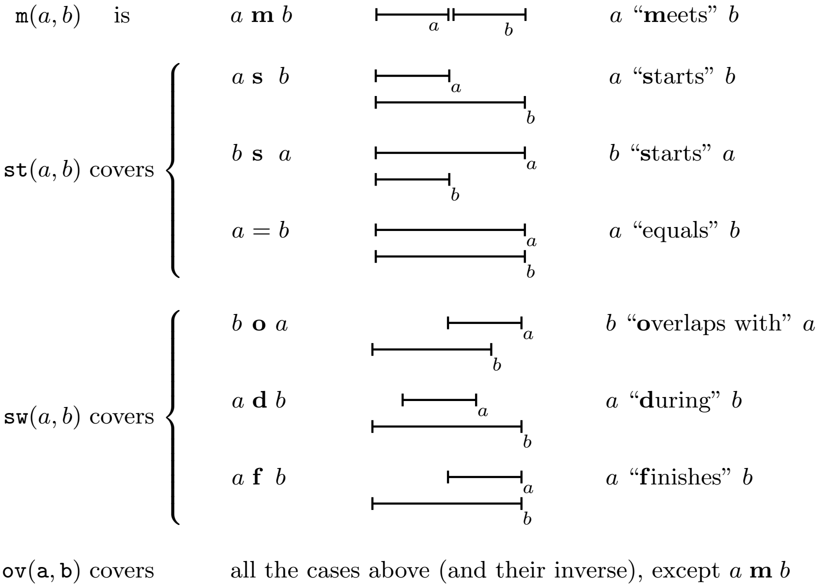


Fig. 4. The temporal relation analyzed in this paper expressed in the notation of Allen [1]

a subgraph isomorphism problem, which is known to be NP-complete. Moreover, not all networks are satisfiable and it is also NP-complete to determine satisfiability. Hence, it is not practical to base a pattern extraction approach on relational patterns. In [2], the latter results are replicated with $\mathcal{SPP}$, an abstract polyphonic pattern representation based on sequencing and layering operators. In Figure 3iii for example, the `sw_cons` and `sw_dis` pattern components are layered to form a temporal relation. This layering holds for events forming a

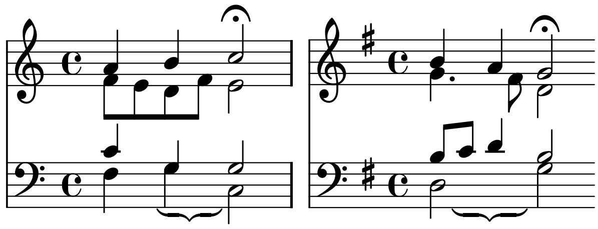


Fig. 5. Dislocated V^7 chords captured by *Pattern 1*: BWV 284 bar 15 (i) and BWV 318 bar 13 (ii)

consonance and dissonance with opposite voices, and satisfying the `sw` temporal relation. $\mathcal{SPP}$ is further elaborated in Section 4.

2 Relational Patterns

A relational pattern r is simply a set of temporal relations over event variables ε:

Definition 1. $$r \in \mathcal{R} ::= \omega, \ldots, \omega \quad \text{with} \quad \omega ::= \texttt{m}(\varepsilon,\varepsilon) \mid \texttt{ov}(\varepsilon,\varepsilon) \mid \texttt{st}(\varepsilon,\varepsilon) \mid \texttt{sw}(\varepsilon,\varepsilon)$$

With appropriate pitch relations, the pattern below could represent the "dislocated" V^7 chords shown in Figure 5. It enforces that chord tones eventually overlap with the root of the chord, but no other temporal relation is enforced:

Pattern 1. $$\texttt{ov}(a,b),\ \texttt{ov}(a,c),\ \texttt{ov}(a,d)$$

Alternatively, relational patterns can be represented as directed labelled graphs, where nodes represent event variables and edges represent relations (see Figure 6).

3 Humdrum

By contrast to $\mathcal{R}$, temporal relations in $\mathcal{H}$ are specified indirectly via a token matrix:

Definition 2. $$h \in \mathcal{H} ::= \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ & \vdots \end{bmatrix} \quad \text{with} \quad h_{ij} ::= \varepsilon \mid (\varepsilon) \mid \star$$

The token ε refers to the beginning of a new event; the token (ε) is the continuation of the preceding event and the token $\star$ is the special "don't care" symbol that enforces no temporal relation. Note that time “flows” from top to bottom in Humdrum, e.g. the token h_{11} is followed by the token h_{21}. A $\mathcal{H}$ pattern is interpreted as follows with respect to the temporal relations it enforces:

Lines

$\mathcal{H}$	$\mathcal{R}$
$\begin{bmatrix} a & b \end{bmatrix}$	$\rightsquigarrow \texttt{st}(a,b)$
$\begin{bmatrix} (a) & b \end{bmatrix}$	$\rightsquigarrow \texttt{sw}(b,a)$
$\begin{bmatrix} a & (b) \end{bmatrix}$	$\rightsquigarrow \texttt{sw}(a,b)$
$\begin{bmatrix} (a) & (b) \end{bmatrix}$	$\rightsquigarrow \texttt{ov}(a,b)$
$\begin{bmatrix} a & \star \end{bmatrix}, \begin{bmatrix} \star & a \end{bmatrix}, \begin{bmatrix} \star & (a) \end{bmatrix}, \ldots, \begin{bmatrix} \star & \star \end{bmatrix}$	$\rightsquigarrow \emptyset$

Columns

$\mathcal{H}$	$\mathcal{R}$
$\begin{bmatrix} a \\ b \end{bmatrix}$	$\rightsquigarrow \texttt{m}(a,b)$
$\begin{bmatrix} (a) \\ b \end{bmatrix}$	$\rightsquigarrow \texttt{m}(a,b)$
$\begin{bmatrix} a \\ (a) \end{bmatrix}, \begin{bmatrix} \star \\ a \end{bmatrix}, \ldots, \begin{bmatrix} \star \\ \star \end{bmatrix}$	$\rightsquigarrow \emptyset$

The Humdrum pattern of Figure 3ii can be simplified to the following (variable names correspond to those of Figure 2ii):

Example 1.
$$\begin{bmatrix} e & (a) \\ (e) & b \end{bmatrix} \quad \rightsquigarrow \quad \texttt{sw}(e,a), \texttt{sw}(b,e), \texttt{m}(a,b)$$

See Figure 6ii for the corresponding temporal network.

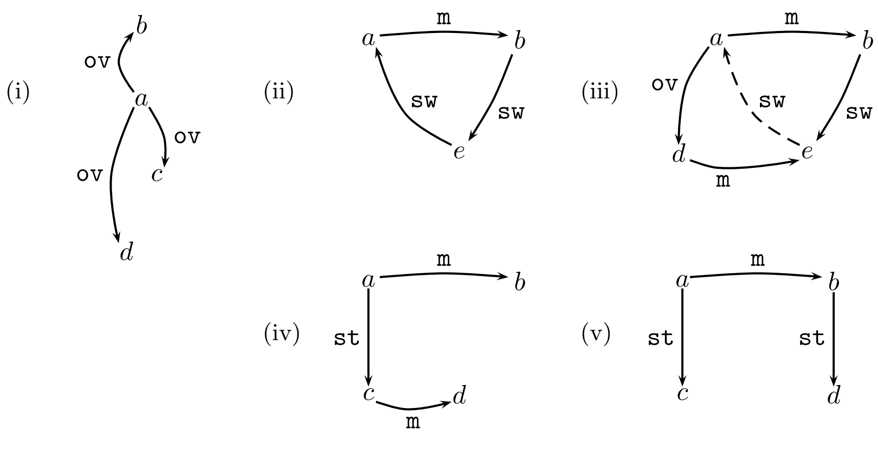


Fig. 6. Temporal networks enforced by (i) *Pattern 1*, (ii) *Example 1*, (iii) *Example 2* (the dashed edge is implied by the network), (iv) *Pattern 2* and (v) *Pattern 3*. The network (iv) can be represented in $\mathcal{SPP}$ but not in Humdrum. The network (v) can be represented in Humdrum but not in $\mathcal{SPP}$.

4 Structured Polyphonic Patterns

Patterns in $\mathcal{SPP}$ are defined according to the syntax below, where ε stands for an event and $-\varepsilon$ for a modified event (when layered, modified events start earlier than other events in the layer); the ";" operator joins two patterns in sequence (such that one finishes as the other starts) and the "==" operator layers two patterns (such that both start at the same time):

Definition 3.

$$\phi \in \mathcal{SPP} ::= \varepsilon \mid -\varepsilon \mid \phi \, ; \phi \mid \frac{\phi}{\phi}$$

A $\mathcal{SPP}$ pattern is interpreted as follows with respect to the temporal relations it enforces:

Layers

$\mathcal{SPP}$	$\mathcal{R}$
$\frac{a}{c}$	$\rightsquigarrow \mathtt{st}(a,c)$
$\frac{-a}{c}$	$\rightsquigarrow \mathtt{sw}(c,a)$
$\frac{a}{-c}$	$\rightsquigarrow \mathtt{sw}(a,c)$
$\frac{-a}{-c}$	$\rightsquigarrow \mathtt{ov}(a,c)$

Layers of sequences

$\mathcal{SPP}$	$\mathcal{R}$
$\frac{a\,;\,\dots}{c\,;\,\dots}$	$\rightsquigarrow \mathtt{st}(a,c)$
$\frac{-a\,;\,\dots}{c\,;\,\dots}$	$\rightsquigarrow \mathtt{sw}(c,a)$
$\frac{a\,;\,\dots}{-c\,;\,\dots}$	$\rightsquigarrow \mathtt{sw}(a,c)$
$\frac{-a\,;\,\dots}{-c\,;\,\dots}$	$\rightsquigarrow \mathtt{ov}(a,c)$

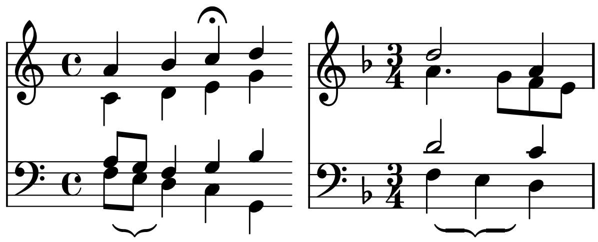
Sequences of layers

$\mathcal{SPP}$	$\mathcal{R}$
$\frac{a}{c}\,;\frac{b}{d}$	$\rightsquigarrow \mathtt{m}(a,b),\, \mathtt{m}(c,d)$
$\frac{-a}{c}\,;\frac{b}{d}$	$\rightsquigarrow \mathtt{m}(a,b),\, \mathtt{m}(c,d)$
$\frac{a}{-c}\,;\frac{b}{d}$	$\rightsquigarrow \mathtt{m}(a,b),\, \mathtt{m}(c,d)$
$\frac{a}{c}\,;\frac{-b}{d}$	$\rightsquigarrow \mathtt{m}(a,b),\, \mathtt{m}(c,d)$
$\vdots$	$\vdots$
$\frac{-a}{-c}\,;\frac{-b}{-d}$	$\rightsquigarrow \mathtt{m}(a,b),\, \mathtt{m}(c,d)$

Sequences

$\mathcal{SPP}$	$\mathcal{R}$
$a\,;\,b$	$\rightsquigarrow \mathtt{m}(a,b)$
$-a\,;\,b$	$\rightsquigarrow \mathtt{m}(a,b)$
$a\,;\,-b$	$\rightsquigarrow \mathtt{m}(a,b)$
$-a\,;\,-b$	$\rightsquigarrow \mathtt{m}(a,b)$

When ignoring pitch relations, the suspension example of Figure 3iii is simplified to the following pattern (also Figure 6iii):



(i) (ii)

Fig. 7. Layered passing tones captured by *Pattern* 2: BWV 255 bar 2 (i) and BWV 320 bar 19 (b)

Example 2.

$$\frac{-a}{-d}\ ;\ \frac{b}{-e} \quad \rightsquigarrow \quad \texttt{ov}(a,d), \texttt{sw}(b,e), \texttt{m}(a,b),\ \ \texttt{m}(d,e)$$

One can verify that the temporal relations enforced by example *Example 2* are indeed consistent with those of a suspension.

5 $\mathcal{H}$ and $\mathcal{SPP}$ are Distinct

Claim 1. $\mathcal{SPP} \not\subseteq \mathcal{H}$: there exists at least one pattern in $\mathcal{SPP}$ that has no equivalent in $\mathcal{H}$.

Consider the following $\mathcal{SPP}$ pattern (also Figure 6iv):

Pattern 2.

$$\frac{a\ ;\ b}{c\ ;\ d} \quad \rightsquigarrow \quad \texttt{st}(a,c), \texttt{m}(a,b), \texttt{m}(c,d)$$

With appropriate pitch relations, the pattern can capture layered passing tones (Figure 7), including pairs of passing tones that do not share the same rhythm: cases when b and d start together (Figure 7i) and cases when they are not synchronized (Figure 7ii).

Pattern 2 is not representable in Humdrum, due to the absence of a temporal relation between b and d. To capture the $\texttt{st}(a,c)$, $\texttt{m}(a,b)$ and $\texttt{st}(c,d)$ temporal relations enforced by the $\mathcal{SPP}$ pattern, the following three Humdrum patterns are possible:

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} \qquad \begin{bmatrix} a & c \\ (a) & d \\ b & \star \end{bmatrix} \qquad \begin{bmatrix} a & c \\ b & (c) \\ \star & d \end{bmatrix}$$

All of the above patterns enforce an additional temporal relation that is not enforced by the $\mathcal{SPP}$ pattern, respectively $\texttt{st}(b,d)$, $\texttt{sw}(d,a)$ and $\texttt{sw}(b,c)$. □

(i) (ii)

Fig. 8. Tritone resolutions captured by *Pattern* 3: BWV 257 bar 2 (i) and BWV 315 bar 13 (ii)

Claim 2. $\mathcal{H} \nsubseteq \mathcal{SPP}$: there exists at least one pattern in $\mathcal{H}$ that has no equivalent in $\mathcal{SPP}$.

Consider the following Humdrum pattern (also Figure 6v):

Pattern 3.
$$\begin{bmatrix} a & c \\ (a) & \star \\ b & d \end{bmatrix} \quad \rightsquigarrow \quad \texttt{st}(a,c), \texttt{st}(b,d), \texttt{m}(a,b)$$

With appropriate pitch relations, this captures both embellished and unembellished tritone resolutions (Figure 8). Figure 8ii, for example, is matched by *Pattern* 3 even if the A forming the tritone in the tenor voice does not meet with the G of the final chord. Rather, there is an embellishment in the form of an anticipation.

Clearly, the Humdrum pattern enforces $\texttt{st}(a,c)$ and $\texttt{st}(b,d)$. The only way to do that in $\mathcal{SPP}$ is to join a,c and b,d with the "$=$"operator. As the Humdrum pattern also enforces $\texttt{m}(a,b)$, these two must be joined by the ";"operator:

$$\frac{a}{c} ; \frac{b}{d}$$

But then, the $\mathcal{SPP}$ pattern will also enforce the temporal relation $\texttt{m}(c,d)$ which is clearly not enforced by *Pattern* 3. □

By similar arguments, one can prove that the dislocated chord pattern (*Pattern 1* and Figure 5) cannot be represented in either Humdrum or $\mathcal{SPP}$. This explains its place in the language hierarchy of Figure 1. This is also why the figure shows that $\mathcal{R}$ properly subsumes $\mathcal{H}$ and $\mathcal{SPP}$.

6 The Common Denominator $\mathcal{SPP}_{\texttt{seq}}$

Characterizing the intersection between Humdrum and $\mathcal{SPP}$, the pattern language $\mathcal{SPP}_{\texttt{seq}}$ restricts $\mathcal{SPP}$ to sequences of layers:

Definition 4.

$$\varphi \in \mathcal{SPP}_{\texttt{seq}} ::= \psi \mid \varphi \,;\, \psi \qquad \text{with} \qquad \psi ::= \varepsilon \mid -\varepsilon \mid \frac{\psi}{\psi}$$

With the additional restriction that there can by only one "−" operator per layer, except for the first layer, in which any number of "−" may appear. One can easily check, for example, that the suspension pattern (e.g. *Example 2*) is in $\mathcal{SPP}_{\texttt{seq}}$. Also, as $\mathcal{SPP}_{\texttt{seq}}$ can be interpreted the same way as the unrestricted $\mathcal{SPP}$ (Section 4), it follows that $\mathcal{SPP}$ subsumes $\mathcal{SPP}_{\texttt{seq}}$.

Claim 3. $\mathcal{SPP}_{\texttt{seq}} \subseteq \mathcal{H}$: for every $\varphi \in \mathcal{SPP}_{\texttt{seq}}$ there exists a pattern $h \in \mathcal{H}$ enforcing the same temporal network.

The proof proceeds by structural induction over the ";" operator (i.e. the claim holds as the sequence grows). The base cases are:

$$\frac{a}{c} \qquad \frac{-a}{c} \qquad \frac{a}{-c} \qquad \frac{-a}{-c}$$

Those are clearly covered by the following Humdrum patterns:

$$\begin{bmatrix} a\ c \end{bmatrix} \qquad \begin{bmatrix} (a)\ c \end{bmatrix} \qquad \begin{bmatrix} a\ (c) \end{bmatrix} \qquad \begin{bmatrix} (a)\ (c) \end{bmatrix}$$

Now, suppose there exists a pattern $h \in \mathcal{H}$ that covers the $\mathcal{SPP}_{\texttt{seq}}$ pattern φ. The induction cases are as follows (the case with two modified events $-\varepsilon$ does not appear; by definition of $\mathcal{SPP}_{\texttt{seq}}$, this is only allowed in the first layer):

$$\varphi \,;\, \frac{a}{c} \qquad \varphi \,;\, \frac{-a}{c} \qquad \varphi \,;\, \frac{a}{-c}$$

Suppose h has n lines. The induction cases are covered by:

$$\begin{matrix} h \\ \cdot \\ \begin{bmatrix} a\ c \end{bmatrix} \end{matrix} \qquad \begin{matrix} \begin{bmatrix} & \vdots & \\ h_{n1} & & h_{n2} \end{bmatrix} \\ \cdot \\ \begin{bmatrix} a & (h_{n2}) \\ (a) & c \end{bmatrix} \end{matrix} \qquad \begin{matrix} \begin{bmatrix} & \vdots & \\ h_{n1} & & h_{n2} \end{bmatrix} \\ \cdot \\ \begin{bmatrix} (h_{n1}) & c \\ a & (c) \end{bmatrix} \end{matrix}$$

The last two cases enforce an extra temporal relation (respectively $\texttt{sw}(a, h_{n2})$ and $\texttt{sw}(c, h_{n1})$) that the $\mathcal{SPP}_{\texttt{seq}}$ pattern does not enforce. However, that relation can be inferred by the temporal relations that the $\mathcal{SPP}_{\texttt{seq}}$ pattern do enforce. That is, referring back to Figure 2ii, whenever the relations $\texttt{m}(a, b)$, $\texttt{m}(d, e)$, $\texttt{sw}(b, e)$ and $\texttt{ov}(a, d)$ are present, then $\texttt{sw}(e, a)$ can be inferred. This inference is also indicated in Figure 6iii by a dashed edge. □

7 Discussion

This paper has presented three approaches that can accurately represent networks of temporal relations. Alternative approaches to polyphonic patterns often lack that accuracy. For example, vertical patterns [3] can only match polyphonic sources that have been expanded and sliced to yield a homophonic texture, hence not supporting the `sw` relation. A point set pattern representation [6] can only encode temporal relations with fixed duration ratios (capturing every instance of a `sw` relation would require a set of patterns, the size of which can grow quickly as many different ratios are likely to be found in the source). Techniques that rely on approximate matching to a source fragment [7] can confuse simultaneous notes with notes that overlap without being simultaneous, hence lacking precision with respect to the `st` relation.

With a little practice the musicologist should find it easy to write $\mathcal{SPP}$ patterns, in contrast to Humdrum, which requires extensive knowledge of Unix command line and regular expression tools. Relational patterns tend to be verbose and one quickly loses sight of the overall temporal structure of the pattern, where as the structure is syntactically expressed in $\mathcal{SPP}$. In Humdrum, this is readable when using the matrix form which this paper has developed. However, negations and disjunctions that can in principle appear in the regular expressions of a Humdrum pattern are not supported.

Finally, notice that $\mathcal{R}$ can express a great many temporal networks with unclear musical relevance (e.g. $\texttt{sw}(a, b)$, $\texttt{sw}(b, c)$) and even networks that are unsatisfiable (e.g. $\texttt{m}(a, b)$, $\texttt{st}(a, b)$). Perhaps there exists a restriction of $\mathcal{R}$ to "common sense" musical patterns. Ideally, such a restriction would preserve most of $\mathcal{R}$'s expressiveness, while being conducive to efficient pattern matching algorithms. Both Humdrum and $\mathcal{SPP}$ are candidate restrictions, yielding relational graphs that are always satisfiable. The graphs are also always connected and perhaps this connectedness an interesting avenue to explore for future research. In parallel, a website with tools and tutorials is being developed in an effort to make the languages presented in this paper more easily applicable to musicological tasks.

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), 832–843 (1983)
2. Bergeron, M., Conklin, D.: Structured polyphonic patterns. In: Ninth International Conference on Music Information Retrieval, Philadelphia, USA, pp. 69–74 (2008)
3. Conklin, D.: Representation and discovery of vertical patterns in music. In: Anagnostopoulou, C., Ferrand, M., Smaill, A. (eds.) ICMAI 2002. LNCS (LNAI), vol. 2445, pp. 32–42. Springer, Heidelberg (2002)
4. Fitsioris, G., Conklin, D.: Parallel successions of perfect fifths in the Bach chorales. In: Fourth Conference on Interdisciplinary Musicology, Thessaloniki, Greece (2008)

5. Jan, S.: Meme hunting with the Humdrum toolkit: Principles, problems, and prospects. Computer Music Journal 28(4), 68–84 (2004)
6. Meredith, D., Lemström, K., Wiggins, G.A.: Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. Journal of New Music Research 31(4), 321–345 (2002)
7. Typke, R., Veltkamp, R.C., Wiering, F.: Searching notated polyphonic music using transportation distances. In: ACM Multimedia Conference, New York, USA, October 2004, pp. 128–135 (2004)