Towards an Equational Theory of Rhythm Notation

Pierre Donat-Bouillud, Florent Jacquemard, Masahiko Sakai

Towards an Equational Theory of Rhythm Notation

Pierre Donat-Bouillud[1], Florent Jacquemard[2], and Masahiko Sakai[3]

[1] ENS Rennes, Ker Lann Campus, France.
pierre.donat-bouillud@ens-rennes.fr
[2] INRIA & Ircam, 1 place Igor Stravinsky, 75004 Paris, France.
florent.jacquemard@inria.fr
[3] Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan. sakai@is.nagoya-u.ac.jp

Trees are classical representations of hierarchical structures in symbolic music, in particular for rhythm notations, where the durations are defined by a hierarchy of subdivisions; see *e.g.* [6] chapter 3 for a survey. Structures called *rhythm trees* have been integrated since a long time into CAC environments such as Patchwork and OpenMusic [1,3], for programming rhythmic objects.

Term rewriting [2] and tree automata and transducers [4] are well established formalisms for transforming and reasoning on trees. With solid theoretical foundations, they are used in a wide range of applications including automatic reasoning, natural language processing, and foundations of web data processing. In this work, we consider a tree structured representation of rhythm suitable for defining a set of rewrite rules (*i.e.* oriented equations) preserving rhythms, while enabling the simplification of notations. This set can be seen as an axiomatization of rhythm notation which can be applied to reasoning on equivalent notations in assisted composition.

Ranked Tree Representation of Rhythm. We propose a tree-structured representation of rhythms based on a finite signature made of ranked symbols. Let us assume given a countable set of variables $\mathcal{X}$, and a finite set containing the first prime integers $\mathbb{P} = \{2, 3, 5, 7, \ldots\}$. A *tree* t is either a single node labeled with a variable $x \in \mathcal{X}$ or one of the following constant symbols n, r, s, 1, or it is made of one *root* node labeled with $p \in \mathbb{P}$ and p *subtrees* $t_1, \ldots, t_p$. In the latter case, t is denoted $p(t_1, \ldots, t_p)$. For $1 \leq i \leq p$, the *previous cousin* of t_i is either t_{i-1} if $i > 1$, or the last children of the previous cousin of t if $i = 1$ and if this tree exists. In other terms, it is the subtree whose root is the node immediately at left of the root of t_i, at the same level.

Intuitively, n represents a note, r a rest, $p(t_1, \ldots, t_p)$ a p-tuplet and the symbols s (as slur) and 1 are used to sum up durations. Formally, we associate a *duration value* to every tree t as follows. If t is not a subtree then we assume that is has been associated a duration which can be *e.g.* one beat (Figure 1), several beats or a whole bar (the actual value of this duration is arbitrary and not relevant for our purpose). If t is the subtree t_i in $t_0 = p(t_1, \ldots, t_p)$ (with $1 \leq i \leq p$) and t_0 has been associated the duration d_0, then the duration d associated to t is d_0/p plus

- the duration associated to its previous cousin if it exists and is labelled by 1, and
- the duration associated to the previous leaf in depth-first traversal (*dft*) if $t_i = \mathsf{s}$.

A tree represents a sequence of durations associated to some of its leafs. Formally, let us call a leaf *principal* if it is not labeled with 1 and if the next leaf wrt *dft* is not labeled with s. The *rhythmic value* of a tree is the sequence of the durations associated to its principal leaves, enumerated in *dft*.

This tree format has similarities with the rhythm trees (RT) of [1] but there are however important differences. In particular, although RT represent durations with integers labeling nodes, we only use the tree structure (the labels in $\mathbb{P}$ are not needed) and labels in a finite (and small) set for leaves. This make this representation more amenable to purely syntactic processing when RT needs arithmetic. This permits also to enforce certain subdivisions in rhythms, following the analogous of a *schema* for XML data defined *e.g.* according to the metre or user's constraints.

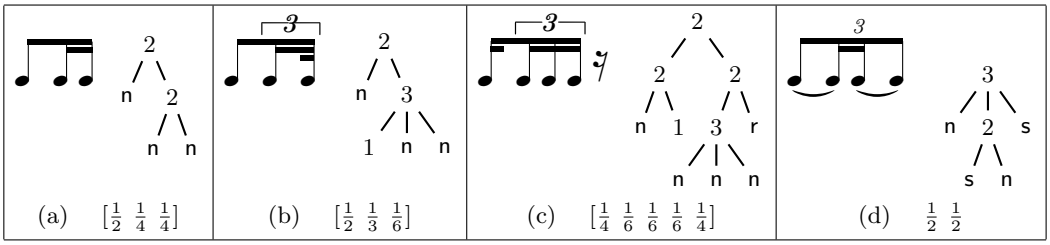

Fig. 1. Some trees with their corresponding rhythm notation and rhythmic values.

Rewrite Rules. We define rewrite rules which, applied to trees, do not change the rhythmic values. They can therefore be used to characterize equivalent notations of the same rhythm. The application of a rule of the form $\ell \to r$ to a tree t works by replacing by r subtrees of t matching ℓ. Some rules apply to successive cousins (the cousin relation begin denoted by ;) like: $1;\mathsf{n} \to \mathsf{n};\mathsf{s}$, $1;\mathsf{s} \to \mathsf{s};\mathsf{s}$, $1;\mathsf{r} \to \mathsf{r};\mathsf{r}$, $\mathsf{r};\mathsf{s} \to \mathsf{r};\mathsf{r}$, and the normalization rules, with variables: $1;2(x_1,x_2) \to x_1;x_2$, and $1;1;3(x_1,x_2,x_3) \to x_1;x_2;x_3$ *etc.* Other more standard rewrite rules [2] are applied to subtrees, such as $2(\mathsf{r},\mathsf{r}) \to \mathsf{r}$, $3(\mathsf{r},\mathsf{r},\mathsf{r}) \to \mathsf{r}$, *etc*, $2(\mathsf{s},\mathsf{s}) \to \mathsf{s}$, $3(\mathsf{s},\mathsf{s},\mathsf{s}) \to \mathsf{s}$, *etc* and $2(\mathsf{n},\mathsf{s}) \to \mathsf{n}$, $3(\mathsf{n},\mathsf{s},\mathsf{s}) \to \mathsf{n}$, *etc.* We also consider rules for redefining subdivisions: $2(x_1,x_2) \to 3(2(1,1),2(x_1,1),2(1,x_2))$, $2(x_1,x_2) \to 5(2(1,1),2(1,1),2(x_1,1),2(1,1),2(1,x_2))$, $\ldots$, $3(x_1,x_2,x_3) \to 2(3(1,x_1,1),3(x_2,1,x_3))$,...

Applying the above rules to the tree of Figure 1(d), we obtain the following rewrite sequence (the positions of application of rewrite rules are marked by circles).

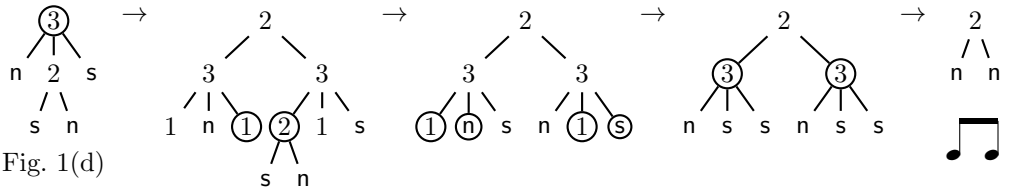

Fig. 1(d)

Discussion. Some additional symbols may be used for dealing with notation details regarding *e.g.* dots, tuplet beaming, ratio notations for tuplets ($n:m$ meaning n in the time of m) [5]. Our tree representation has been integrated in OpenMusic [3] with conversion functions to and from RT of [1]. We are also planning conversion to GUIDO and MusicXML.

Duration values are well defined only for trees satisfying constraints like: every 1 must be the previous cousin of some subtree, successive 1-labeled cousins must be associated the same duration. The subset of such trees is a *context-free tree language* [4]. An interesting research problem is whether this language is invariant under application of the rewrite rules.

The above rewrite sequence can be seen as a notation *simplification* for a given rhythm. In a context of assisted composition (CAC), this approach can permit suggesting a user various notations of the same rhythmic value, with different complexities. Using some standard measures for tree structures (involving depth, number of symbols...) enables to quantify this process.

Some other research questions are concerned with divergence in the application of the rewrite rules, the definition of appropriate rewrite strategies, canonical forms for rhythm notations and the *decision of equivalence* (*i.e.* equality of rhythmic values) of rhythm notations.

1. C. Agon, K. Haddad, and G. Assayag. Representation and rendering of rhythm structures. In *Proc. 2d Int. Conf. on Web Delivering of Music*, pages 109–113, 2002. IEEE Computer Society.
2. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

3. J. Bresson, C. Agon, and G. Assayag. OpenMusic: visual programming environment for music composition, analysis and research. In *Proc. of the 19th ACM int. conf. on Multimedia*. ACM, 2011.
4. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, C. Löding, D. Lugiez, S. Tison, and M. Tommasi. *Tree Automata Techniques and Applications*. http://tata.gforge.inria.fr, 2007.
5. E. Gould. *Behind Bars: The Definitive Guide to Music Notation*. Faber Music, 2011.
6. D. Rizo. *Symbolic music comparison with tree data structures*. PhD thesis, Uni.de Alicante, 2010.