HARMONIC SYNTAX IN TIME
RHYTHM IMPROVES GRAMMATICAL MODELS OF HARMONY

Daniel Harasim[1] **Timothy J. O'Donnell**[2] **Martin Rohrmeier**[1]
[1] Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne, Switzerland
[2] Department of Linguistics, McGill University, Canada
daniel.harasim@epfl.ch

ABSTRACT

Music is hierarchically structured, both in how it is perceived by listeners and how it is composed. Such structure can be elegantly captured using probabilistic grammatical models similar to those used to study natural language. They address the complexity of the structure using abstract categories in a recursive formalism. Most existing grammatical models of musical structure focus on one single dimension of music–such as melody, harmony, or rhythm. While these grammar models often work well on short musical excerpts, accurate analysis of longer pieces requires taking into account the constraints from multiple domains of structure. The present paper proposes abstract product grammars–a formalism which integrates multiple dimensions of musical structure into a single grammatical model–along with efficient parsing and inference algorithms for this formalism. We use this model to study the combination of hierarchically-structured harmonic syntax and hierarchically-structured rhythmic information. The latter is modeled by a novel grammar of rhythm that is capable of expressing temporal regularities in musical phrases. It integrates grouping structure and meter. The combined model of harmony and rhythm outperforms both single-dimension models in computational experiments. All models are trained and evaluated on a treebank of hand-annotated Jazz standards.

1. INTRODUCTION

Music is hierarchically organized, which is probably most evident in the structure of harmonic sequences. Grammatical models of music describe both local and non-local relations between musical objects such as notes or chords by assuming a latent hierarchical structure. Originally inspired by Schenkerian analysis and generative linguistics [9], grammatical models have been used in music theory [14, 19, 24, 25], computational musicology [1, 5, 6, 13, 16, 27], music information retrieval [3, 4, 12, 18, 26], and increasingly also music psychology [7, 20]. Consider for example the Jazz chord sequence C^6 D^7 Dm^7 G^7 C^6 of the A-part of the Jazz standard *Take the A-Train*. A hierarchical analysis of this sequence is shown in Figure 1a. The progression D^7 Dm^7 G^7 forms a *dominant phrase* inside the *tonic phrase* C^6 D^7 Dm^7 G^7 C^6, exhibiting a non-local harmonic relationship between the chords D^7 and G^7. The nesting of the phrases moreover illustrates the idea of how pieces can be decomposed into hierarchically-structured constituents (subtrees) which stand in part-whole relationship with one another [6]. Figure 2 displays a typical case of a non-local harmonic relation in Jazz harmony.

To analyze hierarchical harmonic structures, music theorists make use of many additional structural features such as melody, rhythm, voice-leading, and form, for disambiguation. From this perspective, the latent harmonic structure of a piece cannot be fully inferred from sequences of chord symbols alone. Most existing grammatical models of harmony, however, do not take these other domains of musical structure in account. In this paper, we propose a novel formalism that combines models of different musical features. The mathematical idea is similar to Coupled-context-free Grammars [17]. We extend that approach by a probabilistic model and apply the general construction to improve models of harmonic syntax by incorporating harmonic rhythm.

1.1 Problem Setting

Existing grammatical models of harmony typically do not capture how harmonic structure is laid out in time [21], as shown in Figure 1a. This analysis captures information such as the dependencies between different kinds of musical phrase (tonic, dominant, subdominant), ordering, and hierarchical constituency, but contains no information on the duration of chords. This paper extends models of harmonic syntax to include rhythmic structure illustrated in Figure 1b. This figure shows how the musical phrases in Figure 1a are laid out in time by progressively assigning constituents to a metrical grid consisting of eight measures. The inclusion of the metrical domain reveals previously hidden structure. In the first step, the root of the harmonic tree is assigned to the entire eight bars. In the second step, the tonic phrase is split into equal halves which are assigned to bars 1-4 and bars 5-8 of the metrical grid. In the third step, the second half of the piece is split into equal halves, introducing a V in the first part of the split

Attribution: Daniel Harasim, Timothy J. O'Donnell, Martin Rohrmeier . "Harmonic Syntax in Time Rhythm Improves Grammatical Models of Harmony", 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019.

and limiting the tonic scale degree to the second part. The fourth step, in contrast, splits the first half (measures 1–4) into two and assigns the second half of this split to the second half of the progression (measures 5–8). Measures 3 and 4 are said to be a *harmonic upbeat* to measures 5 and 6. In the following, we present an integrated model of harmony and phrase rhythm [22] that accounts for the structural differences of the steps three and four. Note that we therefore assume the existence of hypermeter, the extension of metrical structures within a single measure to relations between measures [11].

We propose an approach that models the upbeat and the downbeat of harmonic constituents separately. Figure 1c shows a hierarchical analysis integrating harmonic syntax and harmonic rhythm. In this notation, the durations of upbeats are separated from the durations of downbeats by the symbol $\oplus$. The symbol $\ominus$ is used to indicate the "time stealing" from generation step 3 in Figure 1b.

2. GRAMMATICAL MODELS

2.1 Abstract Context-Free Grammars

The following two definitions are adopted from [6], where further explanation and examples can be found.

A *(non-probabilistic) Abstract Context-free Grammar* $G = (T, C, C_0, \Gamma)$ consists of a set T of *terminal symbols*, a set C of *constituent categories*, a set of *start categories* $C_0 \subseteq C$, and a set of partial functions

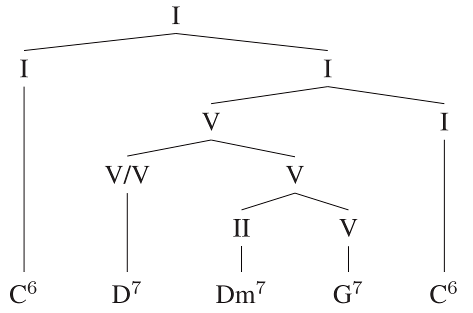
$$\Gamma := \{\, r \mid r : C \nrightarrow (T \cup C)^* \,\}, \tag{1}$$

called *rewrite rules* or *rewrite functions*. The arrow $\nrightarrow$ is used throughout the paper to denote partial functions and $\mathrm{dom}(r)$ denotes the set of arguments for which a partial function r is defined. A sequence $\beta \in (T \cup C)^*$ can be generated from a sequence $\alpha \in (T \cup C)^*$ by one *rule application* of a rewrite function $r \in \Gamma$, denoted by $\alpha \longrightarrow_r \beta$, if there exist $\alpha_1, \alpha_2 \in (T \cup C)^*$ and $A \in C$ such that $\alpha = \alpha_1 A \alpha_2$ and $\beta = \alpha_1 r(A) \alpha_2$. A sequence of rewrite rules $r_1 \dots r_n$ is called a *derivation* of a sequence of terminals $\alpha \in T^*$ if there exists a start category $\alpha_1 \in C_0$, and $\alpha_2, \dots, \alpha_n \in (C \cup T)^*$ such that

$$\alpha_1 \longrightarrow_{r_1} \alpha_2 \longrightarrow_{r_2} \cdots \longrightarrow_{r_n} \alpha, \tag{2}$$

where r_i is always applied to the leftmost category of α_i for $i \in \{1, \dots, n-1\}$. The set of derivations of α is denoted by $D(\alpha)$. The language of the grammar G is the set of terminal sequences that have a derivation in G.

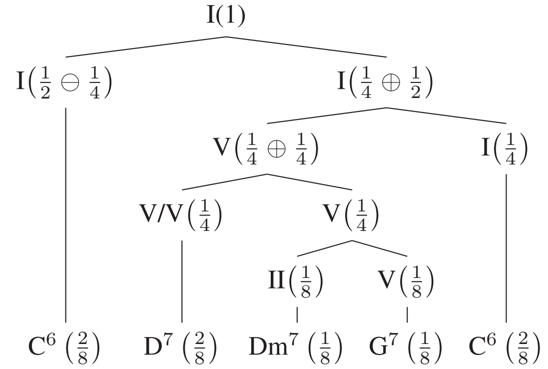
A *Probabilistic Abstract Context-free Grammar* is an Abstract Context-free Grammar where each category $A \in C$ is associated with a random variable X_A over rewrite functions r such that the probability $\mathbb{P}(X_A = r)$ is positive if and only if $r(A)$ is defined, that is $A \in \mathrm{dom}(r)$. In the following, we also use the notation $p(r \mid A) = \mathbb{P}(X_A = r)$ and $p(A \longrightarrow_r \alpha) = \mathbb{P}(X_A = r)\ \mathbb{1}(r(A) = \alpha)$. The probability $p(d)$ of a derivation $d = r_1 \dots r_n$ of a sequence of terminal symbols $\alpha \in T^*$ is defined as the product $\prod_{i=1}^{n} \mathbb{P}(r_i \mid A_i)$ where in each step r_i is applied to a category $A_i \in C$. The probability of α is then defined as $p(\alpha) = \sum_{d \in D(\alpha)} p(d)$.



(a) Generative syntax tree of the harmonic structure. The leafs of the tree are the chord symbols of the A-part. The internal nodes show scale degrees with respect to C major as latent categories. Subtrees form harmonic constituents. The nested structure of the subtrees shows how complex constituents are build from simpler constituents [6].

1:	I							
2:	I				I			
3:	I				V		I	
4:	I		V/V		V		I	
5:	I		V/V		II	V	I	
6:	C^6		D^7		Dm^7	G^7	C^6	

(b) Schematic generation of the chord sequence including their metrical positions. Each row consists of 8 measures and shows one step in the generation process. Chords are tied over following "empty" measures. The third and the fourth step show the two basic kinds of harmonic preparation with respect to their metrical placement. In step three, the preparation of the I by the V pushed the I back by two measures while in step four, the preparation of V by V/V protrudes into the time domain of the preceding I.



(c) Generative syntax tree of the harmonic structure with integrated rhythmic information. The numbers in parentheses denote the duration of the constituents relative to the whole progression. The branch $\mathrm{I}(1) \longrightarrow \mathrm{I}(\frac{1}{2} \ominus \frac{1}{4})\ \ \mathrm{I}(\frac{1}{4} \oplus \frac{1}{2})$ is an instance of a split that anticipates the upbeat preparation of G^7 by D^7. Because of a 2 measures long upbeat, the left child is 2 measures shorter and the right child is 2 measures longer than in a preparation without an upbeat.

Figure 1: Hierarchical analysis of the A-part of the Jazz standard *Take the A-Train* in C major, considering the structural domains of harmony and rhythm.

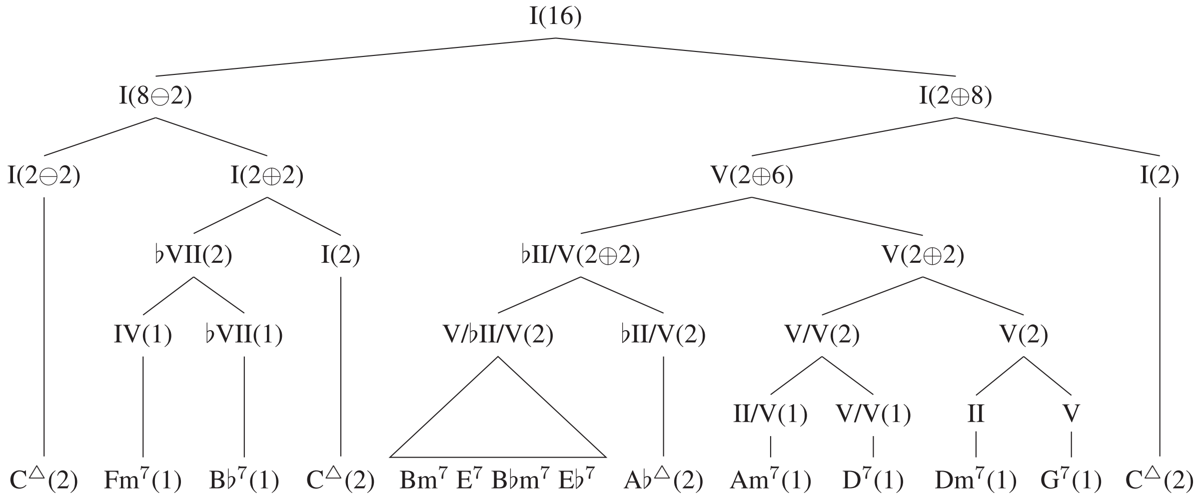


Figure 2: Hierarchical analysis of the Jazz standard *Half Nelson*, integrating harmonic and rhythmic structure. In this tree, a duration of 1 corresponds to one measure for the sake of readability (the whole tune spans 16 measures). The non-local dependency between the chords A♭$^{\triangle}$ and G^7 constitutes a characteristic harmonic relation of the tune.

2.2 Product Grammars

This paper proposes to improve generative grammar models of harmony by forming a product of a harmony grammar and a rhythm grammar.

Let $G = (T, C, C_0, \Gamma)$ and $G' = (T', C', C'_0, \Gamma')$ be two PACFGs and let $\mathrm{ar}(r)$ denote the *arity* of a rule r, which is defined as the length of its right-hand side. The *product grammar*

$$G \bowtie G' = (T \times T', C \times C', C_0 \times C'_0, \Gamma \bowtie \Gamma') \qquad (3)$$

is constructed from the Cartesian products of the sets of terminals, categories, and start categories. The rewrite functions of $G \bowtie G'$ are all pairs of functions of equal arity,

$$\Gamma \bowtie \Gamma' = \{\, (r, r') \in \Gamma \times \Gamma' \mid \mathrm{ar}(r) = \mathrm{ar}(r') \,\}. \qquad (4)$$

For a product category $(A, A') \in C \times C'$ and rewrite functions $r \in \Gamma$ and $r' \in \Gamma'$ of equal arity, the application of (r, r') to (A, A') is defined component-wise,

$$(r, r')(A, A') = (r(A), r'(A')). \qquad (5)$$

By abuse of notation, the right-hand side of this equation does not stand for a pair of sequences, but a sequence of pairs. The probability of a product rule application is defined as the product of the probabilities of the rule application components,

$$p((r, r') \mid (A, A')) = p(r \mid A)\, p(r' \mid A'). \qquad (6)$$

That is, the choice of rule r is set to be independent of A' and r', and the choice of r' is independent of A and r in the generative process.

A helpful intuition of product grammars is that they compute the intersection of two sets of derivation trees for a sequence. The derivation trees of the grammar $G \bowtie G'$ are exactly those which are derivations in both G and G' if the labels of the trees (terminals and categories) are ignored. The probability of a derivation in $G \bowtie G'$ is then also equal to the product of its corresponding derivations in G and G'.

2.3 Rhythm Grammar

2.3.1 Full Rhythm Grammar

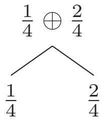
A *rhythmic category* $a \oplus b$ consists of two rational numbers $a \in \mathbb{Q}$ and $b \in \mathbb{Q}$ such that $0 \le a$, $0 < b$, and $a + b \le 1$. The first number a is called the *upbeat* and the second number b is called the *downbeat* of the category. The intuition behind the symbol $\oplus$ is that the total length of a rhythmic category equals the sum of its two components, $\lambda(a \oplus b) := a + b$, where λ is the function that denotes the length of the rhythmic constituent as a proportion of the overall piece, which is fixed to be the unit $1 \in \mathbb{Q}$. The condition $0 \le a$ forbids negative upbeat parts, $0 < b$ ensures positive category lengths, and $a + b \le 1$ ensures that no category is longer than the whole piece. For convenience, we use two additional short-hand notations: a category with no upbeat is denoted by the length of its downbeat, $b = 0 \oplus b$. The category of a rhythmic constituent that loses a portion c of its downbeat (formerly with length b) to the upbeat of the following rhythmic constituent is denoted by $b \ominus c := 0 \oplus (b - c)$. In this case $\lambda(b \ominus c) = b - c$, too.

The start category of the rhythmic grammar is 1, the length of the piece, and any category with zero upbeat is allowed to be a terminal (leaf node). The essential grammar rules are given by two families of rewrite functions, one family of partial functions for splitting the upbeat components of categories $\mathtt{usplit}_v : C \nrightarrow C^*$ and one family of total functions for splitting the downbeats $\mathtt{dsplit}^u_v : C \to C^*$,

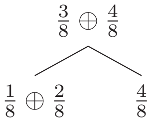
$$\begin{aligned} \mathtt{usplit}_u(a \oplus b) &:= ((1-u)a \oplus ua) \qquad (0 \oplus b) \\ & \qquad\qquad\qquad\qquad\qquad\qquad (7) \\ \mathtt{dsplit}^v_w(a \oplus b) &:= (a \oplus (1 - v - vw)b) \quad (vwb \oplus wb), \end{aligned}$$

where $u, v, w \in \mathbb{Q}$ such that $\frac{1}{2} < u \le 1$ and $a > 0$ in the first equation, and $0 \le v < 1$ and $0 < w < 1$ in the second equation. The parameter u represents the downbeat proportion of the upbeat, v is the upbeat proportion of the second category of a downbeat split, and w is the downbeat proportion of the second category of a downbeat split.

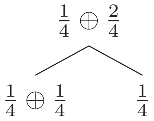
In other words: The upbeat split rule $\texttt{usplit}_u$ separates the upbeat from the downbeat and optionally splits the upbeat again into a new upbeat and downbeat. For example for $u = 1$ and $u = \frac{2}{3}$:



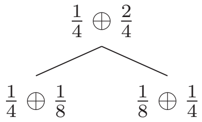
and



In contrast, the downbeat split $\texttt{dsplit}^v_w$ ignores the upbeat and splits the downbeat. It optionally introduces a new upbeat preparation. For example for $v, w = 0, \frac{1}{2}$ and $v, w = \frac{1}{2}, \frac{1}{2}$:



and



One rule $\texttt{unary}(a \oplus b) := a \oplus b$ is added to the grammar to ensure compatability with grammars that use rewrite rules of arity one.

The probability of a rhythmic rewrite functions does not depend on the particular rhythmic category that it rewrites, but only on whether or not the category has an upbeat of length zero. This enables a maximal sharing of probability mass by preserving consistency with the constraints of the rewrite rules. More precisely,

$$\begin{aligned} 1 = \; & p(\texttt{unary} \mid a \oplus b) \\ & + \sum_{\frac{1}{2} < u \leq 1} p(\texttt{usplit}_u \mid a \oplus b) \\ & + \sum_{0 \leq v < 1} \sum_{0 < w < 1} p(\texttt{dsplit}^v_w \mid a \oplus b) \end{aligned} \tag{8}$$

for $a > 0$ and

$$\begin{aligned} 1 = \; & p(\texttt{unary} \mid 0 \oplus b) \\ & + \sum_{0 \leq v < 1} \sum_{0 < w < 1} p(\texttt{dsplit}^v_w \mid 0 \oplus b). \end{aligned} \tag{9}$$

For practical applications, the parameters u, v, and w are limited to a finite set of rational numbers to put a proper normalized prior on the rule distributions.

2.3.2 *Simplified Rhythm Grammar*

For comparison, we additionally consider a simplified version of the rhythm grammar presented above which does not explicitly model upbeats. The rhythmic categories and the terminals of this grammar are rational numbers $0 < a \leq 1$ representing constituent durations relative to the full piece. Apart from the technical unary rule, the rules of the grammar form a family of total rewrite functions

$$\texttt{split}_s(a) := (sa) \qquad (a - sa). \tag{10}$$

The parameter $0 < s < 1$ is called the *temporal split ratio* of the rule. The probabilities of the rewrite rules are set to be independent from the category they rewrite. Therefore,

$$1 = p(\texttt{unary}) + \sum_{a \in \mathbb{Q}} p(\texttt{split}_a). \tag{11}$$

2.4 Harmony Grammar

The harmony grammar used in this paper is a standard probabilistic context-free grammar (Σ, N, S, R) in Chomsky normal form. It consists of a set Σ of chord symbols as terminal symbols, a set of copies of chord symbols N as non-terminal symbols, a distinguished start symbol $S \in N$, and a set of standard rewrite rules

$$R \subseteq \{\, A \longrightarrow B_1\ B_2 \mid B_k \in N, A = B_1 \text{ or } A = B_2 \,\}\,.$$

In particular, rules of the form $A \longrightarrow A\ A$ are included by this definition. Each non-terminal symbol A is also associated with a random variable X_A over rewrite rules that have A as their left-hand side. The symbols, rules, and parameters of the grammar are read from dataset of tree annotations described in the next section.

Note that since every rewrite rule of a standard context-free grammar can be interpreted as a partial function with a singleton domain,

$$\text{dom}(A \longrightarrow \alpha) = \{\, A \,\} \quad \text{for all } \alpha \in (\Sigma \cup N)^*, \tag{12}$$

every standard context-free grammar is also an Abstract Context-free Grammar and can be used in the product grammar construction.

3. DATASET

This study uses a dataset of 75 hand-annotated tree analyses of Jazz chord sequences from the iRealPro dataset [23]. The tree annotations were performed by the authors and a student assistant. Each chord sequence is annotated with a single binary tree that spans the whole piece. In contrast to the introductory examples of this paper, the internal nodes of each tree in the data are not labeled by scale degrees but chord symbols. depth one subtrees corresponds to a rule of the grammar described in the previous section. Figure 3 shows the absolute frequencies of the 20 most frequent harmonic rewrite rules from the dataset, after each sequence was transposed to the root of C. Rules of the form $A \longrightarrow A \quad A$, called *prolongation rules*, and rules of the form $A \longrightarrow B \quad A$ for $A \neq B$, called *preparation rules*, are the most used rule schemes.

The dataset additionally includes the length of each chord in quarter notes. The chord durations of each piece are divided by the total duration of the piece. From the chord durations and the harmonic tree annotations, the duration of each constituent (subtree) can be calculated automatically as shown in Figure 4. The temporal split ratios of the rule applications–as introduced in Equation 10–are then in turn calculated from the durations of the constituents. Consider for example the rule application $\text{G}^7(\frac{5}{32}) \longrightarrow \text{F}^{\triangle}(\frac{2}{32}) \ \ \text{G}^7(\frac{3}{32})$ from Figure 4. Its temporal split ratio is $\frac{2}{5}$.

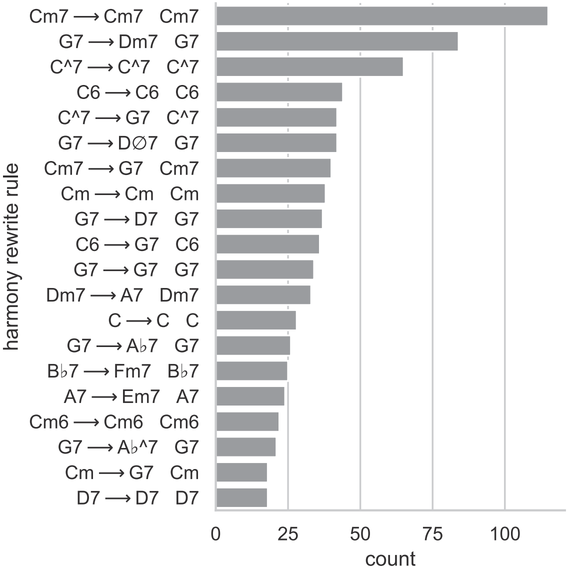


Figure 3: Absolute frequencies of the 20 most frequent harmonic rewrite rules of the tree annotations. All sequences are transposed to the common root C. Major-seventh chords are denotes as `C^7` and `A♭^7`.

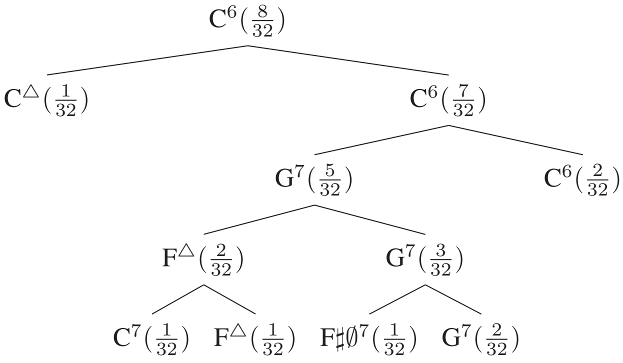


Figure 4: Tree annotation of the last chords of *St. Thomas*. Chord durations are shown relative to the total duration of the tune, $\frac{2}{32}$ corresponds to one measure. The durations of the inner nodes are calculated automatically.

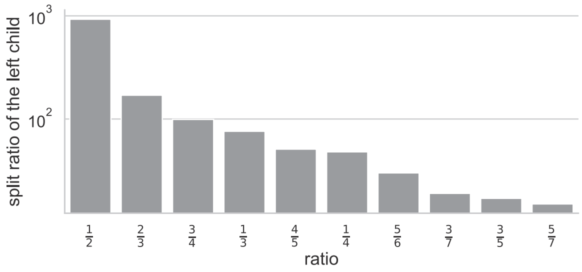


Figure 5: Absolute frequencies of the 10 most frequent split ratios of annotated tree constituents. The split ratio of a binary rewrite rule is defined as the time proportion of the left child. The y-axis is plotted using a logarithmic scale.

The 10 most frequent temporal split ratios are shown in Figure 5. The split ratio $\frac{1}{2}$ is by far the most frequent one. Most of the remaining ratios can be expressed either as $\frac{n-1}{n}$ or as $\frac{1}{n}$ for some $n \in \mathbb{N}$. The former arise for example from chains of descending fifths or applied dominants that accumulate time step by step in the temporal order of the piece. The latter arise from upbeat preparations that can be understood using the rhythmic categories described in Section 2.3.1. Two rhythmic rewrite rules that explain a split ratio of $\frac{1}{n}$ are $\left(\frac{n}{m}\right) \longrightarrow \left(\frac{\frac{n}{2}}{m} \ominus \frac{\frac{n}{2}-1}{m}\right) \quad \left(\frac{\frac{n}{2}-1}{m} \oplus \frac{\frac{n}{2}}{m}\right)$ and $\left(\frac{1}{m} \oplus \frac{n-1}{m}\right) \longrightarrow \left(\frac{1}{m}\right) \quad \left(\frac{n-1}{m}\right)$, where $m \in \mathbb{N}$. The former results from a downbeat split with $w = \frac{1}{2}$ and the latter results from an upbeat split with $u = 1$.

4. PARSING WITH PRODUCT GRAMMARS

A naive approach to parsing against a product grammar would enumerate all product categories and memoize the inverted rewrite rules on these categories. In this section, we show how the inefficient blow-up of the number of categories can be avoided using the independence assumption of Equation 6.

Consider an Abstract Context-Free Grammar in Chomsky normal form. The standard CYK algorithm–here used to calculate the probability of a sequence of terminals $w \in T^*$ of length n, indexed from 0 to $n-1$–can be formulated recursively by the equations

$$p(A, i, i) = \sum_{r \in \Gamma} p(A \longrightarrow_r w_i) \tag{13}$$

and

$$\begin{aligned} &p(A, i, j) \\ &= \sum_{k=i}^{j-1} \sum_{r \in \Gamma} \quad p(A \longrightarrow_r B_1\ B_2) p(B_1, i, k)\ p(B_2, k+1, j) \end{aligned} \tag{14}$$

where $A, B_1, B_2 \in C$ and $i, j \in \mathbb{N}$ such that $0 \leq i < j \leq n-1$. The probability of the sequence is then given by $p(w) = \sum_{A \in C_0} p(A, 0, n-1)$.

Given a product grammar $G \bowtie G'$, a sequence of product terminals can be parsed utilizing Equation 6,

$$p((A, A'), i, i) = \sum_{(r,r') \in \Gamma \bowtie \Gamma'} p(A \longrightarrow_r w_i)\ p(A' \longrightarrow_{r'} w'_i) \tag{15}$$

and

$$\begin{aligned} p((A, A'), i, j) = \sum_{k=i}^{j-1} \sum_{(r,r') \in \Gamma \bowtie \Gamma'} & p(A \longrightarrow_r B_1\ B_2) \\ & p(A' \longrightarrow_{r'} B'_1\ B'_2) p((B_1, B'_1), i, k)\ p((B_2, B'_2), k+1, j) \end{aligned} \tag{16}$$

It is therefore sufficient to parse the component grammars individually at each step. In other words, the combined grammar is computed on-the-fly to achieve efficiency.

5. EXPERIMENTS

We compare four product grammars that integrate harmonic and rhythmic structure. Additionally, we report the performances of their single-domain components and of a random baseline. As first component, we consider the harmony grammar presented in Section 2.4, trained either on the annotations in the original keys of the tunes or on the annotations after each tune was transposed to C major. As second component, we consider the full rhythm grammar presented in Section 2.3.1 that distinguishes upbeats and downbeats of constituents, and its simplification that uses the total length of the constituents, presented in Section 2.3.2. All models are trained and evaluated on the dataset described in Section 3. Apart from the full rhythm grammar, all models are trained by counting the harmonic rewrite rules or the temporal split ratios present in the dataset. The full rhythm grammar is trained using variational Bayesian inference [8]. Every model predicts the latent tree structure of a given sequence using the maximum a posteriori tree. One-fold cross validation was applied to avoid overfitting to the data: 75 times the model was trained on 74 sequences and evaluated on the remaining sequence.

5.1 Evaluation Metric and Baseline

The similarity of two trees is calculated as the unlabeled tree accuracy, defined as follows. Let α be a sequence of n terminals, left-to-right indexed from 0 to $n - 1$, let t be a tree with α as leafs, and let s be a subtree of t. The *span* of s is defined as the pair of the index of its left-most child and the index of its right-most child. The set of spans of t consists of the spans of all subtrees of t that are not leafs. The unlabeled tree accuracy of a tree prediction t to the respective Goldstandard tree t^* is then defined as the cardinality of the correctly predicted spans, divided by the total amount of spans of t^*.

Given a chord sequence of length n, the random baseline uniformly samples one tree from the set of all binary trees with n leafs.

5.2 Results and Discussion

The results of the computational experiments are shown in Figure 6. All combined models of harmony and rhythm perform significantly better than the single-domain harmony grammars and all models perform significantly better than the random baseline ($p < 0.01$ using 2-sample bootstrap tests). There is no statistical difference observable between the not transposed and the transposed harmony models. Surprisingly, the single-domain rhythm grammars perform much better than the single-domain harmony grammars. This is, however, only possible because we consider the unlabeled tree accuracy. Other measures such as perplexity would reveal the obvious incapability of the rhythm grammars to predict chord sequences.

Both rhythm grammars improve the harmony models similarly. As discussed in Section 3, the simplified version of the proposed rhythm grammar is also able to capture some complex rhythmical structures. The music-theoretically more sophisticated formalism, however, facilitates the interpretation and explanation of the observed split ratios.

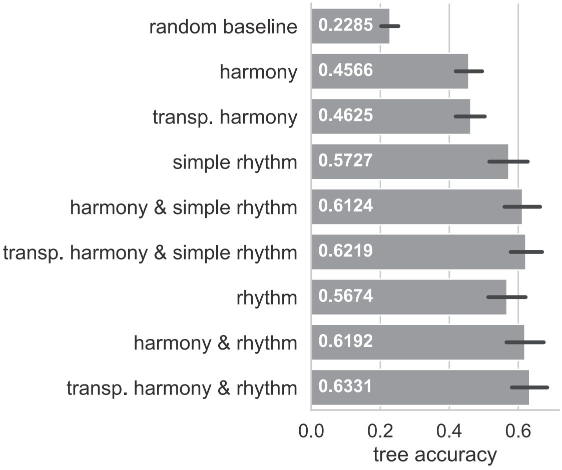


Figure 6: One-fold cross-validated tree accuracies of the tested models and the random baseline. The error bars show 95% bootstrap confidence intervals. The combined models of harmony and rhythm perform significantly better than the plain harmony grammars.

6. CONCLUSION

The usage of rhythmical information is shown to significantly improve the performance of harmonic syntax models. The empirical comparison between a music-theoretical motivated model and its simplified version shows that both models improve the harmony grammar equally well. The simplified model can therefore be used as an algorithmic proxy of the more expressive model. This might, however, only be true for rhythmically regular structures such as the harmonic rhythm of chord sequences from Jazz standards. It is, moreover, surprising how much information is already contained in the rhythm of the sequences, which underpins the importance of the rhythmic dimension of music [10]. In these sequences, both the harmonic syntax and the phrase rhythm work together to strengthen the intentionality of the music.

The here proposed model of interaction between harmony and rhythm is also capable to describe the interaction of pitch and rhythm in melodies. A rewrite function for syncopation could be added for future applications, since syncopation is an essential part of melodic rhythm.

The general product grammar construction presented in this paper integrates multiple domains of structure using strong independence assumptions. Future research can extent the formalism, explicitly modeling inter-domain dependencies. We hope that the presented approach will prove to be useful for applications such as rhythm quantization [2], the definition of similarity metrics [5], and computational composition assistance [15].

7. ACKNOWLEDGEMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 760081 – PMSB and from the Natural Sciences and Engineering Research Council of Canada (NSERC). We thank Claude Latour for supporting this research through the Latour Chair in Digital Musicology. The authors additionally thank the anonymous referees for their valuable comments.

8. REFERENCES

[1] Samer A. Abdallah and Nicolas E. Gold. Comparing models of symbolic music using probabilistic grammars and probabilistic programming. *Proceedings of ICMC/SMC*, pages 1524–1531, 2014.

[2] Francesco Foscarin, Florent Jacquemard, Philippe Rigaux, and Sakai Masahiko. A parse-based framework for coupled rhythm quantization and score structuring. In *Mathematics and Computation in Music*, pages 248–260, Cham, 2019. Springer International Publishing.

[3] Mark Granroth-Wilding and Mark Steedman. A Robust Parser-Interpreter for Jazz Chord Sequences. *Journal of New Music Research*, 43(4):355–374, October 2014.

[4] W Bas De Haas, José Pedro Magalhães, and Frans Wiering. Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 295–300, 2012.

[5] W Bas De Haas, Martin Rohrmeier, and Frans Wiering. Modeling Harmonic Similarity using a Generative Grammar of Tonal Harmony. *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*, 2009.

[6] Daniel Harasim, Martin Rohrmeier, and Timothy J O'Donnell. A Generalized Parsing Framework for Generative Models of Harmonic Syntax. *19th International Society for Music Information Retrieval Conference*, 2018.

[7] Stefan Koelsch, Martin Rohrmeier, Renzo Torrecuso, and Sebastian Jentschke. Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences*, 110(38):15443–15448, 2013.

[8] Kenichi Kurihara and Taisuke Sato. An Application of the Variational Bayesian Approach to Probabilistic Context-Free Grammars. In *International Joint Conference on Natural Language Processing {(IJCNLP-04)} Workshop Beyond Shallow Analyses*, 2004.

[9] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. Cambridge, MA, 1983.

[10] Florence Levé, Richard Groult, Guillaume Arnaud, Cyril Séguin, Rémi Gaymay, and Mathieu Giraud. Rhythm extraction from polyphonic symbolic music. In *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pages 375–380, 2011.

[11] Ryan McClelland. Extended upbeats in the classical minuet: Interactions with hypermeter and phrase structure. *Music Theory Spectrum*, 28(1):23–55, Spring 2006.

[12] Andrew McLeod and Mark Steedman. Meter Detection in Symbolic Music Using a Lexicalized PCFG. *Proceedings of the 14th Sound and Music Computing Conference*, 2017.

[13] Eita Nakamura, Masatoshi Hamanaka, Keiji Hirata, and Kazuyoshi Yoshii. Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 276–280, 2016.

[14] Markus Neuwirth and Martin Rohrmeier. Towards a syntax of the Classical cadence. In *What is a Cadence*, pages 287–338. Leuven University Press, 2015.

[15] Alexandre Papadopoulos, Pierre Roy, and François Pachet. Assisted lead sheet composition using flowcomposer. In Michel Rueher, editor, *Principles and Practice of Constraint Programming*, pages 769–785, Cham, 2016. Springer International Publishing.

[16] Marcus Pearce and Martin Rohrmeier. Musical syntax ii: empirical perspectives. In *Springer handbook of systematic musicology*, pages 487–505. Springer, 2018.

[17] Gisela Pitsch. LR(k)-Parsing of Coupled-Context-Free Grammars. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, 1994.

[18] Donya Quick. Learning production probabilities for musical grammars. *Journal of New Music Research*, 45(4):295–313, 2016.

[19] Martin Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 2011.

[20] Martin Rohrmeier and Ian Cross. Tacit tonality: Implicit learning of context-free harmonic structure. In *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009) Jyväskylä, Finland*, pages 443–452, 2009.

[21] Martin Rohrmeier and Marcus Pearce. Musical syntax i: Theoretical perspectives. In *Springer handbook of systematic musicology*, pages 473–486. Springer, 2018.

[22] Keith Salley and Daniel T. Shanahan. Phrase Rhythm in Standard Jazz Repertoire: A Taxonomy and Corpus Study. *Journal of Jazz Studies*, 11(1):1, 2016.

[23] Daniel Shanahan, Yuri Broze, and Richard Rodgers. A Diachronic Analysis of Harmonic Schemata in Jazz. In *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*, pages 909–917, 2012.

[24] Mark J. Steedman. A Generative Grammar for Jazz Chord Sequences. *Music Perception: An Interdisciplinary Journal*, 2(1):52–77, 1984.

[25] Mark J Steedman. The blues and the abstract truth: Music and mental models. *Mental models in cognitive science: essays in honour of Phil Johnson-Laird*, pages 305–318, 1996.

[26] Hiroaki Tsushima, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Function- and Rhythm-Aware Melody Harmonization Based on Tree-Structured Parsing and Split-Merge Sampling of Chord Sequences. *Proceedings of the Tenth International Conference on Music Information Retrieval (ISMIR)*, pages 205–208, 2017.

[27] Hiroaki Tsushima, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Generative statistical models with self-emergent grammar of chord sequences. *Journal of New Music Research*, 47(3):226–248, May 2018.