REPETITION-STRUCTURE INFERENCE WITH FORMAL PROTOTYPES

Christoph Finkensiep[1,2] **Matthieu Haeberle**[1] **Friedrich Eisenbrand**[1]
Markus Neuwirth[3] **Martin Rohrmeier**[1]
[1] École Polytechnique Fédérale de Lausanne, Switzerland
[2] University of Amsterdam, The Netherlands (corresponding author: c.finkensiep@uva.nl)
[3] Anton Bruckner Privatuniversität Linz, Austria

ABSTRACT

The concept of form in music encompasses a wide range of musical aspects, such as phrases and (hierarchical) segmentation, formal functions, cadences and voice-leading schemata, form templates, and repetition structure. In an effort towards a unified model of form, this paper proposes an integration of repetition structure (i.e., which segments of a piece occur several times) and formal templates (such as AABA). While repetition structure can be modeled using context-free grammars, most prior approaches allow for arbitrary grammar rules. Constraining the structure of the inferred rules to conform to a small set of templates (meta-rules) not only reduces the space of possible rules that need to be considered but also ensures that the resulting repetition grammar remains interpretable in the context of musical form. The resulting formalism can be extended to cases of varied repetition and thus constitutes a building block for a larger model of form.

1. INTRODUCTION

Repetition is one of the most central aspects of music [1] and constitutes a constant across almost all cultures, styles and genres. The repetition of material is one of the major compositional devices for the arrangement of parts in overarching musical form [2, 3, 4], be it a folksong, a minuet, a sonata, a jazz standard, or a pop song. In general, musical form could be characterized in terms of exhaustive segmentation, hierarchical grouping structure, rhythmic-hypermetrical structuring, the form functionality of segments [3], and repetition structure. For the purpose of this paper, three aspects of form are considered: a hierarchical organization [5], which is also reflected in hierarchical harmonic structure [6]; repetition of formal constituents, which is one of the most prominent and salient features of form perception in human music cognition [1]; and prototypes of formal organization (such as AABA) which can characterize classical forms [3] but are also common structures in pop, jazz, and folk songs.

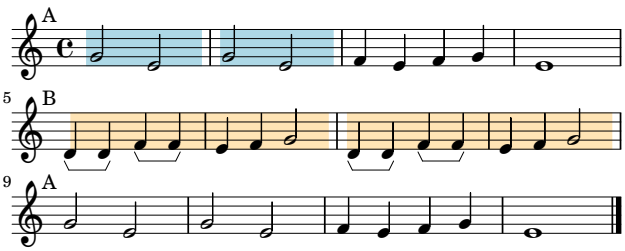


Figure 1: A German 19th-century folksong melody on the lyrics "Stille, stille, kein Geräusch gemacht" or "Bier her, Bier her, oder ich fall um."

The GTTM [5] defines *grouping structure* in terms of a tree of hierarchical containment relations that provides an exhaustive segmentation of the piece. GTTM's preference rules for grouping structure include Gestalt principles [7] as well as repetition. In addition to grouping structure, repetition structure is defined as a hierarchical grouping tree that captures (optimal) reuse of material (exact or in variation) in terms of groups of musical units and recursive groups of groups. Repetition structure provides a full grouping of a piece, however, it may potentially result in a different tree than what is obtained by a general formal analysis of a piece (see Figure 6). For human judgement of form in general, repetition is not the only factor, as features of (hyper-)metrical structure, form functions, or harmony may play a role as well (see also below in section 4.2). Accordingly, the objective of a computational model of repetition structure as an aspect of musical form may ultimately require to take such aspects into account as well.

Repetition structure plays a role within a single piece as well as over a corpus of pieces since abstract repetition patterns generalize over a whole dataset or style. The melody shown in Figure 1, for example, exhibits repetition of parts on several levels: On the highest level, the melody follows an ABA form, as the first four measures are literally repeated at the end (mm. 9-12). The B part (mm. 5-8) itself consists of a repetition of a two-measure phrase (yellow). Similarly, the first measure of the A part is repeated in the second measure (blue). Even on the level of individual notes, the direct repetition of a note is a prominent feature of mm. 5 and 7.

In the context of form, repetition structure refers to the re-occurrence of formal constituents (such as phrases and sections) that form a hierarchical segmentation structure,

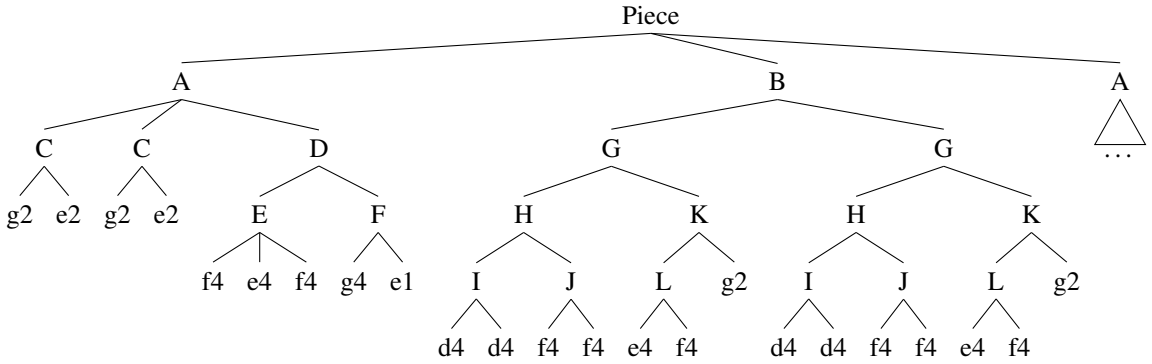


Figure 2: A possible repetition tree of the melody in Figure 1. The leaves of the tree encode pitch and duration of the melody notes. The second occurence of part A is identical to the first (not shown here).

as opposed to motivic or thematic material, for example. An example of such a segmentation structure for the example piece (Figure 1) is shown in Figure 2. Every formal segment of the piece corresponds to a subtree in the repetition tree. Note that all occurrences of the same segment share the same label and have exactly the same subtree structure. For this reason, the repetition structure shown in the tree can be more compactly described as a restricted context-free grammar (CFG) which has one non-terminal symbol for each segment (with terminal symbols for the notes of the piece) and exactly one rule for each symbol, encoding the decomposition of the corresponding segment. This relationship between the repetition structure of a piece and a compact representation of the piece as a CFG has been utilized for compression-based pattern discovery algorithms such as SEQUITUR [8].

Another aspect of the repetition tree shown in Figure 2 is that its rules use a limited set of formal prototypes, such as $\alpha\beta\alpha$ (e.g., Piece $\longrightarrow$ ABA), $\alpha\alpha\beta$ (A $\longrightarrow$ CCD), or $\alpha\alpha$ (B $\longrightarrow$ GG). In order to avoid confusion with the letters for specific form parts, we denote these form templates with greek letters, e.g., $\alpha\beta\alpha$, $\alpha\alpha\beta$, or $\alpha\alpha$. A concrete instance of a form template is denoted by applying the template to specific segments: CCD $= \alpha\alpha\beta(\mathrm{C}, \mathrm{D})$. While the rules in a piece's repetition grammar are specific to a particular segment in that particular piece, the form templates establish a relation between different rules with the same shape, within the same piece or across different pieces. We therefore call them *meta rules*.

This paper is a contribution towards an integrated computational model of musical form, combining two important aspects of form: repetition and formal prototypes. The model characterizes the relationship between meta rules and hierarchical repetition structure and provides a *proof of concept* algorithm and evaluation for repetition structure inference based on *minimal description length* [9].

2. RELATED WORK

Identification of repetition structure is closely related to compression, as identification of redundant information is important to achieve shorter encodings. An early example of grammar-based compression is SEQUITUR, an algorithm that infers a (not globally optimal) grammar for a given sequence in linear time [8, 10]. For an overview of approximate grammar-based compression, see [11, 12]. Besides inference of segmentation structure, grammar-based compression algorithms have been used for tasks such as error detection and tune classification [12, 13]. The principle of minimum description length has also been used outside of grammar-based approaches, e.g., in combination with hidden Markov models [14]. The approach presented in this paper differs from previous smallest-grammar approaches in two ways: the shape of the grammar rules is not arbitrary but constrained to a set of formal prototypes, and this constrained model is evaluated by inferring the global optimum instead of an approximation, which is generally NP-hard and thus only feasible for short sequences.

The segmentation structure of a piece can also be inferred based on criteria other than repetition. The GTTM [5] defines grouping structure based on a set of well-formedness and preference rules for recursively combining events into larger segments. In the MIR community, the analysis of musical form is known as *music structure analysis* (MSA) [15, 16, 17, 18, 19, 20, 21, 22]. MSA comes in a variety of tasks, involving boundary detection, (hierarchical) segmentation, the identification of segment labels and relations, and combinations of these tasks. While MSA uses a wide spectrum of supervised and unsupervised methods, from matrix factorization to deep learning, the definition of musical form in this context is usually given implicitly in the form of a dataset (e.g., [21, 22]) on which the model may be trained, and on which it is evaluated. The present paper, in contrast, presents a theoretical contribution towards an explicit definition of musical form, and the resulting model is not intended as a solution to a computational problem, such as performing a general segmentation and labeling task. As a consequence, our evaluation focuses on exploring the characteristic properties of the model.

3. METHODS AND DATA

3.1 Problem Description

A specific repetition structure for a given piece can be characterized through a piece-specific context-free gram-

$m_1 : \alpha\alpha$	$m_2 : \alpha\beta$
$m_3 : \alpha\alpha\alpha$	$m_4 : \alpha\beta\alpha$
$m_5 : \alpha\alpha\beta$	$m_6 : \alpha\beta\beta$
$m_7 : \alpha\alpha\beta\alpha$	$m_8 : \alpha\beta\beta\alpha$

Table 1: The set of meta rules used in this paper.

mar that generates exactly one string — the piece. We call such a grammar a *local grammar* for the piece. It consists of:

- a set of terminal symbols T, corresponding to the unique atomic segments of the piece; [1]
- a set of non-terminal symbols N, corresponding to the unique composite segments of the piece;
- a starting symbol P that stands for the full piece;
- a set of production rules R.

Since each non-terminal symbol stands for a specific segment, R contains exactly one rule for each non-terminal symbol, signifying the decomposition of the segment and enforcing that all occurrences of the segment are decomposed identically. As a consequence, the rules are not allowed to be (mutually) recursive since a segment cannot contain itself as a proper subsegment. [2]

In order to establish a relation between local repetition grammars and general formal prototypes, the right-hand side (RHS) of each rule must be an instance of a *meta rule*. Meta rules are generally of the shape $\{\alpha, \beta, \gamma, \ldots\}^+$ and are instantiated by creating a bijective mapping between letters and specific non-terminal symbols. For example, the meta rule $\alpha\alpha\beta\alpha$ encodes the formal prototype AABA and can be instantiated as

$$\alpha\alpha\beta\alpha(S,T) = \alpha\alpha\beta\alpha\{\alpha \mapsto S, \beta \mapsto T\} = SSTS \quad (1)$$

where $S \neq T$. Thus, a local grammar can express that a piece has an overarching AABA structure by using a rule

$$P \longrightarrow \alpha\alpha\beta\alpha(S,T) \quad (2)$$

that takes the starting symbol P to an instance of $\alpha\alpha\beta\alpha$ with $\alpha = S$ and $\beta = T$. The set of meta rules can be chosen freely to encode a set of typical formal prototypes. The meta rules used in the following experiments are shown in Table 1.

The goal of repetition structure inference is to find a local grammar for a given piece according to some optimality criterion, such as musical plausibility, probability, or description length (DL). In accordance with prior approaches that use repetition grammars, our proof-of-concept implementation searches for local grammars with minimum description length, defined in analogy to [13] by counting the symbols needed to encode the grammar:

$$DL(R) = \sum_{r \in R} 2 + |\text{params}(r)| \quad (3)$$

where $\text{params}(r)$ denotes the parameters of the meta rule on the RHS of rule r. That is, for each rule we count one symbol for the meta rule, one symbol for each parameter of the meta rule, and one separator symbol [3] marking the end of the rule. For example, the rule in Equation 2 has a description length of 4: one meta-rule symbol ($\alpha\alpha\beta\alpha$ or m_7) [4], two parameters (S and T) and the separator. It is not necessary to encode the left-hand side (LHS) of a rule since there exists a canonical order of rules, starting with the rule for P and then listing the rules in the order in which their LHS symbols are introduced on the RHS of other rules.

3.2 Algorithm

The minimal grammar for a given piece is found in a two-stage process. First, a set of possible rules for each unique segment of the piece is computed. Second, a set of rules is selected from these candidates, ensuring that the resulting grammar is consistent and minimizing the cost of the selected rules.

Algorithm 1 Enumerating all rule candidates.

```
1:  function PARSE(input)
2:      subs ← uniqueSubsequences(input)
3:      chart ← {}
4:      for seq ∈ sortByLength(subs) do
5:          for s from 1 to |seq| − 1 do
6:              cs ← COMPLETE(seq[: s], seq[s + 1 :])
7:              chart[seq] ← cs
8:      return chart

9:  function COMPLETE(left, right)
10:     il ← chart[left]_incomplete
11:     ir ← chart[right]_incomplete
12:     bs ← binaryRules(left, right)
13:     is ← incompleteConstituents(left, right, il, ir)
14:     ns ← nAryRules(left, right, il, ir)
15:     return (complete = bs ∪ ns, incomplete = is)
```

The first stage (Algorithm 1) begins with collecting all unique subsegments of the piece. For each of these subsegments, all possible decompositions according to the meta rules are computed using dynamic programming, analogous to the CYK algorithm: The segment is split at every possible split point (l. 5), generating two subsegments left and right of the split point. For binary meta rules ($\alpha\alpha$ and $\alpha\beta$), an instance of the rule can be identified directly by comparing the subsegments (l. 12). Meta rules

[1] In the case of melodies, these atomic segments correspond to notes and rests, but they could also correspond to polyphonic events (slices), or previously annotated elementary phrases.

[2] A unary identity rule (e.g. $X \longrightarrow X$) is not permitted. Other unary rules are not possible because of the one-to-one correspondence between segments and grammar symbols.

[3] The separator is not strictly necessary since the length of the rule is known from the meta rule, but it is included here to stay as close as possible to [13].

[4] $\alpha\alpha\beta\alpha$ is counted as one symbol since the set of meta rules is assumed to be fixed and cannot be freely extended.

of higher arity are decomposed into binary parts. For example, the meta rule $\alpha\beta\alpha$ can be decomposed into an incomplete constituent $\alpha\beta^*$ and another α. When a segment $S = S_1S_2$ has a decomposition into $\alpha\beta(S_1, S_2)$, it additionally stores an $\alpha\beta^*(S_1, S_2)$ item (l. 13). At a later point, a larger segment $T = SS_3 = S_1S_2S_3$ retrieves the item $S \longrightarrow \alpha\beta^*(S_1, S_2)$, checks whether $S_3 = S_1$, and accordingly stores a rule $T \longrightarrow \alpha\beta\alpha(S_1, S_2)$. Similarly, all larger meta rules are constructed from incomplete constituents such as $\alpha\beta^*$ and $\alpha\alpha^*$ (l. 14).

Once the possible decompositions of each subsequence are known, the second stage of the algorithm converts the set of possible rules into an integer linear program (ILP) which then extracts a set of rules with minimal cost. Each subsequence of the input of lenght ≥ 2 corresponds to a potential non-terminal symbol, so one binary indicator variable s_{symb} for each symbol $symb$ encodes the inclusion of the symbol in the grammar. Similarly, the inclusion of each candidate rule is indicated by a binary variable s_{rule}. The optimization problem is then given by

$$\begin{aligned} \min_{\substack{r\in\{0,1\}^{|rules|}\\ s\in\{0,1\}^{|symbols|}}} \quad & \sum_{rule\in rules} r_{rule} \cdot DL(rule) \\ \text{s.t.} \quad & s_{symb} = \sum_{\substack{rule\in rules\\ \text{LHS}(rule)=symb}} r_{rule} \\ & r_{rule} \leq \sum_{symb\in\text{RHS}(rule)} \frac{s_{symb}}{|\text{RHS}(rule)|} \\ & s_{start} = 1. \end{aligned} \tag{4}$$

Two constraints define the relationship between symbols and rules: each included symbol requires exactly one corresponding rule; and each included rule requires the symbols on its right-hand side. [5] A third constraint requires the presence of the starting symbol which corresponds to the full input sequence. The rules and symbols are then selected by minimizing the total cost of the included rules as defined in Equation 3.

4. RESULTS AND DISCUSSION

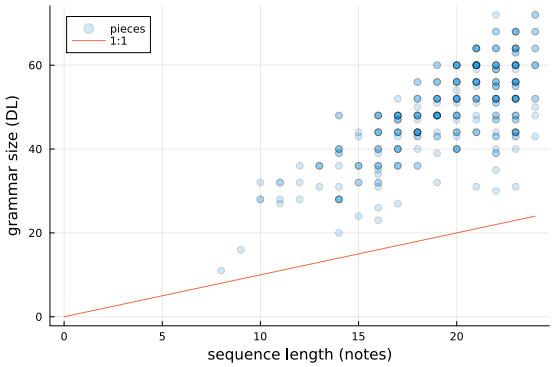
4.1 Quantitative Evaluation on a Dataset

For evaluating the above approach, we infer the minimal grammars (under the meta rules from Table 1) for the 298 shortest melodies from the Essen folksong collection [23], with a length of 8 to 24 notes. The melodies are represented as sequences of notes (including rests), consisting of pitch (or a rest symbol) and duration. Other aspects, such as the position of a note in a measure, are not taken into account. The minimization algorithm is implemented in Julia and is available online. [6] For ILP optimization we use the JuMP framework [24] together with the Gurobi solver backend. [7]

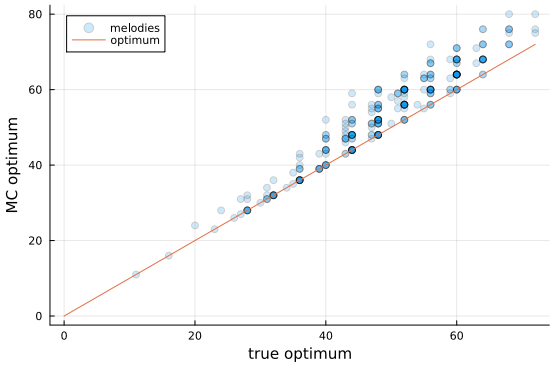
[5] The logical conjunction of the RHS symbols is expressed as a normalized sum instead of a product in order to maintain linear relationships between the variables in the program.

[6] https://github.com/DCMLab/form-repetition-ismir23

[7] Gurobi requires a license, which is provided freely for academic purposes. Alternatively, the JuMP framework supports using different solver backends.



(a) Grammar size vs. input length.

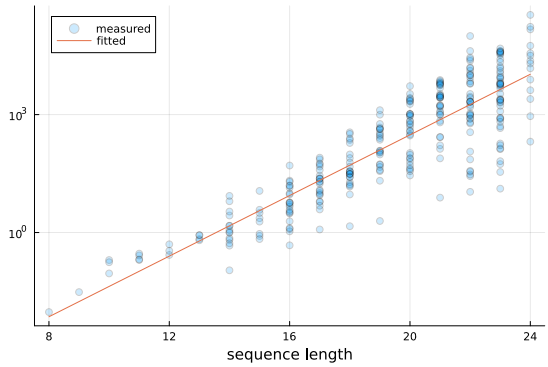


(b) Optimal grammar size vs. Monte-Carlo minimum.

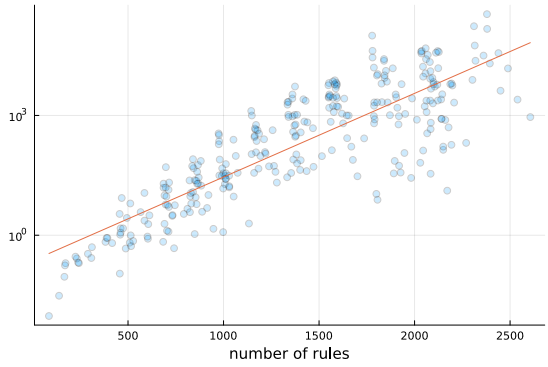
Figure 3: Comparison of the description length of the minimal local grammars to (a) the input sequence length and (b) local grammars obtained through Monte-Carlo minimization.

Since the local repetition grammar formalism is not designed to obtain optimal compression of the input sequence (but uses description length as a rather arbitrary proxy for the plausibility of a specific segmentation), we cannot expect very good compression rates. Indeed, when comparing the length of the input sequences to the total description length of the corresponding minimal grammar, the grammars are usually larger than the original piece (Figure 3a) with a average ratio of 2.47 (geometric mean). This indicates that restricting the grammars to a small set of meta rules is not sufficient to achieve an actual compression of the dataset, at least when only considering exact repetition.

Since finding the global minimum is expensive (see below), most grammar-based compression algorithms only attempt to approximate the global optimum [8, 12, 13]. We estimate the payoff of inferring the global optimum by comparing the optimal description lengths to approximate solutions obtained by a Monte-Carlo minimization process: Beginning with the start symbol, the rule for each required symbol is chosen randomly from the set of possible rules, and the corresponding RHS symbols are added to the list of required symbols. This process is repeated until all required symbols are covered. Out of 10,000 randomly sampled grammars for each piece, the smallest grammar is selected. The results are shown in Figure 3b. For the

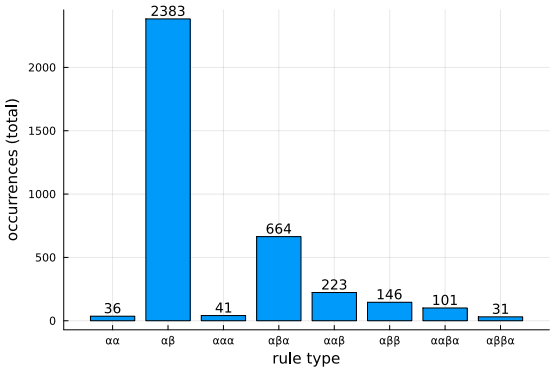


(a) Runtime relative to input length.

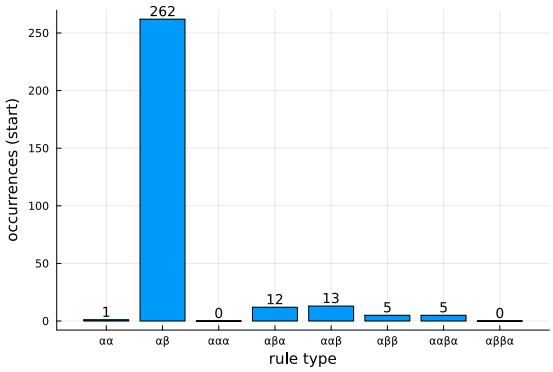


(b) Runtime relative to possible rules.

Figure 4: The measured runtime of the optimization step relative to (a) the length of the input sequence and (b) the number of possible rules for the sequence. Note that in both cases, the time axis is scaled logarithmically, so the fitted exponential curves appear as straight lines.



(a) Overall meta rule usage.



(b) Meta rules used at the top of the form tree.

Figure 5: The meta rules used in the inferred minimal grammars for the melodies in the dataset.

given dataset, the Monte-Carlo minimum is on average 1.08 times longer than the true grammar. In many cases, the Monte-Carlo process finds a true optimum, since the sample size of 10,000 is large enough to find an optimal solution by chance. However, with growing input size, the range of possible grammars grows exponentially, in the worst case.[8] So, while a Monte-Carlo estimate can be a useful approximation on short sequences, it cannot keep up with the size of the search space for longer sequences, unless the sample size is increased exponentially as well.

The runtime behavior of the optimization problem is shown in Figure 4. The problem of finding an unrestricted minimal CFG is known to be NP-hard [13]. The runtime for the restricted case relative to the input length is shown in Figure 4a with logarithmic scaling. Since the actual size of the optimization problem depends not only on the input length but also on the amount of redundancy within the sequence, Figure 4b shows the runtime relative to the number of possible rules obtained in the first stage of the algorithm. In both cases, the runtime grows approximately exponentially with the number of rules. This is supported by an exponential regression in both figures, fit as a linear function in logarithmic space which minimizes the squared *ratio* between measured and predicted runtime instead of the squared difference.

The distribution of meta rules in the inferred grammars is shown in Figure 5a. By far the most common rule type is $\alpha\beta$, which is not surprising since it is the only rule type that does not require any form of repetition. The rule type $\alpha\alpha$ is used very infrequently, which may seem surprising due to its simplicity. However, all other rules (except for $\alpha\beta\beta\alpha$ and $\alpha\alpha\alpha$ which are similarly rare) have one part that does not need to be repeated and are thus applicable to a wider range of situations. The distribution of starting-rule types is shown in Figure 5b. These rule types correspond to the overarching form of the melody in terms of exact repetition. The even stronger prevalence of $\alpha\beta$ in this case indicates that there is very little exact repetition on the highest form level in the given dataset of melodies, which might be biased due to the focus on short melodies. On the other hand, this lack of repetition indicates that a model of formal segmentation cannot be exclusively based on repetition but needs to take into account at least the possibility of varied repetition, as well as other markers of form such as cadences and meter.

4.2 Qualitative Evaluation on an Example Melody

Table 2 displays the minimal grammar for the example piece in Figure 1. Compared to the overall distribution of meta rules, the grammar uses many repeating rules, which reveals that the piece features an unusual amount of inter-

[8] The number of subsequences grows quadratically, and the number of possible grammars is a product over all substrings.

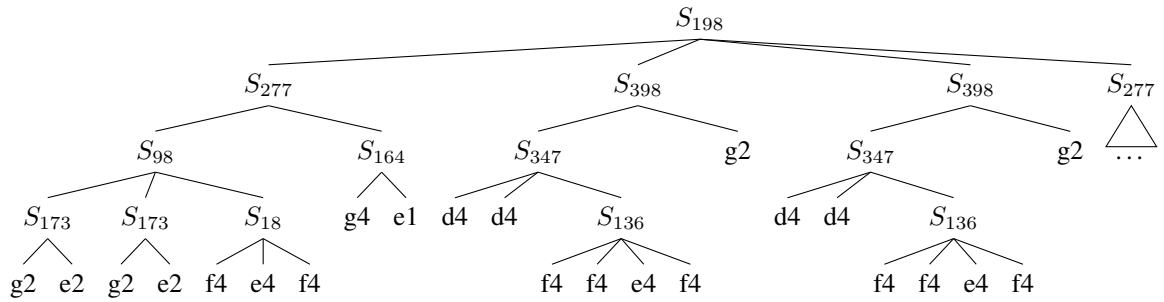


Figure 6: The minimal tree for the example piece in Figure 1.

rule		meta rule	cost
r_1	$S_{198} \longrightarrow S_{277}\ S_{398}$	$\alpha\beta\beta\alpha(S_{277}, S_{398})$	4
r_2	$S_{277} \longrightarrow S_{98}\ S_{164}$	$\alpha\beta(S_{98}, S_{164})$	4
r_3	$S_{398} \longrightarrow S_{347}\ g2$	$\alpha\beta(S_{347}, g2)$	4
r_4	$S_{98} \longrightarrow S_{173}\ S_{18}$	$\alpha\alpha\beta(S_{173}, S_{18})$	4
r_5	$S_{164} \longrightarrow g4\ e1$	$\alpha\beta(g4, e1)$	4
r_6	$S_{347} \longrightarrow d4\ S_{136}$	$\alpha\alpha\beta(d4, S_{136})$	4
r_7	$S_{173} \longrightarrow g2\ e2$	$\alpha\beta(g2, e2)$	4
r_8	$S_{18} \longrightarrow f4\ e4$	$\alpha\beta\alpha(f4, e4)$	4
r_9	$S_{136} \longrightarrow f4\ e4$	$\alpha\alpha\beta\alpha(f4, e4)$	4

Table 2: The minimal grammar for the example piece in Figure 1.

nal repetition. As the derivation tree in Figure 6 shows, the optimal solution found by the algorithm captures many aspects of the human intuition. Similar to the hand-annotated segmentation in Figure 2, the minimal tree captures the overarching repetition of mm. 1-4 and mm. 9-12 as well as mm. 5-6 and mm. 7-8. However, whereas the human intuition groups the single note repetitions together and splits non-repeating segments according to bar units (mm. 5, 7), the algorithm finds that other groupings provide an even more economic description length in terms of rule usage, which leads to a somewhat counter-intuitive dangling half note g at the end of the phrase. This illustrates that human decisions in terms of repetition structure do not purely optimize repetition, but that they take rhythmic-metric boundaries into account. Therefore, the objective of a model of repetition structure that captures or comes close to the human intuition needs to be further developed to also incorporate such features. A candidate model may be the hierarchical model of rhythmic structure as a formal grammar [25].

5. CONCLUSION

In this paper we have presented a computational model of musical repetition structure as an aspect of musical form. Since repetition structure is an aspect of human music cognition, the overarching objective of our approach is to approach human listening. The model captures repetition structure with a special form of context-free grammar, in which the rewrite of each category is only defined once such that it captures a unique repeating fragment of a given piece. A set of meta-rules defines the generic types of repetition patterns that could occur within a piece. The model is very generic and can also be applied to more complex textures as well as music of all styles and cultures, as long as a representation as a sequence of symbols is meaningful.

Inferring the optimal grammar with respect to a suitable objective criterion (such as description length) is able to effectively capture the repetition structure in a piece. Other objective criteria (e.g., prior probabilities of meta rules) can be used in a similar way since the algorithm does not depend on a fixed cost function. On the other hand, maximizing the redundancy that is captured by a segmentation does not ensure that the segmentation is a good analysis of the form of a piece. For one, not all repetition and reuse of material is exact, which is evident from the low proportion of repeating meta rules used in the example dataset. Simple forms of varied repetition could be integrated in our model relatively easily: given a suitable measure of similarity, not only identical segments are grouped together but also sufficiently similar segments. More sophisticated versions of this model could capture how variations are produced through the generative process of the grammar (e.g., by making different decisions in different subtrees), or how only certain aspects of a segment are repeated while others change (e.g., using the same rhythm with a different melodic contour). Furthermore, even when all repetitions are exact (as in the example piece), capturing repetition is not the only criterion for grouping tokens into formal segments, as other criteria such as cadences, rhythm and meter, formal function, or harmonic and contrapuntal schemata interact with grouping as well.

The runtime complexity of finding the smallest grammar for a given piece is generally exponential. For sufficiently short pieces, exact inference can be approximated probabilistically, but there is no guarantee that the resulting suboptimal grammars resemble the true optimum. For larger inputs, the search space grows exponentially, so naive Monte-Carlo approximation can become arbitrarily bad. This indicates that further research is required to find plausible estimates of formal structure, integrating the technical aspect of optimization with the musical problem of defining what constitutes a plausible analysis.

6. 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. This research was supported by the Swiss National Science Foundation within the project "Distant Listening – The Development of Harmony over Three Centuries (1700–2000)" (Grant no. 182811). The authors thank Mr. Claude Latour for generously supporting this research.

7. REFERENCES

[1] E. H. Margulis. *On Repeat: How Music Plays the Mind*. Oxford, New York: Oxford University Press, Feb. 6, 2014. 224 pp.

[2] F. Diergarten and M. Neuwirth. *Formenlehre. Ein Lese- Und Arbeitsbuch Zur Instrumentalmusik Des 18. Und 19. Jahrhunderts*. 2nd ed. Laaber-Verlag, 2020.

[3] W. E. Caplin. *Classical Form: A Theory of Formal Functions for the Instrumental Music of Haydn, Mozart, and Beethoven*. Oxford University Press, May 14, 1998. 320 pp.

[4] Y. Greenberg. *How Sonata Forms: A Bottom-Up Approach to Musical Form*. Oxford Studies in Music Theory. Oxford, New York: Oxford University Press, June 10, 2022. 264 pp.

[5] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT press, 1983.

[6] M. Rohrmeier. "The Syntax of Jazz Harmony: Diatonic Tonality, Phrase Structure, and Form". In: *Music Theory and Analysis (MTA)* 7.1 (Apr. 30, 2020), pp. 1–63. DOI: 10.11116/MTA.7.1.1.

[7] M. Wertheimer. "Laws of Organization in Perceptual Forms". In: *A source book of Gestalt Psychology* 1 (1923).

[8] C. G. Nevill-Manning and I. H. Witten. "Identifying Hierarchical Structure in Sequences: A Linear-Time Algorithm". In: *Journal of Artificial Intelligence Research* 7 (Sept. 1, 1997), pp. 67–82. DOI: 10.1613/jair.374.

[9] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Sept. 25, 2003. 694 pp.

[10] C. G. Nevill-Manning and I. H. Witten. "Compression and Explanation Using Hierarchical Grammars". In: *The Computer Journal* 40 (2_and_3 Jan. 1997), pp. 103–116. DOI: 10.1093/comjnl/40.2_and_3.103.

[11] E. Lehman and A. Shelat. "Approximation Algorithms for Grammar-Based Compression". In: *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics., 2002, pp. 205–212.

[12] K. A. Sidorov, A. Jones, and A. D. Marshall. "Music Analysis as a Smallest Grammar Problem." In: *ISMIR*. 2014, pp. 301–306.

[13] D. Humphreys, K. Sidorov, A. Jones, and D. Marshall. "An Investigation of Music Analysis by the Application of Grammar-Based Compressors". In: *Journal of New Music Research* 50.4 (Aug. 8, 2021), pp. 312–341. DOI: 10.1080/09298215.2021.1978505.

[14] P. Mavromatis. "Minimum Description Length Modelling of Musical Structure". In: *Journal of Mathematics and Music* 3.3 (Nov. 1, 2009), pp. 117–136. DOI: 10.1080/17459730903313122.

[15] M. Buisson, B. Mcfee, S. Essid, and H.-C. Crayencour. "Learning Multi-Level Representations for Hierarchical Music Structure Analysis". In: International Society for Music Information Retrieval (ISMIR). Dec. 4, 2022.

[16] B. McFee. "Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications". In: 3.1 (1 Dec. 11, 2020), pp. 246–263. DOI: 10.5334/tismir.54.

[17] B. McFee and D. Ellis. "Analyzing Song Structure with Spectral Clustering." In: *ISMIR*. Citeseer, 2014, pp. 405–410.

[18] F. Kaiser and T. Sikora. "Music Structure Discovery in Popular Music Using Non-negative Matrix Factorization." In: *ISMIR*. 2010, pp. 429–434.

[19] M. Levy and M. Sandler. "Structural Segmentation of Musical Audio by Constrained Clustering". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.2 (Feb. 2008), pp. 318–326. DOI: 10.1109/TASL.2007.910781.

[20] M. C. McCallum. "Unsupervised Learning of Deep Features for Music Segmentation". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). May 2019, pp. 346–350. DOI: 10.1109/ICASSP.2019.8683407.

[21] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie. "Design and Creation of a Large-Scale Database of Structural Annotations." In: *ISMIR*. Vol. 11. Miami, FL, 2011, pp. 555–560.

[22] C.-i. Wang, G. J. Mysore, and S. Dubnov. "Re-Visiting the Music Segmentation Problem with Crowdsourcing." In: *ISMIR*. 2017, pp. 738–744.

[23] H. Schaffrath. "The Essen folksong collection in the Humdrum Kern Format (D. Huron, Ed.)" In: *Menlo Park, CA: Center for Computer Assisted Research in the Humanities* (1995).

[24] M. Lubin, O. Dowson, J. D. Garcia, J. Huchette, B. Legat, and J. P. Vielma. "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization". In: *Mathematical Programming Computation* (2023). In press.

[25] M. Rohrmeier. "Towards a Formalization of Musical Rhythm". In: *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*. Ed. by J. Cumming, J. H. Lee, B. McFee, M. Schedl, J. Devaney, C. McKay, E. Zangerle, and T. de Reuse. 2020, pp. 621–629.