Finding Grammar in Music by Evolutionary Linguistics

Hiroki Sudo
Japan Advanced Institute
of Science and Technology,
Research Fellow of the JSPS,
Email: hiroki.sudo@jaist.ac.jp

Masaya Taniguchi
Japan Advanced Institute
of Science and Technology,
Email: taniguchi@jaist.ac.jp

Satoshi Tojo
Japan Advanced Institute
of Science and Technology,
Email: tojo@jaist.ac.jp

***Abstract*—In this paper, we assume that the progression rules of music are in a subclass of context-free language, and we let computers find them autonomously. We employ the Iterated Learning Model (ILM) by Simon Kirby, and ask if the computer can find a music knowledge that is common to us, and also if the computers can compose music independently of our music knowledge. In this research, we have shown an example set of rules found in the 25 études of Burgmüller by beat. Although many of categories in the tree seem redundant and futile, some of them reflect probable progressions, which well match with our human intuition. This experiment has several virtues compared with other grammar-based formalism for music. One is that we do not need to provide a dictionary beforehand. The other is that we can exclude the human-biased intuition, which had hindered the definition of creativity.**

I. Introduction

It is generally accepted that the origin of language and that of music are one and the same [12], [14]; we employ throat to utter or to sing, and ears to hear, and furthermore we are said to use the same parts of our brain to articulate them. We say birds sing, but this is only a metaphor; in most of the species only male birds utter languages to woo female birds to bear descendants

Thus far, many linguists and musicologists, as well as computer scientists, have tried to find a grammar structure in music [7]–[9], [11], [13]. Some of them are successful enough to analyze certain genre of music, *e.g.*, we can find the rules of chord progression in context-free grammar for early classicist music. Among which, the approach from the Generative Theory of Tonal Music (GTTM) [3]–[6] seems worth noting as the theory has had the viewpoint of Chomskian hierarchy of tree structure. However, as many musicologists would agree, the syntactic rule for music is quite loose compared with languages, to agree rather spontaneous change of rules. Furthermore, the history of music has devoted to deviate from the old tradition, to tolerate freer framework. Therefore, it seems less fruitful for us to fix a certain class of formal language, which plausibly resides between context-free and context-sensitive grammar in Chomsky hierarchy, in music.

In traditional linguistics, the main stream has been set to distinguish the theory of syntax and that of semantics, aside from phonetics and pragmatics. We regard that the discussion on what is the meaning of music is beyond the current scope; instead, we consider that the tree structure owes a part of what we call meaning. Also, we contend that the syntax is the universal generative faculty of human beings, that is the source of expressive power common in language, mathematics, morality, and music.

Cope has discussed in [1], [2] the creativity by computers, in which he has claimed that we human are biased to degrade the computer music only because they are composed by the computer. Again, we would avoid the discussion on what is the creativity, but we simply try to make the computer compose music in a quite naïve sense, disregarding its quality.

In this paper, we assume that the progression rules of music are in a subclass of context-free language, and we let computers find them autonomously. We employ the Iterated Learning Model (ILM) by Simon Kirby [10]. The research questions of us are two-fold; the first is if the computer can find a music knowledge that is common to us without listening to music. The second is if the computers can compose music independently of the music knowledge, that is familiar to us.

This paper is composed of as follows. In the following section 2, we introduce our methodology of ILM, where those basic operations as *chunk*, *merge*, and *replace* are introduced. In Section 3, we adapt the ILM to our objective, that is, we abandon the predicate-argument structure (PAS) for the meaning and instead we adopt our own categorical structure: labeled context-free grammar. In Section 4, we discuss how we evaluate our method. In Section 5, we show our experimental result and in Section 6 we conclude.

II. Cultural Evolution of Language

Infants learn language from their parents, without observing the internal language of them, that is the innate but hidden grammar formalism. As a result, the infants may acquire a different language from their mothers. In Fig. 1, the baby guess possible chunking from the few sample sentences given by his/her mother.

The iterated learning model (ILM) suggests that the infant grows up to be the teacher of the language in the successive generation; since the number of the sentences given by the parents is limited, this bottleneck is said to contribute to increase the number of the expressible meanings (Fig. 2).

In ILM, agent's knowledge represented by Labeled Context Free Grammar (LCFG) as follows.

Fig. 1. Chunking by Infants

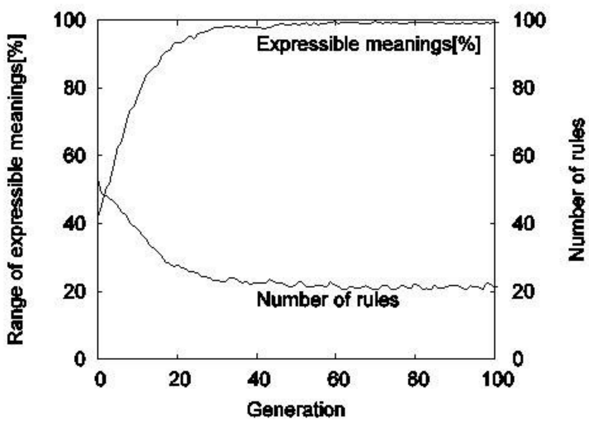

Fig. 2. Expressible Meanings by Generation

Labeled Context Free Grammar G

$G = (N, T, M, V, P, S)$
N : Non-terminal symbols
T : Terminal symbols
M : Predefined meanings (PAS)
V : Variables for M
P : A set of rules
$\alpha \to \beta \left(\alpha \in N \times (M \cup V), \beta \in ((N \times V) \cup T)^+\right)$
S : Start non-terminal symbol $S \in N$
α must have all variables in β.

The three operations employed in the ILM are as follows.

- Chunk

$$
\begin{aligned}
&S/like(mary, john) \to marylikesjohn \\
&S/love(mary, john) \to marylovesjohn \\
&\qquad\qquad \downarrow chunk \\
&S/X_1(mary, john) \to mary \;\; N_0/X_1 \;\; esjohn \\
&N_0/like \to lik \\
&N_0/love \to lov
\end{aligned}
$$

- Merge

$$
\begin{aligned}
&S/hate(X_2, X_3) \to N_1/X_2 \;\; hates \;\; N_2/X_3 \\
&N_1/gavin \to gavin \\
&N_1/pete \to pete \\
&N_2/gavin \to gavin \\
&\qquad\qquad \downarrow merge \\
&S/hate(X_2, X_3) \to N_2/X_2 \;\; hates \;\; N_2/X_3 \\
&N_2/gavin \to gavin \\
&N_2/pete \to pete
\end{aligned}
$$

- Replace

$$
\begin{aligned}
&S/admire(john, pete) \to johnadmirespete \\
&N_3/admire \to admire \\
&\qquad\qquad \downarrow replace \\
&S/X_1(john, pete) \to john \;\; N_3/X_1 \;\; pete \\
&N_3/admire \to admire
\end{aligned}
$$

III. Extension of ILM to Music Analysis

A model which we propose has two important features. One is importing categorial level into the model. This is to express categorial equivalence between different meanings like hyponymy and hypernymy. The other is unlimitation on number of arguments and variety of meanings. Number of arguments and variety of meanings are dynamically increased by learning of input data. In this model, a meaning is defined by agent's self. These features implement an unlimited hierarchical compositional meaning which has recursion. It is different from ILM which is using predefined meanings (PAS).

A. Categorial Level

We employ a set of attributes as an interpretation of a meaning "${m_i}^{\mathcal{I}}$" as follows.

$$\text{Ex. } {m_i}^{\mathcal{I}} = \{\langle measure\rangle, \langle I/F\rangle, \langle F\rangle\}$$

This interpretation means that m_i is a measure in a piece of music which has a chord "F" and a chord with represented by tonal "I/F" ("m_i is a $\langle measure\rangle$" $\wedge$ "m_i is a $\langle F\rangle$" $\wedge$ "m_i is a $\langle I/F\rangle$").

We define three operations ($\cap, \cup, -$) for interpretation of a meaning.

$$
\begin{aligned}
{m_i}^{\mathcal{I}} \cap {m_j}^{\mathcal{I}} &= \left\{x | x \in {m_i}^{\mathcal{I}} \wedge x \in {m_j}^{\mathcal{I}}\right\} \\
{m_i}^{\mathcal{I}} \cup {m_j}^{\mathcal{I}} &= \left\{x | x \in {m_i}^{\mathcal{I}} \vee x \in {m_j}^{\mathcal{I}}\right\} \\
{m_i}^{\mathcal{I}} - {m_j}^{\mathcal{I}} &= \left\{x | x \in {m_i}^{\mathcal{I}} \wedge x \notin {m_j}^{\mathcal{I}}\right\}
\end{aligned}
$$

Operation $\cap$, $\cup$ are the same as set theory. Operation $-$ is a calculation of diff, and explained by $A \backslash B$ in set theory ($A = {m_i}^{\mathcal{I}}, B = {m_j}^{\mathcal{I}}$). The interpretation of a new meaning is defined by these operation in learning.

B. Extension of Expression of Meanings

In ILM, a predefined meaning formed into PAS enables to map between a meaning and a symbol string in a knowledge. We define syntax for a meaning to express unlimited meaning as follows.

"$p(\epsilon)$" and "$p(args,\epsilon)$" is represented by "p" and "$p(args)$" as an abbreviation respectively.

$$\begin{aligned}\phi &::= p(args)\\ args &::= \phi \mid \phi, args \mid \epsilon\\ &\quad (Comma``,'' is a part of syntax.)\end{aligned}$$

Ex. $m_1(m_2(m_3, m_4(m_5), m_6), m_7(m_8))$
p is an atomic meaning.

$$\begin{aligned}\phi' &::= p(args')\\ args' &::= x' \mid x', args'\end{aligned}$$

x is a variable.

We call ϕ a meaning and ϕ represents a combination of grammar rules and categorial meaning. ϕ' represents a variable expression in a rule. Each variable matches ϕ.

A size of meaning sequence $|\phi|$ is defined as follows.

Ex. $|\phi| = |p(args)| = |p(\phi,\cdots)| = \langle$number of $\phi\rangle$
$|m_1(m_2(m_3, m_4(m_5), m_6), m_7(m_8))| = 2$
$|m_2(m_3, m_4(m_5), m_6)| = 3$

A size of ϕ is defined by learning, and possible to be an any number. It implies that structure of a meaning is un-predefined in our model.

A predefined meaning in ILM is represented by extended expression in our model (1). In following equation, m_i denotes unclear meanings of sentence and structure of PAS (The PAS needs three arguments.).

$$like(mary, john) \subseteq m_i(like, mary, john) \tag{1}$$

C. Application of Extensions to LCFG

We employed categorial level and unlimited expression of meaning. Agent's grammar also reflected these extensions as follows.

Extended Labeled Context Free Grammar G'

$$\begin{aligned}G' &= (N, T, M_a, V_a, P, S)\\ N &: \text{Non-terminal symbols}\\ T &: \text{Terminal symbols}\\ M_a &: \text{Atomic meanings}\\ V_a &: \text{Atomic variables}\\ M &: \text{Meanings formed into } \phi',\ p \in M_a, x \in V_a\\ P &: \text{A set of rules}\\ & \quad \alpha \to \beta\ (\alpha \in N \times M, \beta \in ((N \times V_a) \cup T)^+)\\ S &: \text{Start non-terminal symbol } S \in N\end{aligned}$$

Number of variables in α must be number of $(N \times V)$ in β.

M_a and V_a are not predefined and what an agent creates through learning. Therefore, size of M is possible to be infinite because syntax ϕ has recursion.

D. Extension of Learning

In ILM, after an agent has accepted a certain number of utterances, she tries to build a new set of generation rules. There are three operations *chunk*, *merge*, and *replace* based on the generalized method. In the conventional ILM, a rule is applicable when a pair of meaning and a part of a sequence of symbols in utterance matches. However, this condition is still ambiguous because the same local sequence of symbols may own different meanings. In our model, we define the applicable conditions by two interpretation of a meaning. Extended LCFG rule in agent's knowledge has the unique atomic meaning respectively. Thus, although every rule can possess a different meaning from others, these difference may not reflect the different sets of attributes. In our formalism, therefore, we can understand the sameness of the attributes by the interpretation.

- Chunk

  We supposes the conditions on chunk as follows.

$$\begin{aligned}{m_a}^{\mathcal{I}} \cap {m_b}^{\mathcal{I}} &\neq \emptyset\\ {m_1}^{\mathcal{I}} &= {m_a}^{\mathcal{I}} \cap {m_b}^{\mathcal{I}}\\ {m_2}^{\mathcal{I}} &= {m_a}^{\mathcal{I}} - {m_1}^{\mathcal{I}}\\ {m_3}^{\mathcal{I}} &= {m_b}^{\mathcal{I}} - {m_1}^{\mathcal{I}}\end{aligned}$$

  - For two no compositional rules

$$\begin{aligned}&S/m_a \to CDCCDEF^{\#}GC\\ &S/m_b \to CDCDEEF^{\#}GC\\ &\qquad\qquad \downarrow chunk\\ &S/m_1(x_1) \to CDC\ \ N_1/x_1\ \ EF^{\#}GC\\ &N_1/m_2 \to CD\\ &N_1/m_3 \to DE\end{aligned}$$

  - For compositional rule and no compositional one

$$\begin{aligned}&S/m_b(x_1) \to C^{\#}D\ \ N_1/x_1\ \ F^{\#}GC^{\#}\\ &S/m_a \to C^{\#}DCDEF^{\#}GC^{\#}\\ &\qquad\qquad \downarrow chunk\\ &S/m_1, x_1 \to C^{\#}D\ \ N_1/x_1\ \ F^{\#}GC^{\#}\\ &N_1/m_2 \to C^{\#}DE\end{aligned}$$

  - The diff of generalized strings has some non-terminal symbol.

$$\begin{aligned}&S/m_a(x_1, x_2) \to CD\ \ N_1/x_2\ \ N_2/x_1\\ &S/m_b(x_1, x_2) \to CDCDE\ \ N_3/x_1\\ &\qquad\qquad \downarrow chunk\\ &S/m_1(x_1) \to CD\ \ N_4/x_1\\ &N_4/m_2(x_1, x_2) \to N_1/x_2\ \ N_2/x_1\\ &N_4/m_3(x_1) \to CDE\ \ N_3/x_1\end{aligned}$$

- Merge

  We suppose the conditions on merge as follows.

$$\begin{aligned}\left({m_i}^{\mathcal{I}} \supseteq {m_j}^{\mathcal{I}}\right) &\vee \left({m_j}^{\mathcal{I}} \supseteq {m_i}^{\mathcal{I}}\right)\\ {m_1}^{\mathcal{I}} &= {m_a}^{\mathcal{I}} \cup {m_b}^{\mathcal{I}}\end{aligned}$$

- Merge two rules

$$S/m_2\left(x_1, x_2\right) \rightarrow N_1/x_1 \;\; E\flat FA \;\; N_2/x_2$$
$$N_1/m_a \rightarrow CD$$
$$N_1/m_3 \rightarrow AB\flat$$
$$N_2/m_b \rightarrow CD$$
$$\downarrow merge$$
$$S/m_2\left(x_1, x_2\right) \rightarrow N_3/x_1 \;\; E\flat FA \;\; N_3/x_2$$
$$N_3/m_1 \rightarrow CD$$
$$N_3/m_3 \rightarrow AB\flat$$

- For start rule and not start rule

$$S/m_a\left(x_1, x_2\right) \rightarrow N_1/x_1 \;\; E\flat FA \;\; N_2/x_2$$
$$N_1/m_b\left(x_1, x_2\right) \rightarrow N_1/x_1 \;\; E\flat FA \;\; N_2/x_2$$
$$\downarrow merge$$
$$S/m_1\left(x_1, x_2\right) \rightarrow N_1/x_1 \;\; E\flat FA \;\; N_2/x_2$$

- Replace

We suppose the conditions on replace as follows.

$$m_i{}^{\mathcal{I}} \supseteq m_j{}^{\mathcal{I}}$$
$$m_1{}^{\mathcal{I}} = m_a{}^{\mathcal{I}} - m_b{}^{\mathcal{I}}$$

- Replace a part of string in left side to the other rule.

$$S/m_a \rightarrow F^{\#}GCEDAF^{\#}$$
$$N_1/m_b \rightarrow ED$$
$$\downarrow replace$$
$$S/m_1\left(x_1\right) \rightarrow F^{\#}GC \;\; N_1/x_1 \;\; AF^{\#}$$
$$N_1/m_b \rightarrow ED$$

- For two compositional rules

$$S/m_a\left(x_1\right) \rightarrow F^{\#}GC \;\; N_1/x_1 \;\; AF^{\#}$$
$$N_2/m_b\left(x_2\right) \rightarrow F^{\#}GC \;\; N_1/x_2$$
$$\downarrow replace$$
$$S/m_1\left(x_1\right) \rightarrow N_2/x_1 \;\; AF^{\#}$$
$$N_2/m_b\left(x_2\right) \rightarrow F^{\#}GC \;\; N_2/x_2$$

E. Generation of Symbol String

In our model, each meaning possesses a unique label. Therefore, when given a label, the composisiton of a sequence of symbols is straightforward; the composite label structure directly represents the tree, being different from other generative models like deep learning or probabilistic CFG. In the following example, we can easily understand that m_1 is the top node and has two branches of m_2 and $m_4(m_1(\cdots))$.

- Target label: $m_1\left(m_2, m_4\left(m_1\left(m_3, m_5\left(m_2\right)\right)\right)\right)$

Agent's Knowledge

$$S/m_1\left(x_1, x_2\right) \rightarrow N_1/x_1 \;\; N_2/x_2$$
$$N_1/m_2 \rightarrow F^{\#}GD$$
$$N_1/m_3 \rightarrow CDC$$
$$N_2/m_4\left(x_3\right) \rightarrow F^{\#}GC \;\; S/x_3$$
$$N_2/m_5\left(x_4\right) \rightarrow CD \;\; N_1/x_4$$

$$S/m_1\left(x_1, x_2\right) \rightarrow N_1/x_1 \;\; N_2/x_2$$
$$S/m_1\left(m_2, x_2\right) \rightarrow F^{\#}GD \;\; N_2/x_2$$
$$S/m_1\left(m_2, m_4\left(x_3\right)\right) \rightarrow F^{\#}GDF^{\#}GC \;\; S/x_3$$
$$S/m_1\left(m_2, m_4\left(m_1\left(x_4, x_5\right)\right)\right)$$
$$\rightarrow F^{\#}GDF^{\#}GC \;\; N_1/x_4 \;\; N_2/x_5$$
$$S/m_1\left(m_2, m_4\left(m_1\left(m_3, x_5\right)\right)\right)$$
$$\rightarrow F^{\#}GDF^{\#}GCCDC \;\; N_2/x_5$$
$$S/m_1\left(m_2, m_4\left(m_1\left(m_3, m_5\left(x_6\right)\right)\right)\right)$$
$$\rightarrow F^{\#}GDF^{\#}GCCDCCD \;\; N_1/x_6$$
$$S/m_1\left(m_2, m_4\left(m_1\left(m_3, m_5\left(m_2\right)\right)\right)\right)$$
$$\rightarrow F^{\#}GDF^{\#}GCCDCCDF^{\#}GD$$

IV. Evaluation Method for Syntactic Structure

In natural language processing, when we evaluate the validity of the process, we compose a parse tree of sentences and investigate if the tree reflects the adequate structure or not because we believe the tree reflects the meaning of a sentence. In the similar way, we consider the adequacy of the tree structure of music piece even though music does not own explicit meaning like natural language.

However, in general, we may not be able to enumerate all the probable trees for a given music piece since our LCFG has a property of recursion and the combination of applicable generation rules can be infinite. In order to avoid this combinatorial explosion of computational complexity of $O\left(2^n\right)$, we restrict the number of probable trees to one for each music piece.

Based on the newly acquired generative rules, we can compose the exactly same music since every generation process can be preserved in those rules. In addition, the new rules compose a new piece of music. If we can record the history of the generation process, we can detect how the new rules were applied differently from the original one; that is, the original piece is decomposed by the rules and the each part is recomposed in a different way. Then, we give an analysis system of the logs of composition.

A rule of agent's learned knowledge has a condition that the order of variables in left side is the same with one in right side. m_a and m_b are atomic meanings before apply a learning method, m_1, m_2 and m_3 are used in new rules. We can denote a rewriting rule as follows.

$$args: x, y, z, x', z'$$
$$m_a, m_b, m_1, m_2, m_3 \in M$$

- Chunk

$$m_a\left(x, y, z\right) \Rightarrow m_1\left(x, m_2\left(y\right), z\right)$$
$$m_b\left(x', y, z'\right) \Rightarrow m_1\left(x', m_3\left(y\right), z'\right)$$

- Merge

$$m_a \Rightarrow m_1$$
$$m_b \Rightarrow m_1$$

- Replace

$$m_a\left(x, y, z\right) \Rightarrow m_1\left(x, m_2\left(y\right), z\right)$$

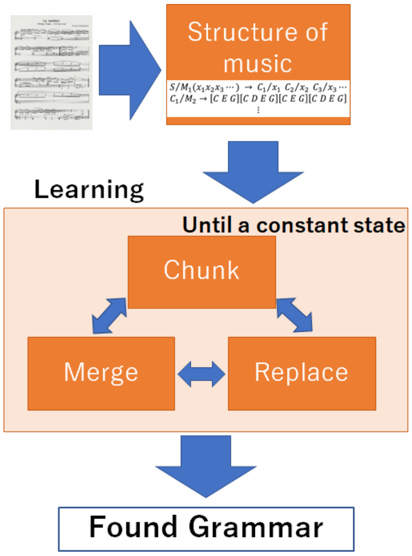

Fig. 3. An Image of the Simulation

$$C_1/M_1 \rightarrow [C\ E\ F\ G]\ [C\ D\ E\ G]\ [C\ E\ F\ G]\ [C\ D\ E\ G]$$

Fig. 4. Translation of Measure to Symbol String

V. Experimental Result

We experimented to find grammar in the 25 études of Burgmüller (Fig. 3). At first, we divide each piece beat by beat, and we have grouped a set of notes in one beat. Then, we translate a piece of music to a sequence of sets of simultaneously sounding notes. Then, an agent tries to find generative rules in these pieces.

In this translation process we have embedded the notion of *measure*. We have explicitly defined that the sequence of a certain number of beats composes a measure (Fig. 4), giving a single atomic label on each as terminal symbols. Therefore, each rule which represents a measure has four terminal symbols if the piece is four beat music.

A whole piece of music is represented by a rule with the start non-terminal symbol. The rule has a string of non-terminal symbols in right hand side.

A. Detection of Music Knowledge

Two salient results are as follows.

1) progression to stable chord: In Fig. 6, we can find the solution to a harmonic chord from a set of dissonant notes. Fig. 5 shows the first bar of *Le Candeur* of Burgmüller. The second beat includes a passing note of D, but is resolved to C in the third beat. We can find a corresponding generation rule that represents a progression from the dissonant to consonant chord.

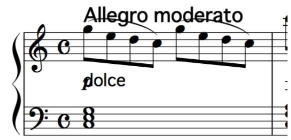

Fig. 5. Part of a Piece: Recursion to a stable chord in the 25 études of Burgmüller

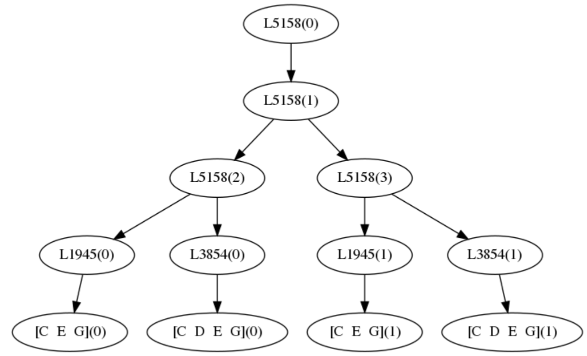

Fig. 6. Parse Tree: Recursion to a stable chord in the 25 études of Burgmüller

2) phrase detection: In early classicist music, the music phrase and motif is articulated by 4 bars, 8 bars, and 16 bars. *La Candeur* also suggests us the phrasing by 8 bars. Our rule has found that the articulation after the 17-th bar after the first repetition consisting of eight-bars, though we have not explicitly put the punctuation here (Fig. 7, 8).

B. Composition of Music Score

After learning the generative rules, an agent can compose a piece of music. Fig. 9 is created by a learned agent.

VI. Conclusion

In this research, we have modified the ILM by Kirby, and have applied it to grammar detection in music. We have chopped off the 25 études of Burgmüller by beat, and have found the statistically plausible connections between them. Although many of those nodes in tree, that are the categories

Fig. 7. Part of Piece: Phrase detection in the 25 études of Burgmüller

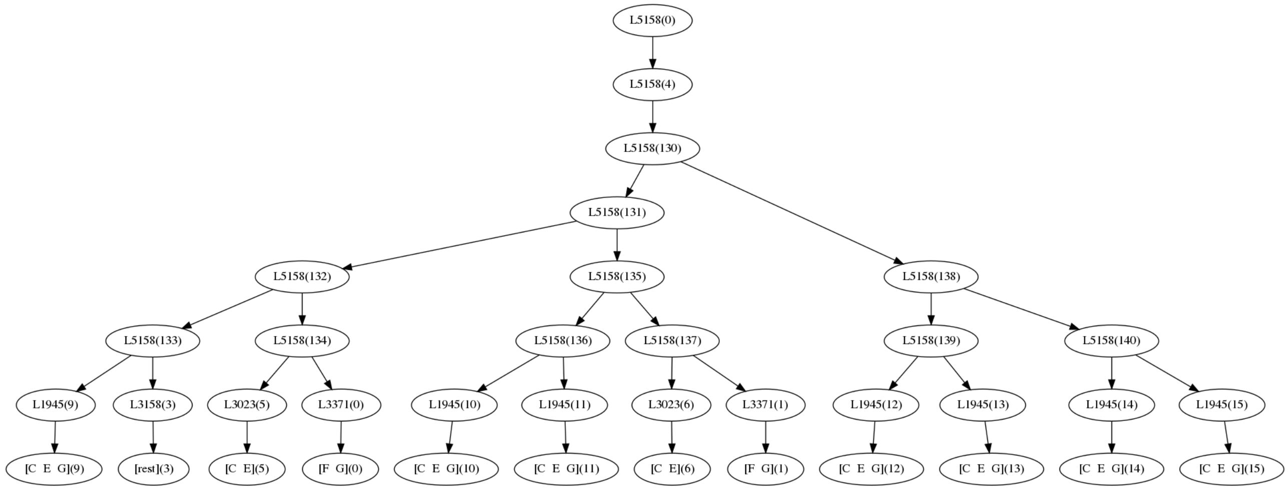

Fig. 8. Parse Tree: Phrase detection in the 25 études of Burgmüller

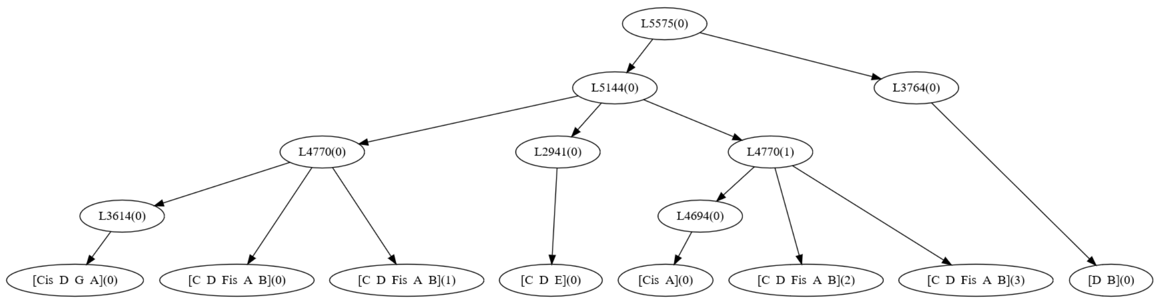

Fig. 9. Parce Tree: Creation of a piece

of the syntax, seem redundant and futile, we still found that some of them reflected the probable progression, which well matched with our human intuition, *e.g.*, the progression from a set of dissonant notes including suspension or appoggiatura to a chordal set.

This experiment has several virtues compared with other grammar-based formalism for music. One is that we do not need to provide a dictionary beforehand. The other is that we can exclude the human-biased intuition, which had hindered the definition of creativity.

Our experiment has stayed still in only one generation. We need to continue to the *iterated* model, to observe the further development of the grammar.

References

[1] David Cope, Experiments in Musical Intelligence, A-R Editions (1996).

[2] David Cope, Computer Models of Music Creativity, the MIT Press (2005)

[3] M. Hamanaka, K. Hirata and S. Tojo. Implementing A Generative Theory of Tonal Music, *Journal of New Music Research*, 35(4), pp.249–277 (2007)

[4] Masatoshi Hamanaka and Keiji Hirata, Implementing Methods for Analysing Music Based on Lerdahl and Jackendoffs Generative Theory of Tonal Music, David Meredith (Ed), Computational Music Analysis, Chapter 9, pp.221–249, Springer (2016).

[5] Fred Lerdahl and Ray Jackendoff, A Generative Theory of Tonal Music, The MIT Press (1983).

[6] Keiji Hirata, Satoshi Tojo, and Masatoshi Hamanaka, An Algebraic Approach to Time-Span Reduction, David Meredith (Ed), Computational Music Analysis, Chapter 10, pp.251–270, Springer (2016).

[7] M. Granroth-Wilding and M. Steedman. Statistical Parsing for Harmonic Analysis of Jazz Chord Sequences, *Proc. of ICMC2012* (2012)

[8] M. Granroth-Wilding and M. Steedman. A Robust Parser-Interpreter for Jazz Chord Sequences, *Journal of New Music Research*, vol.43, pp.354–374 (2014)

[9] P. N. Johnson-Laird, O. E. Kang, and Y, C. Leong. On Musical Dissonance, *Journal of Music Perception*, vol.30, no.1,pp. 19-35 (2012)

[10] Kirby, Simon. Learning Bottlenecks and the Evolution of Recursive Syntax, in T. Briscoe (Ed.), Linguistic Evolution through Language Acquisition, Cambridge University Press, (2002)

[11] S. Tojo, Y. Oka, and M. Nishida. Analysis of Chord Progression by HPSG, in *AIA'06 Proceedings of the 24th IASTED international conference on Artificial intelligence and applications*, ACTA Press (2006)

[12] N. L. Wallin, B. Merker, and S. Brown. (eds.). *The Origins of Music*. The MIT Press (2000)

[13] T. Winograd. Linguistics and the computer analysis of tonal harmony, *Journal of Music Theory*, 12(1), pp.2–49 (1968)

[14] P. Ball; The Music Instinct, Vintage Books (2010)