The music box operad: Random generation of musical phrases from patterns

Samuele Giraudo
Université du Québec à Montréal, LACIM,
Pavillon Président-Kennedy, 201 Avenue du Président-Kennedy, Montréal, H2X 3Y7, Canada.
giraudo.samuele@uqam.ca

Abstract

We introduce the notion of multi-patterns, a combinatorial abstraction of polyphonic musical phrases. The interest of this approach in encoding musical phrases lies in the fact that it becomes possible to compose multi-patterns in order to produce new ones. This composition is parameterized by a monoid structure on the scale degrees. This embeds the set of the musical phrases into an algebraic framework since the set of the multi-patterns is endowed with the structure of an operad. Operads are algebraic structures offering a formalization and an abstraction of the notion of operators and their compositions. Seeing musical phrases as operators allows us to perform computations on phrases and admits applications in generative music. Indeed, given a set of initial multi-patterns, we propose various algorithms to randomly generate a new and longer phrase emulating the style suggested by the inputted multi-patterns. The designed algorithms use types of grammars working with operads and colored operads, known as bud generating systems.

1 Introduction

Generative music is a subfield of computational musicology in which the focus lies on the automatic creation of musical material (Collins, 2008; Diaz-Jerez, 2000; Fernández & Vico, 2013). This creation is based on algorithms accepting inputs to influence the result obtained. Such algorithms could have a randomized behavior in the sense that two executions of the algorithm with the same inputs produce different results. One of the challenges to overcome in designing such an algorithm consists in building a procedure offering a balance between freedom of what is created and adequacy with the specified inputs. As a matter of fact, the expression *generative music* is usually understood as a class of methods producing music by rules, with a possibly deterministic behavior (Loy & Abbott, 1985).

Several very different approach exist, each with their own advantages and areas of applications. For instance, some algorithms use Markov chains (Ames, 1989; Shapiro & Huber, 2021), others genetic algorithms (Gartland-Jones & Copley, 2003; Matić, 2010), still others neural networks (Briot et al., 2020), or even —and directly related to the present work— formal grammars (Bel, 1989; Eibensteiner, 2018; Garcia Salas et al., 2011; Holtzman, 1981; Hudak & Quick, 2018; Lerdhal & Jackendoff, 1996; McCormack, 1996; Quick & Hudak, 2013; Roads, 1979; Steedman, 1984) (see (Hopcroft et al., 2006) for a general presentation of formal grammars). In the case of the Markov chain approach, the input is a corpus of musical pieces. The algorithm essentially builds a Markov chain for the probabilities of a note to be played according to some previous ones, and then, uses it to build randomly a new musical piece which inherits in some sense from the inputted ones. The genetic algorithm approach is based on a fitness function, a crucial object to evaluate the quality of the generated phrases and make sure that, over the iterations, the output converges to a satisfying one. Neural networks produce potentially very interesting results provided that an adequate learning phase is carried out. Nevertheless, the inherent disadvantage of this method is that it is a black box whose internal mechanisms are difficult to understand. It is therefore in most of the cases hard for the human composer using the computer as a source of inspiration to slightly modify the generation parameters in order to direct the algorithm to generate phrases of a certain kind. In the case of the formal grammar approach, the input data is a formal grammar specifying the language of the possible results. The algorithm builds a musical piece by performing a random generation of a word of the language specified by the grammar by selecting, using various strategies, a rule of the grammar at random at each step.

The way in which such algorithms represent and manipulate musical data is crucial. Indeed, the data structures used to represent musical phrases orient the nature of the operations we can define on them. Operations that produce new phrases from old ones are an important ingredient in specifying algorithms similar to those that manipulate formal grammars to randomly generate music. A possible way for this purpose consists in giving at input some musical phrases and the algorithm creates a new one by blending them through operations. Therefore, the willingness to endow the infinite set of the musical phrases with operations in order to obtain suitable algebraic structures is a promising approach. Such interactions between music and algebra form a fruitful field of investigation (Andreatta, 2018; Hudak, 2004; Jedrzejewski, 2019; Tymoczko, 2011; Warrack, 1945). For instance, in (Hudak, 2004) (see also (Hudak & Quick, 2018)), musical phrases are encoded as treelike structures wherein phrases are associated by concatenation or by superposition. In (Morris, 1987), an abstraction of phrases is proposed wherein pitch classes are used instead of absolute notes. In (Eibensteiner, 2018), a phrase is a sequence of tuples containing data including a pitch, a duration, and a volume. Moreover, (Cambouropoulos et al., 2001) reviews a wide range of other data structures.

In the present work, we propose to use tools coming from combinatorics and algebraic combinatorics to represent musical phrases and operations on them, in order to introduce generative music algorithms close to the family of those based on formal grammars. More precisely, we present the music box model, a model to represent polyphonic phrases by combinatorial objects called multi-patterns. Here, a “multi-pattern” (and its monophonic version, a “pattern”) should not be understood as, in the usual way, a fragment of a piece admitting multiples occurrences of it. This terminology of multi-pattern and pattern should be understood as follows: a multi-pattern is rather an abstraction of a non-fixed musical phrase (that is, the same pattern can give rise to different musical phrases when it is seen under different interpretations) which can be combined with others to form bigger phrases. The music box model is designed to handle musical phrases written for the family of lamellophones. Instruments of this family, like the kalimba, are capable of striking notes whose duration is determined by how long they resonate, rather than being deliberately chosen. This is precisely the musical framework of application of the tools designed in this work.

The infinite set of the multi-patterns admits the structure of an operad. This specific property is the starting point of this work. Operads originate from algebraic topology (Boardman & Vogt, 1973; May, 1972) and are used nowadays also in algebraic combinatorics and in computer science (Giraudo, 2018; Loday & Vallette, 2012; Méndez, 2015). Roughly speaking, in these structures, the elements are operations with several inputs and one output, and the composition law mimics the usual composition of such operators. Since the set of multi-patterns forms an operad, one can regard each multi-pattern as an operation. The fallout of this is that each multi-pattern is, at the same time, a musical phrase (under some interpretation) and an operation acting on musical phrases. In this way, our music box model and its associated operad provide an algebraic and combinatorial framework to perform computations on musical phrases. To the best of our knowledge, the present work is the first one to build a bridge between operad theory and generative music.

The music box model and the constructed operads on multi-patterns admit direct applications to design random generation algorithms since, as introduced by the author in (Giraudo, 2019), given an operad there exist algorithms to generate some of its elements. These algorithms are based upon bud generating systems, which are general formal grammars based on colored operads (Yau, 2016). In the present work, we introduce three different variations of these algorithms to produce new musical phrases from old ones. Each algorithm works more or less as follows. It takes as input a finite set of multi-patterns and an integer value to influence the size of the output. It chooses iteratively a multi-pattern from the initial collection in order to alter the current one by performing a composition using the operad structure. As we shall explain, the initial multi-patterns are endowed with colors in order to forbid certain compositions and, in this way, avoid specific musical intervals, particular rhythmic motives, or impose a certain general structure. These colors play a similar role to the one of nonterminal symbols of formal grammars. Even though this is feasible, these generation algorithms are not intended to write complete musical pieces; rather, they aim to generate a new, longer, and similar pattern from short primitive multi-patterns, potentially bringing new ideas to the human composer. These algorithms can be used to generate phrases satisfying rules similar to the ones of species counterpoint (see (Komosinski & Szachewicz, 2015) for the description of an automated composition method in this context), or to create a piece by emulating and mixing the style of some input phrases.

This contribution presents some new interactions between music and mathematics, and a new framework for automatic composition. Our method is not supposed to be better than the other existing ones and has it is own advantages and disadvantages. As main features, the model allows us to compute over musical phrases since, as mentioned before, each multi-pattern is in fact an operation which admits as inputs other multi-patterns and produces a new multi-pattern. Such operations are parameterizable by specifying how the scale degrees are transformed by providing a monoid structure on the scale degrees (in other words, the operad of the multi-patterns is parameterized by a monoid). A feature of the model is that we can express some musical transformations (like transpositions, inversion, retrograde, *etc.*) either as multi-patterns seen like operations or through operad morphisms. In all the cases, these operations are defined by the same language as the one used by the music box model. Besides, according to what is intended, it is possible to specify different monoids in order to change the behavior of the operations and also of the generated patterns. The other main features and limitations of the model are discussed in more details further.

This text is organized as follows. Section 2 is devoted to present the music box model. Degree patterns, rhythm patterns, patterns, and multi-patterns are defined. We explain how a multi-pattern describes a musical phrase given an interpretation. We discuss also the strengths and the weaknesses of this model. In Section 3, we begin by presenting a brief overview of operad theory and we build step by step the music box operad. For this, we introduce first an operad on sequences of scale degrees (depending as explained above on a monoid structure in order to encode how to compute on degrees), an operad on rhythm patterns, and then an operad on monophonic patterns to finish with the operad of multi-patterns. After providing some background on colored operads and bud generating systems, three random generation algorithms for multi-patterns are introduced in Section 4. Section 5 provides some concrete applications of the previous algorithms. We first describe an implementation (Giraudo, 2023) of the music box model (ready to use by the reader) and the related algorithms. Then, we construct four particular bud generating systems, each with a specific objective. The paper ends by presenting in Section 6 an evaluation of the generation algorithms. This evaluation is based on a questionnaire aimed at experts, intended to understand how pieces generated by our methods are received.

This paper is an extended version of (Giraudo, 2020) containing some new results (like the complete study of the introduced operads) and their proofs. Moreover, we describe in this present version a more general model than the one presented in the previous work: now, the scale degrees of the multi-patterns are elements of a monoid.

General notations and conventions For any integers i and j, $[i, j]$ denotes the set $\{i, i+1, \dots, j\}$. For any integer i, $[i]$ denotes the set $[1, i]$. A word is a finite sequence of elements. For any set A, A^* is the set of the words on A. Given a word u, $\ell(u)$ is the length of u, and for any $i \in [\ell(u)]$, $u(i)$ is the i-th letter of u. For any $i \leq j \in [\ell(u)]$, $u(i, j)$ is the word $u(i)u(i+1)\dots u(j)$. If a is a letter and n is a nonnegative integer, a^n is the word consisting of n occurrences of a. In particular, a^0 is the empty word ϵ. If u and u' are two words, their concatenation is the word denoted by $u \centerdot u'$ or simply by uu' when the context is clear.

2 The music box model

The purpose of this section is to introduce multi-patterns, the main combinatorial objects used in this work to propose abstractions of musical phrases. This encoding of musical phrases forms the music box model.

2.1 Patterns and multi-patterns

We introduce here degree patterns, rhythm patterns, patterns, and finally multi-patterns which are the components of music box model, described in the next section.

2.1.1 Degree patterns

A *scale degree* (or, for short, a *degree*) is any element of $\mathbb{Z}$. Negative degrees are denoted by putting a bar above their absolute value. For instance, the degree -3 is denoted by $\bar{3}$. A *degree pattern* is a word $\mathbf{d}$ of degrees. The *arity* of $\mathbf{d}$, also denoted by $|\mathbf{d}|$, is the length of $\mathbf{d}$ as a word. For instance, $\mathbf{d} := 0\bar{1}210$ is a degree pattern of arity 5.

2.1.2 Rhythm patterns

A *rhythm pattern* $\mathbf{r}$ is a word on the alphabet $\{\square, \blacksquare\}$. The symbol $\square$ is a *rest* and the symbol $\blacksquare$ is a *beat*. The *length* $\ell(\mathbf{r})$ of $\mathbf{r}$ is the length of $\mathbf{r}$ as a word and the *arity* $|\mathbf{r}|$ of $\mathbf{r}$ is its number of occurrences of beats. The *duration sequence* of $\mathbf{r}$ is the unique word σ of nonnegative integers and of length $|\mathbf{r}| + 1$ such that

$$\mathbf{r} = \square^{\sigma(1)} \; \blacksquare \; \square^{\sigma(2)} \; \dots \; \blacksquare \; \square^{\sigma(|\mathbf{r}|+1)}. \tag{1}$$

For instance, $\mathbf{r} := \square\,\square\,\blacksquare\,\square\,\blacksquare\,\blacksquare\,\square$ is a rhythm pattern of length 7, arity 3, and of duration sequence 2101.

2.1.3 Patterns

A *pattern* is a pair $\mathbf{p} := (\mathbf{d}, \mathbf{r})$ such that $|\mathbf{d}| = |\mathbf{r}|$. The *arity* $|\mathbf{p}|$ of $\mathbf{p}$ is the common arity of both $\mathbf{d}$ and $\mathbf{r}$, and the *length* $\ell(\mathbf{p})$ of $\mathbf{p}$ is the length $\ell(\mathbf{r})$ of $\mathbf{r}$. For instance, $\mathbf{p} := (1\bar{1}2, \blacksquare\,\blacksquare\,\square\,\square\,\square\,\blacksquare\,\square)$ is a pattern of arity 3 and of length 7.

In order to handle concise notations, we shall write any pattern $(\mathbf{d}, \mathbf{r})$ as a word $\mathbf{p}$ on the alphabet $\{\square\} \cup \mathbb{Z}$ where the subword of $\mathbf{p}$ obtained by removing all occurrences of $\square$ is the degree pattern $\mathbf{d}$, and the word obtained by replacing in $\mathbf{p}$ each integer by $\blacksquare$ is the rhythm pattern $\mathbf{r}$. For instance,

$$1\square\square\bar{2}\square 12 \tag{2}$$

is the concise notation for the pattern

$$(1\bar{2}12, \blacksquare\,\square\,\square\,\blacksquare\,\square\,\blacksquare\,\blacksquare). \tag{3}$$

By following this convention, in the sequel we shall see and treat any pattern $\mathbf{p}$ as a word on the alphabet $\{\square\} \cup \mathbb{Z}$. Therefore, for any $i \in [\ell(\mathbf{p})]$, $\mathbf{p}(i)$ is the i-th letter of $\mathbf{p}$ which is either $\square$ or a degree. Remark that the length of $\mathbf{p}$ is the length of $\mathbf{p}$ as a word and that the arity of $\mathbf{p}$ is the number of letters of $\mathbb{Z}$ it has.

2.1.4 Multi-patterns

A *multi-pattern* $\mathbf{m}$ is a word of length $m \geq 1$ of patterns such that all patterns $\mathbf{m}(i)$, $i \in [m]$, have the same arity and the same length. The *arity* $|\mathbf{m}|$ of $\mathbf{m}$ is the common arity of all the $\mathbf{m}(i)$, the *length* $\ell(\mathbf{m})$ of $\mathbf{m}$ is the common length of all the $\mathbf{m}(i)$, and the *multiplicity* $\mathfrak{m}(\mathbf{m})$ of $\mathbf{m}$ is m. The *stacking* of two multi-patterns $\mathbf{m}_1$ and $\mathbf{m}_2$ having the same arity and the same length (but possibly different multiplicities), is the multi-pattern $\mathbf{m}_1 \,\mathbf{.}\, \mathbf{m}_2$. This multi-pattern has the same arity and length as the ones of $\mathbf{m}_1$ and $\mathbf{m}_2$, and its multiplicity is $\mathfrak{m}(\mathbf{m}_1) + \mathfrak{m}(\mathbf{m}_2)$.

A multi-pattern $\mathbf{m}$ is represented by a matrix of dimension $\mathfrak{m}(\mathbf{m}) \times \ell(\mathbf{m})$, where the i-th row contains the pattern $\mathbf{m}(i)$ for any $i \in [\mathfrak{m}(\mathbf{m})]$. For any $i \in [\mathfrak{m}(\mathbf{m})]$ and any $j \in [\ell(\mathbf{m})]$, $\mathbf{m}(i)(j)$ is hence the letter $\mathbf{p}(j)$ where $\mathbf{p}$ is the pattern $\mathbf{m}(i)$. For instance,

$$\mathbf{m} := \begin{vmatrix} 0 & \square & 1 & \square & 1 \\ \square & \bar{2} & \bar{3} & \square & 0 \end{vmatrix} \tag{4}$$

is a multi-pattern of multiplicity 2, having 3 as arity, and 5 as length. It satisfies $\mathbf{m}(1)(1) = 0$, $\mathbf{m}(1)(4) = \square$, and $\mathbf{m}(2)(3) = \bar{3}$.

In the sequel, for obvious reasons of practicality, we shall consider that a pattern is a multi-pattern of multiplicity 1 and conversely.

2.2 Interpretations

Let us now explain how to convert multi-patterns into musical phrases. We introduce also the music box model and discuss some of its drawbacks and advantages.

2.2.1 From multi-patterns to musical phrases

Let $\mathcal{N}$ be any set of musical notes. A *degree interpretation* is a map $\rho : \mathbb{Z} \to \mathcal{N}$. A *rhythm interpretation* is a duration δ. An *interpretation* is a pair (ρ, δ) where ρ is a degree interpretation and δ is a rhythm interpretation. The *(ρ, δ)-interpretation* of a pattern $\mathbf{p}$ is the musical phrase composed of the sequence of rests and notes obtained by translating each rest of $\mathbf{p}$ as a rest and each degree d of $\mathbf{p}$ as the note $\rho(d)$, both lasting the duration prescribed by δ. The *(ρ, δ)-interpretation* of a multi-pattern $\mathbf{m}$ is the musical phrase obtained by superimposing the (ρ, δ)-interpretations of each pattern $\mathbf{m}(i)$, $i \in [\mathfrak{m}(\mathbf{m})]$.

2.2.2 Standard interpretation

The *standard interpretation* is the interpretation (ρ, δ) where ρ sends the degree 2 to the “middle C” and the other degrees accordingly with the diatonic scale (for instance, the degree 0 is interpreted as the note A located three semitones below the middle C, the degree 9 is interpreted as the note C located one octave above the middle C, and the degree $\bar{1}$ is interpreted as the note G located five semitones below the middle C) and δ is the duration of $\frac{1}{2}$ seconds.

For instance, the multi-pattern

$$\mathbf{m} := \begin{vmatrix} 0 & \square & 1 & 2 & 1 & \square & 1 & \square & 2 & 3 & 2 & \square \\ \square & 2 & 3 & 4 & 3 & \square & 3 & 4 & 5 & 4 & \square & \square \\ \bar{7} & \square & \bar{6} & \bar{5} & \bar{6} & \square & \bar{6} & \square & \bar{5} & \bar{4} & \bar{5} & \square \end{vmatrix} \tag{5}$$

is interpreted through the standard interpretation as the musical phrase

Unless otherwise stated, all the next multi-patterns of this paper are interpreted through the standard interpretation.

2.2.3 Music box model

Due to the fact that multi-patterns evoke paper tapes of a programmable music box, we call the model just described the *music box model* to represent musical phrases by multi-patterns within the context of interpretations. Interpretations of this model are musical phrases that may be particularly suitable to be played by the family of lamellophones (including the mbira and the kalimba) or keyboard percussive instruments (including the marimba and the xylophone).

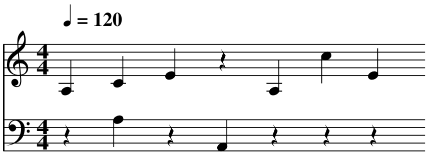
2.2.4 Limitations of the Music box model

Despite its simplicity, this model suffers from at least the following three main apparent limitations:

1. all patterns of a multi-pattern must have the same length;
2. all patterns of a multi-pattern must have the same arity;
3. the interpretation of a multi-pattern uses only rests and notes all having the same duration.

The first limitation, which requires that all patterns within a multi-pattern have the same length, ensures that all patterns of a multi-pattern have the same total duration. This limitation can be lifted by adding final rests to the shorter patterns within a multi-pattern so that they have virtually the same length as the longest.

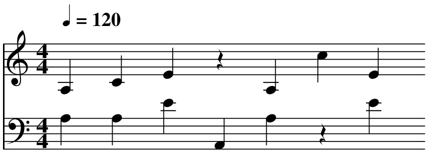
The second limitation, which requires that all patterns within a multi-pattern have the same arity, is a particularity of our model and comes from algebraic reasons which will be clarified later in the text. Here this limitation can be lifted by replacing, in a pattern with a smaller arity, some of its rests by degrees of any other pattern of the multi-pattern. For instance, consider the phrase



As it stands, no multi-pattern interprets into this phrase because the first voice is composed of six quarter notes while the second of only two. Nevertheless, as we have just explained, the multi-pattern

$$\begin{vmatrix} 0 & 2 & 4 & \square & 0 & 9 & 4 \\ 0 & 0 & 4 & \bar{7} & 0 & \square & 4 \end{vmatrix} \tag{6}$$

interprets as



and this phrase sounds like the previous one.

The last presented limitation, which restricts the interpretation of multi-patterns to rests and notes of the same duration, can be lifted through different techniques. One approach is to make each note of the interpretation of a multi-pattern continue to sound until the next note of the same pattern begins. This means for instance that if a degree is followed by three rests, this degree and the three rests are interpreted as a note of duration $1 + 3$ times the duration prescribed by δ, where δ is a rhythm interpretation. A more sophisticated solution involves using a map $\tau : \mathbb{N} \setminus \{0\} \to \mathbb{N}^2$ that assigns any $\alpha \in \mathbb{N} \setminus \{0\}$ to a pair (α_1, α_2) where $\alpha = \alpha_1 + \alpha_2$. This approach interprets any degree followed by $\alpha - 1$ rests as a note lasting α_1 times the duration prescribed by δ and then a rest lasting α_2 times the duration prescribed by δ, where $\tau(\alpha) = (\alpha_1, \alpha_2)$. For instance, by considering such a map τ satisfying $\tau(1) = (1, 0)$, $\tau(2) = (1, 1)$, $\tau(3) = (2, 1)$, and $\tau(4) = (4, 0)$, the multi-pattern

$$\begin{vmatrix} 2 & \square & 1 & \square & \square & 0 & 0 & \square \\ 4 & \square & \square & \square & 2 & 2 & \square & 7 \end{vmatrix} \tag{7}$$

is interpreted through the standard interpretation as the musical phrase

2.2.5 Strengths of the music box model

Let us now discuss some of the main strengths of the music box model. Multi-patterns have the following features:

1. the translation of multi-patterns to musical phrases through interpretations is almost transparent and is flexible;
2. more than just representing musical phrases, multi-patterns are in fact operations on musical phrases.

First point is obvious: with a little training, and with an interpretation in mind, it is easy to imagine how a multi-pattern sounds. Moreover, this representation is flexible in the sense that a multi-pattern does not specify a fixed sequence of rests and notes but rather a scheme (whence the appellation of "pattern" and "multi-pattern"). For instance, the pattern $|0\ 2\ 4\ 2\ 4|$ interprets as a minor arpeggio in the standard interpretation but interprets also as a major arpeggio in any major scale, among other possibilities.

The second point will be clarified in the sequel. It basically says that multi-patterns are not limited to simply representing musical phrases since they are in fact operations on musical phrases. This feature is perhaps for us the most compelling aspect of the model as it allows for the construction of larger multi-patterns by using a natural notion of composition. As we shall see, this composition is also highly flexible, as it can be parameterized according to the specific needs and desired musical outcomes.

3 Operad structures

The purpose of this section is to introduce an operad structure on multi-patterns, called music box operad. The main interest of endowing the set of multi-patterns with the structure of an operad is that this leads to an algebraic framework to perform parameterizable computations on musical phrases.

3.1 A primer on operads

We begin by setting here the elementary notions of operad theory used in the sequel. Most of them come from (Giraudo, 2018).

3.1.1 Graded sets

A *graded set* is a set $\mathcal{O}$ decomposing as a disjoint union

$$\mathcal{O} := \bigsqcup_{n \in \mathbb{N}} \mathcal{O}(n), \tag{8}$$

where the $\mathcal{O}(n)$, $n \in \mathbb{N}$, are sets. For any $x \in \mathcal{O}$, there is by definition a unique $n \in \mathbb{N}$ such that $x \in \mathcal{O}(n)$. This integer n is the *arity* of x and is denoted by $|x|$. Let $\mathcal{O}'$ be a second graded set. A map $\phi : \mathcal{O} \to \mathcal{O}'$ is a *graded set morphism* if for any $x \in \mathcal{O}$, $\phi(x) \in \mathcal{O}'(|x|)$. The identity graded set morphism is denoted by I. Besides, $\mathcal{O}'$ is a *graded subset* of $\mathcal{O}$ if $\mathcal{O}'(n) \subseteq \mathcal{O}(n)$ for any $n \in \mathbb{N}$.

The *Hadamard product* of two graded sets $\mathcal{O}$ and $\mathcal{O}'$ is the graded set $\mathcal{O} \boxtimes \mathcal{O}'$ defined, for any $n \in \mathbb{N}$, by $(\mathcal{O} \boxtimes \mathcal{O}')(n) := \mathcal{O}(n) \times \mathcal{O}'(n)$. Observe that if $\mathcal{O}''$ is another graded set, the graded sets $(\mathcal{O} \boxtimes \mathcal{O}') \boxtimes \mathcal{O}''$ and $\mathcal{O} \boxtimes (\mathcal{O}' \boxtimes \mathcal{O}'')$ are isomorphic, so that $\boxtimes$ is an associative operation. By a slight abuse of notation, for any other graded sets $\mathcal{O}_1$ and $\mathcal{O}'_1$, and any graded set morphisms $\phi : \mathcal{O} \to \mathcal{O}_1$ and $\phi' : \mathcal{O}' \to \mathcal{O}'_1$, we denote by $\phi \boxtimes \phi'$ the map from $\mathcal{O} \boxtimes \mathcal{O}'$ to $\mathcal{O}_1 \boxtimes \mathcal{O}'_1$ defined, for any $(x, x') \in \mathcal{O} \boxtimes \mathcal{O}'$, by $(\phi \boxtimes \phi')((x, x')) := (\phi(x), \phi'(x'))$. This map is a graded set morphism.

3.1.2 Nonsymmetric operads

A *nonsymmetric operad*, or an *operad* for short, is a triple $(\mathcal{O}, \circ_i, \mathbf{1})$ such that $\mathcal{O}$ is a graded set, $\circ_i$ is a map

$$\circ_i : \mathcal{O}(n) \times \mathcal{O}(m) \to \mathcal{O}(n + m - 1), \qquad 1 \leq i \leq n, \tag{9}$$

called *partial composition* map, and $\mathbf{1}$ is a distinguished element of $\mathcal{O}(1)$, called *unit*. This data has to satisfy, for any $x, y, z \in \mathcal{O}$, the three relations

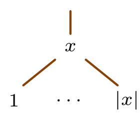
$$(x \circ_i y) \circ_{i+j-1} z = x \circ_i (y \circ_j z), \qquad 1 \leq i \leq |x|, \ \ 1 \leq j \leq |y|, \tag{10a}$$

$$(x \circ_i y) \circ_{j+|y|-1} z = (x \circ_j z) \circ_i y, \qquad 1 \leq i < j \leq |x|, \tag{10b}$$

$$\mathbf{1} \circ_1 x = x = x \circ_i \mathbf{1}, \qquad 1 \leq i \leq |x|. \tag{10c}$$

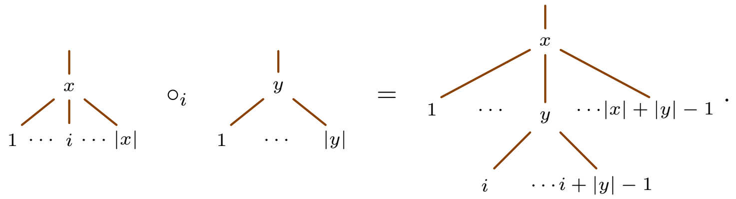
3.1.3 Abstract operators and intuition

From an intuitive point of view, an operad is an algebraic structure wherein each element x can be seen as an operator having $|x|$ inputs and one output. Such an operator is depicted as



(11)

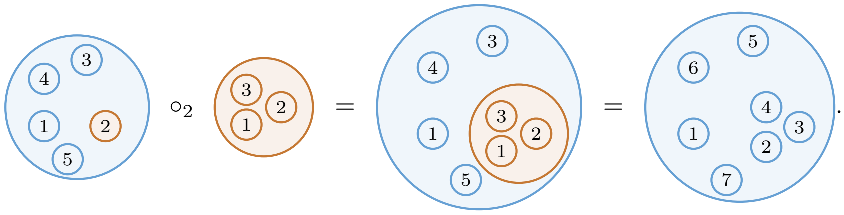
where the inputs are at the bottom and numbered from 1 to $|x|$, and the output is at the top. Given two operations x and y of $\mathcal{O}$, the partial composition $x \circ_i y$ is a new operator obtained by composing y onto the i-th input of x. Pictorially, this partial composition expresses as



(12)

Relations (10a), (10b), and (10c) become clear when they are interpreted into this context of abstract operators and rooted trees (also called syntax trees).

In a complementary manner, an interesting way to see the elements of an operad $\mathcal{O}$ consists in regarding any $x \in \mathcal{O}(n)$, $n \in \mathbb{N}$, as a combinatorial object with n substitute sectors labeled from 1 to n. The partial composition $x \circ_i y$ consists in replacing the substitute sector of x having i as label by y and by shifting the labels accordingly. For instance, here is a schematic composition of an element of arity 3 at the 2-nd position inside an element of arity 5:



In the context of this work, the combinatorial objects on which an operad structure will be defined are the multi-patterns. The substitute sectors of a multi-pattern $\mathbf{m}$ are its degrees, from the first labeled by 1 to the last labeled by n, where n is the arity of $\mathbf{m}$. Of course, this operad will be presented in detail in the following.

The operad axioms can be understood on multi-patterns in the following way (even if at this stage, the operad structure on these objects has not yet been defined). Relation (10a) says that, given three multi-patterns $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$, composing $\mathbf{m}_2$ into $\mathbf{m}_1$ and then, $\mathbf{m}_3$ into the area occupied by $\mathbf{m}_2$ of the obtained multi-pattern is the same as composing $\mathbf{m}_3$ into $\mathbf{m}_2$ and then, the obtained multi-pattern into $\mathbf{m}_1$. Relation (10b) says that, given three multi-patterns $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$, composing $\mathbf{m}_2$ into $\mathbf{m}_1$ and then $\mathbf{m}_3$ to the right of the area occupied by $\mathbf{m}_2$ of the obtained multi-pattern is the same as composing $\mathbf{m}_3$ into $\mathbf{m}_1$ and then, $\mathbf{m}_3$ to the left of the area occupied by $\mathbf{m}_2$ of the obtained multi-pattern. Finally, Relation (10c) says simply that there exists a multi-pattern playing the role of a unit element. These relations are the right ones to be in position to see multi-patterns as operations and to compose them as such.

3.1.4 Elementary definitions

Let $(\mathcal{O}, \circ_i, \mathbf{1})$ be an operad. The *full composition* map of $\mathcal{O}$ is the map

$$\circ : \mathcal{O}(n) \times \mathcal{O}(m_1) \times \cdots \times \mathcal{O}(m_n) \to \mathcal{O}(m_1 + \cdots + m_n) \tag{13}$$

defined for any $x \in \mathcal{O}(n)$ and $y_1, \ldots, y_n \in \mathcal{O}$ by

$$x \circ [y_1, \ldots, y_n] := (\ldots((x \circ_n y_n) \circ_{n-1} y_{n-1}) \ldots) \circ_1 y_1. \tag{14}$$

Intuitively, $x \circ [y_1, \ldots, y_n]$ is obtained by grafting simultaneously the outputs of all the y_i onto the i-th inputs of x. Additionally, the *homogeneous composition* map of $\mathcal{O}$ is the map

$$\odot : \mathcal{O}(n) \times \mathcal{O}(m) \to \mathcal{O}(nm), \tag{15}$$

defined, for any $x \in \mathcal{O}(n)$ and $y \in \mathcal{O}(m)$ by

$$x \odot y := x \circ \underbrace{[y, \ldots, y]}_{n \text{ elements}}. \tag{16}$$

Let $(\mathcal{O}', \circ'_i, \mathbf{1}')$ be a second operad. A graded set morphism $\phi : \mathcal{O} \to \mathcal{O}'$ is an *operad morphism* if $\phi(\mathbf{1}) = \mathbf{1}'$ and for any $x, y \in \mathcal{O}$ and $i \in [|x|]$,

$$\phi(x \circ_i y) = \phi(x) \circ'_i \phi(y). \tag{17}$$

If instead (17) holds by replacing the second occurrence of i by $|x| - i + 1$, then ϕ is an *operad antimorphism*. We say that $\mathcal{O}'$ is a *suboperad* of $\mathcal{O}$ if $\mathcal{O}'$ is a graded subset of $\mathcal{O}$, $\mathbf{1} = \mathbf{1}'$, and for any $x, y \in \mathcal{O}'$ and $i \in [|x|]$, $x \circ_i y = x \circ'_i y$. For any subset $\mathfrak{G}$ of $\mathcal{O}$, the *operad generated* by $\mathfrak{G}$ is the smallest suboperad $\mathcal{O}^{\mathfrak{G}}$ of $\mathcal{O}$ containing $\mathfrak{G}$. When $\mathcal{O}^{\mathfrak{G}} = \mathcal{O}$ and $\mathfrak{G}$ is minimal with respect to the inclusion among the subsets of $\mathfrak{G}$ satisfying this property, $\mathfrak{G}$ is a *minimal generating set* of $\mathcal{O}$ and its elements are *generators* of $\mathcal{O}$.

The *Hadamard product* of $\mathcal{O}$ and $\mathcal{O}'$ is the graded set $\mathcal{O} \boxtimes \mathcal{O}'$ endowed with the partial composition map $\circ''_i$ defined, for any $(x, x'), (y, y') \in \mathcal{O} \boxtimes \mathcal{O}'$ and $i \in [|(x, x')|]$, by

$$(x, x') \circ''_i (y, y') := (x \circ_i y, x' \circ'_i y'), \tag{18}$$

and having $(\mathbf{1}, \mathbf{1}')$ as unit. This graded set $\mathcal{O} \boxtimes \mathcal{O}'$ is an operad. Moreover, for any other operads $\mathcal{O}_1$ and $\mathcal{O}'_1$, and any operads morphisms (resp. antimorphisms) $\phi : \mathcal{O} \to \mathcal{O}_1$ and $\phi' : \mathcal{O}' \to \mathcal{O}'_1$, the graded set morphism $\phi \boxtimes \phi'$ is also an operad morphism (resp. antimorphism) from $\mathcal{O} \boxtimes \mathcal{O}'$ to $\mathcal{O}_1 \boxtimes \mathcal{O}'_1$.

3.2 Music box operads

We build operads on multi-patterns step by step by introducing operads on degree patterns and an operad on rhythm patterns. The operads of patterns are constructed as the Hadamard product of the two previous ones. Finally, the operads of multi-patterns are suboperads of iterated Hadamard products of operads of patterns with themselves. These operads (except the operad of rhythm patterns) depend on a monoid structure on $\mathbb{Z}$ in order to encode how to compute on degrees.

3.2.1 Operads of degree patterns

The construction of an operad on degree patterns is based upon the *construction* $\mathbf{T}$, a construction from monoids to operads introduced in (Giraudo, 2015) which we recall now. Given a monoid $(\mathcal{M}, \star, \mathrm{e})$, where $\star$ is an associative operation admitting e as unit, let $\mathbf{T}(\mathcal{M})$ be the graded set of the words on $\mathcal{M}$, where the arity of a word is its length. Let $\bar{\star} : \mathcal{M} \times \mathbf{T}(\mathcal{M}) \to \mathbf{T}(\mathcal{M})$ be the map defined for any $a \in \mathcal{M}$ and $u \in \mathbf{T}(\mathcal{M})$ by

$$a \,\bar{\star}\, u := a \star u(1) \centerdot \cdots \centerdot a \star u(\ell(u)). \tag{19}$$

The graded set $\mathbf{T}(\mathcal{M})$ is endowed with a partial composition map $\circ_i$, defined for any $u, u' \in \mathbf{T}(\mathcal{M})$ and $i \in [|u|]$, by

$$u \circ_i u' := u(1, i-1) \centerdot\ u(i) \,\bar{\star}\, u' \ \centerdot u(i+1, \ell(u)). \tag{20}$$

It is shown in (Giraudo, 2015) that $(\mathbf{T}(\mathcal{M}), \circ_i, \mathbf{1})$, where $\mathbf{1}$ is the element e of $\mathbf{T}(\mathcal{M})(1)$, is an operad. To be perfectly precise, in (Giraudo, 2015), the presented construction is so that in $\mathbf{T}(\mathcal{M})$ there is no element of arity 0. We consider here the same construction with the difference that $\mathbf{T}(\mathcal{M})(0)$ is the singleton $\{\epsilon\}$. This is justified in the present work because we wish to manipulate degree patterns of arity 0.

Besides, by extending some results of (Giraudo, 2015), it is possible to show that $\mathbf{T}(\mathcal{M})$ admits as a minimal generating set the set

$$\{\epsilon, \mathrm{ee}\} \cup \mathfrak{G}_{\mathcal{M}} \tag{21}$$

where $\mathfrak{G}_{\mathcal{M}}$ is a minimal generating set of $\mathcal{M}$ as a monoid. Besides, if $\mathcal{M}'$ is another monoid and $\phi : \mathcal{M} \to \mathcal{M}'$ is a monoid morphism, let $\mathbf{T}(\phi) : \mathbf{T}(\mathcal{M}) \to \mathbf{T}(\mathcal{M}')$ be the map defined, for any $u \in \mathbf{T}(\mathcal{M})$, by

$$\mathbf{T}(\phi)(u) := \phi(u(1)) \centerdot \cdots \centerdot \phi(u(\ell(u))). \tag{22}$$

It is also shown in (Giraudo, 2015) that $\mathbf{T}(\phi)$ is an operad morphism preserving injections and surjections.

Let $\mathrm{mir} : \mathbf{T}(\mathcal{M}) \to \mathbf{T}(\mathcal{M})$ be the map defined, for any $u \in \mathbf{T}(\mathcal{M})$, by

$$\mathrm{mir}(u) := u(\ell(u)) \centerdot \cdots \centerdot u(1). \tag{23}$$

The word $\mathrm{mir}(u)$ is the *mirror* of u.

Proposition 3.1. *For any monoid $\mathcal{M}$, the map* mir *is an operad anti-automorphism of* $\mathbf{T}(\mathcal{M})$.

Proof. This is a straightforward verification based upon the fact that for any $u \in \mathbf{T}(\mathcal{M})$, the i-th letter of $\operatorname{mir}(u)$ is $u(\ell(u) - i + 1)$. □

A *degree monoid* is any monoid $(D, \star, \mathrm{e})$ such that $D \subseteq \mathbb{Z}$. By construction, for any $n \in \mathbb{N}$, each $\mathbf{d} \in \mathbf{T}(D)(n)$ is a sequence of integers of length n, and thus $\mathbf{d}$ is a degree pattern of arity n. For this reason, $\mathbf{T}(D)$ is an operad having as underlying graded set the graded set of the degree patterns having elements of D as degrees. We denote by

$$\mathsf{DP}^D := \mathbf{T}(D) \tag{24}$$

this operad, called *D-degree pattern operad*.

Let us consider three important examples depending on different natural degree monoids.

1. By denoting by $\mathbb{Z}$ the additive monoid $(\mathbb{Z}, +, 0)$, $\mathsf{DP}^{\mathbb{Z}}$ contains all degree patterns. In $\mathsf{DP}^{\mathbb{Z}}$, we have
$$\bar{1}01\bar{3}2 \circ_4 211 = \bar{1}01(-3+2)(-3+1)(-3+1)2 = \bar{1}01\bar{1}\bar{2}\bar{2}2. \tag{25}$$
Moreover, since $\mathbb{Z}$ admits $\{-1, 1\}$ as a minimal generating set, the operad $\mathsf{DP}^{\mathbb{Z}}$ admits $\{\epsilon, \bar{1}, 1, 00\}$ as a minimal generating set.
2. By denoting, for any $k \geq 1$, by $\mathbb{C}_k$ the cyclic monoid $(\mathbb{Z}/k\mathbb{Z}, +, 0)$, $\mathsf{DP}^{\mathbb{C}_k}$ contains the degree patterns having degrees between 0 and $k-1$. In $\mathsf{DP}^{\mathbb{C}_3}$ we have
$$20101 \circ_3 2120 = 20020101. \tag{26}$$
Moreover, since $\mathbb{C}_k$ admits $\{1\}$ as a minimal generating set, the operad $\mathsf{DP}^{\mathbb{C}_k}$ admits $\{\epsilon, 1, 00\}$ as a minimal generating set.
3. By denoting, for any subset Z of $\mathbb{Z}$ having a lower bound z, by $\mathbb{M}_Z$ the monoid $(Z, \max, z)$, $\mathsf{DP}^{\mathbb{M}_Z}$ contains all degree patterns having degrees in Z. In $\mathsf{DP}^{\mathbb{M}_{[0,2]}}$ we have
$$20010 \circ_4 2120 = 20021210. \tag{27}$$
Moreover, since $\mathbb{M}_Z$ admits $Z \setminus \{z\}$ as a minimal generating set, the operad $\mathsf{DP}^{\mathbb{M}_Z}$ admits $\{\epsilon\} \cup (Z \setminus \{z\}) \cup \{00\}$ as a minimal generating set.

By Proposition 3.1, for any degree monoid D, the map $\operatorname{mir} : \mathsf{DP}^D \to \mathsf{DP}^D$ is an operad anti-automorphism. Moreover, we can consider on the operads $\mathsf{DP}^{\mathbb{Z}}$, $\mathsf{DP}^{\mathbb{C}_k}$, and $\mathsf{DP}^{\mathbb{M}_Z}$ the following other morphisms.

1. For any $\alpha \in \mathbb{Z}$, let $\operatorname{mul}_\alpha : \mathsf{DP}^{\mathbb{Z}} \to \mathsf{DP}^{\mathbb{Z}}$ be the map defined by $\operatorname{mul}_\alpha := \mathbf{T}(\phi)$ where ϕ is the monoid morphism satisfying $\phi(d) = \alpha d$ for any $d \in \mathbb{Z}$. For instance,
$$\operatorname{mul}_{-2}(1\bar{2}003) = \bar{2}400\bar{6}. \tag{28}$$
Since ϕ is a monoid morphism, $\operatorname{mul}_\alpha$ is an operad endomorphism. Moreover, when $\alpha \neq 0$, $\operatorname{mul}_\alpha$ is injective, and when $\alpha \in \{-1, 1\}$, $\operatorname{mul}_\alpha$ is bijective.
2. For any $k \geq 1$, let also $\operatorname{red}_k : \mathsf{DP}^{\mathbb{Z}} \to \mathsf{DP}^{\mathbb{C}_k}$ be the map defined by $\operatorname{red}_k := \mathbf{T}(\phi)$ where ϕ is the monoid morphism satisfying $\phi(d) = d \bmod k$ for any $d \in \mathbb{Z}$. For instance,
$$\operatorname{red}_3(1\bar{2}003) = 11000. \tag{29}$$
Since ϕ is a surjective monoid morphism, $\operatorname{red}_\alpha$ is a surjective operad morphism.
3. For any subsets Z and Z' of $\mathbb{Z}$ having respective lower bounds z and z', a map $\theta : Z \to Z'$ is a *rooted weakly increasing map* if $\theta(z) = z'$ and, for any $d, d' \in Z$, $d \leq d'$ implies $\theta(d) \leq \theta(d')$. If θ is such a map, let $\operatorname{incr}_\theta : \mathsf{DP}^{\mathbb{M}_Z} \to \mathsf{DP}^{\mathbb{M}'_Z}$ be the map defined by $\operatorname{incr}_\theta := \mathbf{T}(\theta)$. For instance, for $Z := [0, 3]$ and $Z' := [2, 5]$, and θ satisfying $\theta(d) = d + 2$ for any $d \in Z$,
$$\operatorname{incr}_\theta(12003) = 34225. \tag{30}$$
Since θ is a monoid morphism, $\operatorname{incr}_\theta$ is an operad morphism. This morphism is not necessarily injective nor surjective.

3.2.2 Operad of rhythm patterns

Let us introduce a new construction from monoids to operads, similar to the construction $\mathbf{T}$ recalled in Section 3.2.1. Given a monoid $(\mathcal{M}, \star, \mathrm{e})$, let $\mathbf{U}(\mathcal{M})$ be the graded set of the nonempty words on $\mathcal{M}$, where the arity of a word is its length minus 1. Let $\overleftarrow{\star} : \mathcal{M} \times \mathbf{U}(\mathcal{M}) \to \mathbf{U}(\mathcal{M})$ and $\overrightarrow{\star} : \mathbf{U}(\mathcal{M}) \times \mathcal{M} \to \mathbf{U}(\mathcal{M})$ be the maps defined for any $a \in \mathcal{M}$ and $u \in \mathbf{U}(\mathcal{M})$ by

$$a \overleftarrow{\star} u := a \star u(1) \centerdot u(2, \ell(u)) \tag{31}$$

and

$$u \overrightarrow{\star} a := u(1, \ell(u) - 1) \centerdot u(\ell(u)) \star a. \tag{32}$$

The graded set $\mathbf{U}(\mathcal{M})$ is endowed with a partial composition map $\circ_i$, defined for any $u, u' \in \mathbf{U}(\mathcal{M})$ and $i \in [|u|]$, by

$$u \circ_i u' := u(1, i-1) \centerdot u(i) \overleftarrow{\star} u' \overrightarrow{\star} u(i+1) \centerdot u(i+2, \ell(u)). \tag{33}$$

Note that the associativity of $\star$ ensures that (33) is well-defined in particular when $\ell(u') = 1$. Let us also denote by $\mathbf{1}$ the element ee of $\mathbf{U}(\mathcal{M})(1)$. Moreover, if $\mathcal{M}'$ is another monoid and $\phi : \mathcal{M} \to \mathcal{M}'$ is a monoid morphism, let $\mathbf{U}(\phi) : \mathbf{U}(\mathcal{M}) \to \mathbf{U}(\mathcal{M}')$ be the map defined, for any $u \in \mathbf{U}(\mathcal{M})$, by

$$\mathbf{U}(\phi)(u) := \phi(u(1)) \centerdot \cdots \centerdot \phi(u(\ell(u))). \tag{34}$$

Proposition 3.2. *For any monoid $\mathcal{M}$, $\mathbf{U}(\mathcal{M})$ is an operad. Moreover, for any monoids $\mathcal{M}$ and $\mathcal{M}'$, and any monoid morphism $\phi : \mathcal{M} \to \mathcal{M}'$, $\mathbf{U}(\phi)$ is an operad morphism.*

Proof. By using the fact that the product of $\mathcal{M}$ is associative and that $\mathcal{M}$ has a unit, it follows by a straightforward but technical verification that Relations (10a), (10b), and (10c) are satisfied. Finally, the fact that ϕ is a monoid morphism says that ϕ commutes with the product of $\mathcal{M}$. As a straightforward computation shows, this implies that $\mathbf{U}(\phi)$ is an operad morphism. $\square$

Let $\mathrm{mir} : \mathbf{U}(\mathcal{M}) \to \mathbf{U}(\mathcal{M})$ be the map defined, for any $u \in \mathbf{U}(\mathcal{M})$, by

$$\mathrm{mir}(u) := u(\ell(u)) \centerdot \cdots \centerdot u(1). \tag{35}$$

The word $\mathrm{mir}(u)$ is the *mirror* of u.

Proposition 3.3. *For any commutative monoid $\mathcal{M}$, the map* mir *is an operad anti-automorphism of* $\mathbf{U}(\mathcal{M})$.

Proof. This is a straightforward verification based upon the fact that for any $u \in \mathbf{U}(\mathcal{M})$, the i-th letter of $\mathrm{mir}(u)$ is $u(\ell(u) - i + 1)$. The commutativity of $\mathcal{M}$ is important here. $\square$

By Proposition 3.2,

$$\mathsf{RP} := \mathbf{U}(\mathbb{N}) \tag{36}$$

where $\mathbb{N}$ is the additive monoid $(\mathbb{N}, +, 0)$ is an operad. By construction, for any $n \in \mathbb{N}$, each $\sigma \in \mathsf{RP}(n)$ is a sequence of nonnegative integers of length $n+1$, and thus, σ is a duration sequence of a rhythm pattern of arity n (see Section 2.1.2). For this reason, RP can be seen as an operad having as underlying graded set the graded set of the rhythm patterns. This operad is the *rhythm pattern operad*. We have for instance,

$$00121 \circ_3 110 = 002121, \tag{37a}$$

$$110 \circ_1 12 = 230, \tag{37b}$$

$$211 \circ_1 2 = 51. \tag{37c}$$

Directly on rhythm patterns, the partial composition rephrases as follows. For any rhythm patterns $\mathbf{r}$ and $\mathbf{r}'$, and any integer $i \in [|\mathbf{r}|]$, $\mathbf{r} \circ_i \mathbf{r}'$ is obtained by replacing the i-th occurrence of $\blacksquare$ in $\mathbf{r}$ by $\mathbf{r}'$. For instance, on rhythm patterns, (37a), (37b), and (37c) translate respectively as

$$\blacksquare\blacksquare\square\blacksquare\square\square\blacksquare\square \circ_3 \square\blacksquare\square\blacksquare = \blacksquare\blacksquare\square\square\blacksquare\square\blacksquare\square\square\blacksquare\square, \tag{38a}$$

$$\square\blacksquare\square\blacksquare \circ_1 \square\blacksquare\square\square = \square\square\blacksquare\square\square\square\blacksquare, \tag{38b}$$

$$\square\square\blacksquare\square\blacksquare\square \circ_1 \square\square = \square\square\square\square\square\blacksquare\square. \tag{38c}$$

Proposition 3.4. *The operad* RP *admits* $\{\epsilon, \square, \blacksquare\blacksquare\}$ *as a minimal generating set.*

Proof. Let us denote by $\mathfrak{G}$ the candidate minimal generating set for RP described in the statement of the proposition. Any rhythm pattern $\mathbf{r} \in \mathsf{RP}$ such that $\ell(\mathbf{r}) \geq 2$ decomposes as

$$\mathbf{r} = (\underbrace{\blacksquare\blacksquare \circ_1 \cdots \circ_1 \blacksquare\blacksquare}_{\ell(\mathbf{r})-1 \text{ terms}}) \circ [\mathbf{r}(1), \dots, \mathbf{r}(\ell(\mathbf{r}))]. \tag{39}$$

Since for any $i \in [\ell(\mathbf{r})]$, $\mathbf{r}(i) \in \{\blacksquare, \square\}$ and $\blacksquare$ is the unit of RP and $\square$ belongs to $\mathfrak{G}$, $\mathbf{r}$ is an element of $\mathsf{RP}^{\mathfrak{G}}$. Additionally, all rhythm patterns of lengths 0 or 1 belong to $\mathsf{RP}^{\mathfrak{G}}$ because $\epsilon \in \mathfrak{G}$, $\square \in \mathfrak{G}$, and $\blacksquare$ is the unit of RP. This shows that $\mathfrak{G}$ is a generating set of RP. Finally, this generating set is minimal since no element of $\mathfrak{G}$ can be expressed by partial compositions of some different other ones. □

For the next lemma, recall that when $\mathbf{r}$ is a rhythm pattern, $\ell(\mathbf{r})$ is the length of $\mathbf{r}$ and this quantity is the number of occurrences of $\square$ plus the number of occurrences $\blacksquare$ in $\mathbf{r}$.

Lemma 3.1. *For any two rhythm patterns $\mathbf{r}$ and $\mathbf{r}'$, and $i \in [|\mathbf{r}|]$, in the operad* RP, $\ell(\mathbf{r} \circ_i \mathbf{r}') = \ell(\mathbf{r}) + \ell(\mathbf{r}') - 1$.

Proof. This is a direct consequence of the interpretation of the partial composition of RP in terms of rhythm patterns: the rhythm pattern $\mathbf{r} \circ_i \mathbf{r}'$ is obtained by replacing a $\blacksquare$ of $\mathbf{r}$ by $\mathbf{r}'$. □

By Proposition 3.3, since $\mathbb{N}$ is a commutative monoid, the map $\mathrm{mir} : \mathsf{RP} \to \mathsf{RP}$ is an operad anti-automorphism. On rhythm patterns, given $\mathbf{r} \in \mathsf{RP}$, $\mathrm{mir}(\mathbf{r})$ is the rhythm pattern obtained by reading $\mathbf{r}$ from right to left. For instance,

$$\mathrm{mir}(\square\blacksquare\blacksquare\square\blacksquare\blacksquare\blacksquare) = \blacksquare\blacksquare\blacksquare\square\blacksquare\blacksquare\square. \tag{40}$$

For any $\beta \in \mathbb{N}$, let $\mathrm{dil}_\beta : \mathsf{RP} \to \mathsf{RP}$ be the map defined by $\mathrm{dil}_\beta := \mathbf{U}(\phi)$ where ϕ is the monoid morphism satisfying $\phi(s) = \beta s$ for any $s \in \mathbb{N}$. Since ϕ is a monoid morphism of $\mathbb{N}$, by Proposition 3.3, dil_β is an operad endomorphism. When $\beta \neq 0$, dil_β is injective and dil_β is surjective if and only if $\beta = 1$. Interpreted on rhythm patterns, $\mathrm{dil}_\beta(\mathbf{r})$ is obtained by replacing each occurrence of $\square$ in $\mathbf{r}$ by $\square^\beta$. For instance,

$$\mathrm{dil}_2(\square\blacksquare\blacksquare\square\square) = \square\square\blacksquare\blacksquare\square\square\square\square, \tag{41a}$$

$$\mathrm{dil}_0(\square\blacksquare\blacksquare\square\square) = \blacksquare\blacksquare. \tag{41b}$$

3.2.3 Operads of patterns

For any degree monoid $(D, \star, \mathrm{e})$, let P^D be the operad defined as

$$\mathsf{P}^D := \mathsf{DP}^D \boxtimes \mathsf{RP}. \tag{42}$$

By construction, for any $n \in \mathbb{N}$, each $\mathbf{p} \in \mathsf{P}^D(n)$ is a pair $(\mathbf{d}, \mathbf{r})$ such that $\mathbf{d}$ is a degree pattern of arity n having elements of D as degrees and $\mathbf{r}$ is a rhythm pattern of arity n. For this reason, P^D is an operad having as underlying graded set the graded set of the patterns having elements of D as degrees. We call P^D the *D-pattern operad*. In $\mathsf{P}^{\mathbb{Z}}$, we have for instance

$$(\bar{2}31, \square\blacksquare\blacksquare\square\blacksquare) \circ_2 (0\bar{1}, \blacksquare\square\blacksquare) = (\bar{2}321, \square\blacksquare\blacksquare\square\blacksquare\square\blacksquare), \tag{43a}$$

$$(112, \blacksquare\square\blacksquare\square\blacksquare) \circ_1 (\bar{1}, \square\square\blacksquare) = (012, \square\square\blacksquare\square\blacksquare\square\blacksquare). \tag{43b}$$

Directly by using the concise notation for patterns described in Section 2.1.3, the partial composition rephrases as follows. For any patterns $\mathbf{p}$ and $\mathbf{p}'$, and any integer $i \in [|\mathbf{p}|]$, $\mathbf{p} \circ_i \mathbf{p}'$ is obtained by replacing the i-th degree d of $\mathbf{p}$ by $\mathbf{p}''$ where $\mathbf{p}''$ is the pattern obtained by replacing each degree d' of $\mathbf{p}'$ by $d \star d'$. For instance, by using the concise notation for patterns, (43a) and (43b) translate as

$$\square\bar{2}3\square 1 \circ_2 0\square\bar{1} = \square\bar{2}3\square 2\square 1, \tag{44a}$$

$$1\square 1\square 2 \circ_1 \square\square\bar{1} = \square\square 0\square 1\square 2. \tag{44b}$$

Proposition 3.5. *Let $(D, \star, \mathrm{e})$ be a degree monoid admitting $\mathfrak{G}_D$ as a minimal generating set. The operad P^D admits $\{\epsilon, \square, \mathrm{ee}\} \cup \mathfrak{G}_D$ as a minimal generating set.*

Proof. Let us denote by $\mathfrak{G}$ the candidate minimal generating set for P^D described in the statement of the proposition. Any pattern $\mathbf{p} \in \mathsf{P}^D$ such that $\ell(\mathbf{p}) \geq 2$ decomposes as

$$\mathbf{p} = \underbrace{(\mathrm{ee} \circ_1 \cdots \circ_1 \mathrm{ee})}_{\ell(\mathbf{p})-1 \text{ terms}} \circ [\mathbf{p}(1), \dots, \mathbf{p}(\ell(\mathbf{p}))]. \tag{45}$$

By definition of patterns, for any $i \in [n]$, $\mathbf{p}(i)$ is either $\square$ or an element of D. Therefore, since $\mathfrak{G}$ contains $\mathfrak{G}_D$, each $\mathbf{p}(i)$ which is an element of D can be expressed by composing some elements of $\mathfrak{G}_D$. Since moreover $\square$ belongs to $\mathfrak{G}$, this shows that $\mathbf{p}$ is an element of $\mathsf{P}^{D^{\mathfrak{G}}}$. Additionally, all patterns of length 0 or 1 belong to $\mathsf{P}^{D^{\mathfrak{G}}}$ because $\epsilon \in \mathfrak{G}$, $\square \in \mathfrak{G}$, and since $\mathfrak{G}_D$ is a generating set of D, each pattern $d \in D$ of length 1 belongs to $\mathsf{P}^{D^{\mathfrak{G}}}$. Therefore, $\mathfrak{G}$ is a generating set of P^D. Finally, this generating set is minimal since no element of $\mathfrak{G}$ can be expressed by partial compositions of some different other ones. $\square$

By Proposition 3.5,

1. the operad $\mathsf{P}^{\mathbb{Z}}$ admits $\{\epsilon, \square, \bar{1}, 1, 00\}$ as a minimal generating set;
2. for any $k \geq 1$, the operad $\mathsf{P}^{\mathbb{C}_k}$, admits $\{\epsilon, \square, 1, 00\}$ as a minimal generating set;
3. for any subset Z of $\mathbb{Z}$ having a lower bound z, $\mathsf{P}^{\mathbb{M}_Z}$ admits $\{\epsilon, \square, zz\} \cup (Z \setminus \{z\})$ as a minimal generating set.

Due to the construction of P^D as the Hadamard product of DP^D and RP, by Propositions 3.1 and 3.3, the map $\mathrm{mir} \boxtimes \mathrm{mir}$ is an operad anti-automorphism of P^D. Moreover, again due to the construction of P^D and the existence of the operad morphisms involving DP^D and RP presented in Sections 3.2.1 and 3.2.2, one can consider on the operads $\mathsf{P}^{\mathbb{Z}}$, $\mathsf{P}^{\mathbb{C}_k}$, and $\mathsf{P}^{\mathbb{M}_Z}$ the following morphisms.

1. For any $\alpha \in \mathbb{Z}$, the map $\mathrm{mul}_\alpha \boxtimes \mathrm{I}$ is an operad endomorphism of $\mathsf{P}^{\mathbb{Z}}$.
2. For any $k \geq 1$, the map $\mathrm{red}_k \boxtimes \mathrm{I}$ is an operad morphism from $\mathsf{P}^{\mathbb{Z}}$ to $\mathsf{P}^{\mathbb{C}_k}$.
3. For any subsets Z and Z' of $\mathbb{Z}$ having lower bounds and any rooted weakly increasing map $\theta : Z \to Z'$, the map $\mathrm{incr}_\theta \boxtimes \mathrm{I}$ is an operad morphism from $\mathsf{P}^{\mathbb{M}_Z}$ to $\mathsf{P}^{\mathbb{M}_{Z'}}$.
4. For any degree monoid D and $\beta \in \mathbb{N}$, the map $\mathrm{I} \boxtimes \mathrm{dil}_\beta$ is an operad endomorphism of P^D.

Let us describe some morphisms involving the operad P^D and the previous operads DP^D and RP. Let the map $\mathrm{dp}_D : \mathsf{DP}^D \to \mathsf{P}^D$ be defined, for any $\mathbf{d} \in \mathsf{DP}^D$ by

$$\mathrm{dp}_D(\mathbf{d}) := \left(\mathbf{d}, \blacksquare^{|\mathbf{d}|}\right). \tag{46}$$

This map is an injective operad morphism. Let also the map $\mathrm{rp}_D : \mathsf{RP} \to \mathsf{P}^D$ be defined, for any $\mathbf{r} \in \mathsf{RP}$ by

$$\mathrm{rp}_D(\mathbf{r}) := \left(\mathrm{e}^{|\mathbf{r}|}, \mathbf{r}\right), \tag{47}$$

where e is the unit of D. This map is an injective operad morphism.

3.2.4 Operads of multi-patterns

For any degree monoid D and any positive integer m, let $\mathsf{P}_m^{D\prime}$ be the operad defined as

$$\mathsf{P}_m^{D\prime} := \underbrace{\mathsf{P}^D \boxtimes \cdots \boxtimes \mathsf{P}^D}_{m \text{ terms}}. \tag{48}$$

By definition of the Hadamard product of graded sets, the elements of $\mathsf{P}_m^{D\prime}$ are m-tuples (or, equivalently, words of length m) of patterns. Let also P_m^D be the graded subset of the underlying graded set of $\mathsf{P}_m^{D\prime}$ restrained to the m-tuples of patterns having all an equal length.

Theorem 3.1. *For any degree monoid D and any positive integer m, P_m^D is an operad.*

Proof. We have to prove that the set P_m^D forms a suboperad of P'_m. Let us denote by e the unit of D. Since e is also the unit of the operad P^D, the m-tuple $(\mathrm{e}, \dots, \mathrm{e})$ belongs to $\mathsf{P}_m^D(1)$ and is the unit of P_m^D. Moreover, by Lemma 3.1, given two patterns $\mathbf{p}$ and $\mathbf{p}'$, for any $i \in [|\mathbf{p}|]$, the length of $\mathbf{p} \circ_i \mathbf{p}'$ is $\ell(\mathbf{p}) + \ell(\mathbf{p}') - 1$. This shows that all patterns of the multi-pattern resulting of a partial composition of two multi-patterns have the same length. This implies that the partial composition of two elements of P_m^D is also in P_m^D and hence, that P_m^D is an operad. $\square$

By construction, for any $n \in \mathbb{N}$, each $\mathbf{m} \in \mathsf{P}_m^D(n)$ is an m-tuple of patterns all having n as arity, all having elements of D as degrees, and all having the same length. For this reason, P_m^D is an operad having as underlying graded set the graded set of the multi-patterns of multiplicity m and having elements of D as degrees. By Theorem 3.1, P_m^D is an operad, and we call it the *D-music box operad* of multiplicity m. This construction of P_m^D explains why all the patterns of a multi-pattern must have the same arity. This is a consequence of the general definition of the Hadamard product of operads.

By using the matrix notation for multi-patterns described in Section 2.1.4, we have for instance respectively in $\mathsf{P}_2^{\mathbb{Z}}$ and in $\mathsf{P}_3^{\mathbb{Z}}$,

$$\begin{vmatrix} \square & \bar{2} & \bar{1} & \square & 0 \\ 0 & 1 & \square & \square & 1 \end{vmatrix} \circ_2 \begin{vmatrix} 1 & \square \\ \bar{3} & \square \end{vmatrix} = \begin{vmatrix} \square & \bar{2} & 0 & \square & \square & 0 \\ 0 & \bar{2} & \square & \square & \square & 1 \end{vmatrix}, \tag{49a}$$

$$\begin{vmatrix} 0 & \square & 0 \\ 2 & \square & 0 \\ 4 & 4 & \square \end{vmatrix} \circ_2 \begin{vmatrix} 7 & 7 \\ 0 & 0 \\ \bar{7} & \bar{7} \end{vmatrix} = \begin{vmatrix} 0 & \square & 7 & 7 \\ 2 & \square & 0 & 0 \\ 4 & \bar{3} & \bar{3} & \square \end{vmatrix}. \tag{49b}$$

Due to the construction of P_m^D as a suboperad of an operad obtained by an iterated Hadamard product of P^m and to the fact that, as explained in Section 3.2.3, $\mathrm{mir} \boxtimes \mathrm{mir}$ is an anti-automorphism of P, the map

$$\mathrm{mir} := \underbrace{(\mathrm{mir} \boxtimes \mathrm{mir}) \boxtimes \cdots \boxtimes (\mathrm{mir} \boxtimes \mathrm{mir})}_{m \text{ terms}} \tag{50}$$

is an operad anti-automorphism of P_m^D. Moreover, again due to the construction of P_m^D and the operad endomorphisms of P^D presented in Section 3.2.3, one can consider on the operads P_m^Z, $\mathsf{P}_m^{\mathbb{C}_k}$, and $\mathsf{P}_m^{\mathbb{M}_Z}$ the following morphisms.

1. For any $\alpha_1, \dots, \alpha_m \in \mathbb{Z}$, the map
$$\mathrm{mul}_{\alpha_1, \dots, \alpha_m} := (\mathrm{mul}_{\alpha_1} \boxtimes \mathrm{I}) \boxtimes \cdots \boxtimes (\mathrm{mul}_{\alpha_m} \boxtimes \mathrm{I}) \tag{51}$$
is an operad endomorphism of $\mathsf{P}_m^{\mathbb{Z}}$.
2. For any $k \geq 1$, the map
$$\mathrm{red}_k := \underbrace{(\mathrm{red}_k \boxtimes \mathrm{I}) \boxtimes \cdots \boxtimes (\mathrm{red}_k \boxtimes \mathrm{I})}_{m \text{ terms}} \tag{52}$$
is an operad morphism from $\mathsf{P}_m^{\mathbb{Z}}$ to $\mathsf{P}_m^{\mathbb{C}_k}$.
3. For any subsets Z and Z' of $\mathbb{Z}$ having lower bounds and any rooted weakly increasing maps $\theta_i : Z \to Z'$, $i \in [m]$, the map
$$\mathrm{incr}_{\theta_1, \dots, \theta_m} := (\mathrm{incr}_{\theta_1} \boxtimes \mathrm{I}) \boxtimes \cdots \boxtimes (\mathrm{incr}_{\theta_m} \boxtimes \mathrm{I}) \tag{53}$$
is an operad morphism from $\mathsf{P}_m^{\mathbb{M}_Z}$ to $\mathsf{P}_m^{\mathbb{M}_{Z'}}$.
4. For any degree monoid D and $\beta \in \mathbb{N}$,
$$\mathrm{dil}_\beta := \underbrace{(\mathrm{I} \boxtimes \mathrm{dil}_\beta) \boxtimes \cdots \boxtimes (\mathrm{I} \boxtimes \mathrm{dil}_\beta)}_{m \text{ terms}} \tag{54}$$
is an operad endomorphism of P_m^D.

Let also, for any $m \geq 1$, the map $\mathrm{cp}_m : \mathsf{P}^D \to \mathsf{P}_m^D$ be defined, for any $\mathbf{p} \in \mathsf{P}^D$, by

$$\mathrm{cp}_m(\mathbf{p}) := \underbrace{\mathbf{p} \centerdot \cdots \centerdot \mathbf{p}}_{m \text{ elements}}. \tag{55}$$

In other words, $\mathrm{cp}_m(\mathbf{p})$ is the multi-pattern obtained by stacking the pattern $\mathbf{p}$ with itself m times. This map is an injective operad morphism.

The full diagram involving the operads DP^D, RP, P^D, and P_m^D is

$$\begin{tikzcd}
 & \mathsf{P}_m^D & \\
\mathsf{DP}^D \arrow[r, tail, "\mathrm{dp}_D"] & \mathsf{P}^D = \mathsf{P}_1^D \arrow[u, tail, "\mathrm{cp}_m"'] & \mathsf{RP} \arrow[l, tail, "\mathrm{rp}_D"']
\end{tikzcd} \tag{56}$$

where D is a degree monoid, $m \geq 1$, and the arrows $\rightarrowtail$ are injective operad morphisms.

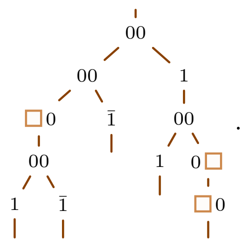
3.3 Operations on musical phrases

Thanks to the D-music box operads and more precisely, to the operad structures on multi-patterns, we can see any multi-pattern as an operator acting on multi-patterns. At the level of interpretations, a multi-pattern is hence an operator acting on musical phrases. We first describe an application of the D-music box operads to decompose multi-patterns in smaller pieces, then introduce a nomenclature for some special multi-patterns, and finish by explaining how to express some natural transformations on musical phrases in the language of the D-music box operads.

3.3.1 Decomposing multi-patterns

Instead of using the operads P_m^D to build multi-patterns, we can use these structures in the opposite way to decompose multi-patterns into smaller pieces. Indeed, an operad structure allows us to factorize its elements by means of treelike structures (see Section 3.1.3). More precisely, given a graded subset S of P_m^D and $\mathbf{m} \in \mathsf{P}_m^D$, an *$S$-decomposition* of $\mathbf{m}$ is a planar rooted tree $\mathfrak{t}$ having all internal nodes decorated by elements of S and such that the evaluation (as a syntax tree) of $\mathfrak{t}$ is $\mathbf{m}$.

For instance, the pattern $\mathbf{p} := \square 1 \bar{1} \bar{1} 2 \square 1 \square$ decomposes in $\mathsf{P}_1^{\mathbb{Z}} = \mathsf{P}^{\mathbb{Z}}$ as the tree

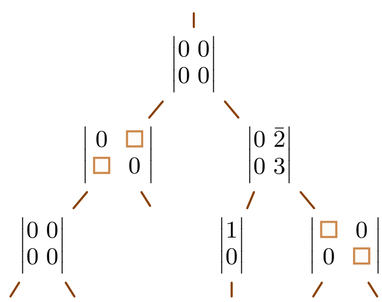


. (57)

This is a $\mathfrak{G}$-decomposition of $\mathbf{p}$ where $\mathfrak{G}$ is the on the minimal generating set of $\mathsf{P}^{\mathbb{Z}}$ described in Section 3.2.3. Besides, the multi-pattern

$$\mathbf{m} := \begin{vmatrix} 0 & 0 & \square & 1 & \square & \bar{2} \\ \square & 0 & 0 & 0 & 3 & \square \end{vmatrix} \tag{58}$$

admits in $\mathsf{P}_2^{\mathbb{Z}}$ the $\mathsf{P}_2^{\mathbb{Z}}$-decomposition



. (59)

At the musical level, this notion of decomposition is related to the parsing of musical phrases (see also (Lerdhal & Jackendoff, 1996)). Understanding the $\mathfrak{G}$-decompositions of a multi-pattern $\mathbf{m}$, where $\mathfrak{G}$ is a minimal generating set of P_m^D, brings information about the structure of $\mathbf{m}$, including its repetitions, its self-similarities, and its symmetries. It is possible to develop, within this framework of the bud music box model, a notion of complexity for multi-patterns by studying the different $\mathfrak{G}$-decompositions a multi-pattern has.

Nevertheless, the description of minimal generating sets for P_m^D when $m \geq 2$ seems to be much more intricate than for P_1^D and the previous operads (than for DP^D and for RP). For the time being, we do not have good and usable descriptions. Therefore, we do not have an effective way to propose $\mathfrak{G}$-decompositions of multi-patterns of multiplicity greater than 1 for the time being where $\mathfrak{G}$ is a minimal generating set of P_m^D when $m \geq 2$.

3.3.2 Particular multi-patterns

We define here some multi-patterns that play special roles.

1. **Chords.** A multi-pattern $\mathbf{m}$ is a *chord* if $\mathfrak{m}(\mathbf{m}) \geq 2$, $|\mathbf{m}| = 1$, and $\ell(\mathbf{m}) = 1$. For instance,
$$\begin{vmatrix} \bar{7} \\ 0 \\ 2 \\ 4 \end{vmatrix} \tag{60}$$
is a chord.
2. **Flat multi-patterns.** A multi-pattern $\mathbf{m}$ is *flat* if all degrees of its patterns $\mathbf{m}(i)$, $i \in [\mathfrak{m}(\mathbf{m})]$, are 0. For instance,
$$\begin{vmatrix} 0 & \square & \square & 0 \\ \square & 0 & 0 & \square \\ 0 & \square & 0 & \square \end{vmatrix} \tag{61}$$
is flat.
3. **Arpeggio shapes.** A multi-pattern $\mathbf{m}$ is an *arpeggio shape* if $\mathbf{m}$ is flat, $\mathfrak{m}(\mathbf{m}) \geq 2$, $|\mathbf{m}| = 1$, and, for any $i, i' \in [\mathfrak{m}(\mathbf{m})]$ and $j \in [\ell(\mathbf{m})]$, $\mathbf{m}(i)(j) \neq \square \neq \mathbf{m}(i')(j)$ implies $i = i'$. For instance,
$$\begin{vmatrix} \square & 0 & \square & \square \\ 0 & \square & \square & \square \\ \square & \square & 0 & \square \\ \square & \square & \square & 0 \end{vmatrix} \tag{62}$$
is an arpeggio shape.
4. **Arpeggios.** A multi-pattern $\mathbf{m}$ is an *arpeggio* if $\mathbf{m}$ writes as $\mathbf{m}' \odot \mathbf{m}''$ where $\mathbf{m}'$ is a chord of multiplicity m, $\mathbf{m}''$ is an arpeggio shape of multiplicity m, and $\odot$ is the homogeneous composition of the operad $\mathsf{P}_m^{\mathbb{Z}}$. For instance, since
$$\begin{vmatrix} \square & \bar{7} & \square & \square \\ 0 & \square & \square & \square \\ \square & \square & 2 & \square \\ \square & \square & \square & 4 \end{vmatrix} = \begin{vmatrix} \bar{7} \\ 0 \\ 2 \\ 4 \end{vmatrix} \odot \begin{vmatrix} \square & 0 & \square & \square \\ 0 & \square & \square & \square \\ \square & \square & 0 & \square \\ \square & \square & \square & 0 \end{vmatrix}, \tag{63}$$
the multi-pattern on left-hand side is an arpeggio.

3.3.3 Operations

Let us describe here some transformations on musical phrases by using the operad $\mathsf{P}_m^{\mathbb{Z}}$ and the related operad morphisms.

1. **Mimesis.** For any multi-patterns $\mathbf{m}$ and $\mathbf{m}'$ of the same multiplicity, $\mathbf{m} \odot \mathbf{m}'$ is the *mimesis* of $\mathbf{m}$ according to $\mathbf{m}'$. For instance
$$|\bar{1}\ \square\ \square\ 1| \odot |0\ 2\ 4\ \square| = |\bar{1}\ 1\ 3\ \square\ \square\ \square\ 1\ 3\ 5\ \square|. \tag{64}$$
Observe that since
$$|0\ 2\ 4\ \square| \odot |\bar{1}\ \square\ \square\ 1| = |\bar{1}\ \square\ \square\ 1\ 1\ \square\ \square\ 3\ 3\ \square\ \square\ 5\ \square|, \tag{65}$$

this operation is not commutative. Moreover, when $\mathbf{m}$ is a chord and $\mathbf{m}'$ is an arpeggio shape, the mimesis of $\mathbf{m}$ according to $\mathbf{m}'$ is by definition an arpeggio. For instance,

$$\left|\begin{matrix} \bar{7} \\ 0 \\ 2 \\ 4 \end{matrix}\right| \odot \left|\begin{matrix} \square & 0 & \square & \square \\ 0 & \square & \square & \square \\ \square & \square & 0 & \square \\ \square & \square & \square & 0 \end{matrix}\right| = \left|\begin{matrix} \square & \bar{7} & \square & \square \\ 0 & \square & \square & \square \\ \square & \square & 2 & \square \\ \square & \square & \square & 4 \end{matrix}\right|. \tag{66}$$

2. **Concatenation.** For any two multi-patterns $\mathbf{m}$ and $\mathbf{m}'$ of the same multiplicity m,

$$\mathrm{conc}(\mathbf{m}, \mathbf{m}') := \mathrm{cp}_m(00) \circ [\mathbf{m}, \mathbf{m}'] \tag{67}$$

is the concatenation of $\mathbf{m}$ and $\mathbf{m}'$. For instance,

$$\mathrm{conc}\left(\left|\begin{matrix} \bar{2} & \square & \square & 1 & \square \\ \square & 2 & \square & 3 & \square \end{matrix}\right|, \left|\begin{matrix} 0 & \square & 0 & 1 \\ \bar{1} & \bar{1} & \square & 3 \end{matrix}\right|\right) = \left|\begin{matrix} \bar{2} & \square & \square & 1 & \square & 0 & \square & 0 & 1 \\ \square & 2 & \square & 3 & \square & \bar{1} & \bar{1} & \square & 3 \end{matrix}\right|. \tag{68}$$

3. **Repetition.** For any multi-pattern $\mathbf{m}$ of multiplicity m and any $k \geq 1$,

$$\mathrm{rep}_k(\mathbf{m}) := \mathrm{cp}_m\left(0^k\right) \odot \mathbf{m} \tag{69}$$

is the k-fold repetition of $\mathbf{m}$. For instance,

$$\mathrm{rep}_3\left(\left|\begin{matrix} 0 & \square & 0 & 1 \\ \bar{1} & \bar{1} & \square & 3 \end{matrix}\right|\right) = \left|\begin{matrix} 0 & \square & 0 & 1 & 0 & \square & 0 & 1 & 0 & \square & 0 & 1 \\ \bar{1} & \bar{1} & \square & 3 & \bar{1} & \bar{1} & \square & 3 & \bar{1} & \bar{1} & \square & 3 \end{matrix}\right|. \tag{70}$$

4. **Transposition.** For any multi-pattern $\mathbf{m}$ of multiplicity m and any $d \in \mathbb{Z}$,

$$\mathrm{tran}_d(\mathbf{m}) := \mathrm{cp}_m(d) \odot \mathbf{m} \tag{71}$$

is the transposition of $\mathbf{m}$ by d degrees. For instance,

$$\mathrm{tran}_{-2}\left(\left|\begin{matrix} \bar{2} & \square & \square & 1 & \square \\ \square & 2 & \square & 3 & \square \end{matrix}\right|\right) = \left|\begin{matrix} \bar{4} & \square & \square & \bar{1} & \square \\ \square & 0 & \square & 1 & \square \end{matrix}\right|. \tag{72}$$

5. **Temporization.** For any multi-pattern $\mathbf{m}$ of multiplicity m and any $k \geq 0$,

$$\mathrm{temp}_k(\mathbf{m}) := \mathbf{m} \odot \mathrm{cp}_m\left(0\square^k\right) \tag{73}$$

is the temporization of $\mathbf{m}$ by k units of time. For instance,

$$\mathrm{temp}_2\left(\left|\begin{matrix} \bar{2} & \square & \square & 1 & \square \\ \square & 2 & \square & 3 & \square \end{matrix}\right|\right) = \left|\begin{matrix} \bar{2} & \square & \square & \square & \square & 1 & \square & \square & \square \\ \square & 2 & \square & \square & \square & 3 & \square & \square & \square \end{matrix}\right|. \tag{74}$$

6. **Inverse.** For any multi-pattern $\mathbf{m}$,

$$\mathrm{inv}(\mathbf{m}) := \mathrm{mul}_{-1,\dots,-1}(\mathbf{m}) \tag{75}$$

is the inverse of $\mathbf{m}$. For instance,

$$\mathrm{inv}\left(\left|\begin{matrix} \bar{2} & \square & \square & 1 & \square \\ \square & 2 & \square & 3 & \square \end{matrix}\right|\right) = \left|\begin{matrix} 2 & \square & \square & \bar{1} & \square \\ \square & \bar{2} & \square & \bar{3} & \square \end{matrix}\right|. \tag{76}$$

7. **Retrograde.** For any multi-pattern $\mathbf{m}$, $\mathrm{mir}(\mathbf{m})$ is the retrograde of $\mathbf{m}$. For instance,

$$\mathrm{mir}\left(\left|\begin{matrix} \bar{2} & \square & \square & 1 & \square \\ \square & 2 & \square & 3 & \square \end{matrix}\right|\right) = \left|\begin{matrix} \square & 1 & \square & \square & \bar{2} \\ \square & 3 & \square & 2 & \square \end{matrix}\right|. \tag{77}$$

8. **Retrograde inverse.** For any multi-pattern $\mathbf{m}$,

$$\mathrm{minv}(\mathbf{m}) := \mathrm{mir}(\mathrm{inv}(\mathbf{m})). \tag{78}$$

is the retrograde inverse of $\mathbf{m}$. Notice that $\mathrm{minv}(\mathbf{m})$ is also equal to $\mathrm{inv}(\mathrm{mir}(\mathbf{m}))$ since the two maps mir and inv commute. For instance,

$$\mathrm{minv}\left(\left|\begin{matrix} \bar{2} & \square & \square & 1 & \square \\ \square & 2 & \square & 3 & \square \end{matrix}\right|\right) = \left|\begin{matrix} \square & \bar{1} & \square & \square & 2 \\ \square & \bar{3} & \square & \bar{2} & \square \end{matrix}\right|. \tag{79}$$

9. **Harmonization.** For any pattern $\mathbf{p}$ and any chord multi-pattern $\mathbf{m}$ of multiplicity m,

$$\operatorname{har}(\mathbf{p}, \mathbf{m}) := \operatorname{cp}_m(\mathbf{p}) \odot \mathbf{m} \tag{80}$$

is the harmonization of $\mathbf{p}$ according to $\mathbf{m}$. For instance,

$$\operatorname{har}\left(\left| \bar{1} \ \square \ 1 \ \square \ 0 \ 2 \right|, \left| \begin{matrix} 0 \\ 2 \\ 4 \end{matrix} \right| \right) = \left| \begin{matrix} \bar{1} & \square & 1 & \square & 0 & 2 \\ 1 & \square & 3 & \square & 2 & 4 \\ 3 & \square & 5 & \square & 4 & 6 \end{matrix} \right|. \tag{81}$$

10. **Arpeggiation.** For any pattern $\mathbf{p}$ and any arpeggio multi-pattern $\mathbf{m}$ of multiplicity m,

$$\operatorname{arp}(\mathbf{p}, \mathbf{m}) := \operatorname{cp}_m(\mathbf{p}) \odot \mathbf{m} \tag{82}$$

is the arpeggiation of $\mathbf{p}$ according to $\mathbf{m}$. For instance,

$$\operatorname{arp}\left(\left| \bar{1} \ \square \ 1 \ \square \ 0 \ 2 \right|, \left| \begin{matrix} 1 & \square & \square \\ \square & \square & 4 \\ \square & \bar{2} & \square \end{matrix} \right| \right) = \left| \begin{matrix} 0 & \square & \square & \square & 2 & \square & \square & \square & 1 & \square & \square & 3 & \square & \square \\ \square & \square & 3 & \square & \square & \square & 5 & \square & \square & \square & 4 & \square & \square & 6 \\ \square & \bar{3} & \square & \square & \square & \bar{1} & \square & \square & \square & \bar{2} & \square & \square & 0 & \square \end{matrix} \right|. \tag{83}$$

Any operation preserving the lengths and the arities is *homogeneous*. The operations tran_d, $d \in \mathbb{Z}$, inv, mir, and minv are homogeneous.

We are now in position to be more specific in what we explained at the end of Section 2.2.5 about the fact that multi-patterns are operations on musical phrases. On the one hand, this feature is illustrated by the existence of the previous operations. On the other hand, observe that if $\mathbf{m}$ is a multi-pattern, it can be seen as an operation taking as input $|\mathbf{m}|$ multi-patterns $\mathbf{m}_1, \dots, \mathbf{m}_{|\mathbf{m}|}$ and outputting the multi-pattern $\mathbf{m} \circ \left[\mathbf{m}_1, \dots, \mathbf{m}_{|\mathbf{m}|}\right]$. For instance, the multi-pattern

$$\mathbf{m} := \left| \bar{1} \ 0 \ 2 \right| \tag{84}$$

can be seen as an operation of arity 3 such that, for any multi-patterns $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$ of multiplicity 1 as inputs, in $\mathsf{P}_1^{\mathbb{Z}}$, $\mathbf{m} \circ [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3]$ is the multi-pattern of multiplicity 1 obtained by concatenating a version of $\mathbf{m}_1$ transposed one degree down, with a copy of $\mathbf{m}_2$, and with a version of $\mathbf{m}_3$ transposed two degrees up. Similarly, the multi-pattern

$$\mathbf{m} := \left| \begin{matrix} 0 & \square \\ \square & 0 \end{matrix} \right| \tag{85}$$

can be seen an operation of arity 1 such that, for any multi-pattern $\mathbf{m}_1$ of multiplicity 2 as input, in $\mathsf{P}_2^{\mathbb{Z}}$, $\mathbf{m} \circ [\mathbf{m}_1]$ is the multi-pattern obtained by adding a final rest to the first pattern of $\mathbf{m}_1$ and by adding an initial rest to the second pattern of $\mathbf{m}_1$. The interpretation of this result is a musical phrase wherein the second voice has been shifted one unit of time. A lot of other operations modifying both the degrees and the rhythm can be constructed in this way.

3.3.4 Some benefits

Let us conclude this section by highlighting two additional significant benefits of the music box model, in addition to the ones presented in Section 2.2.5 for multi-patterns:

1. the model is homogeneous;
2. the composition of multi-patterns is flexible.

The first benefit concerns the homogeneity of the music box model. Operations on musical phrases are represented by using the same language as the musical phrases themselves, due to the fact that multi-patterns are operations (as mentioned in Section 3.3.2). This is in contrast to other systems where musical phrases and operations are defined using different languages, as seen in some other models (see for instance (Fernández & Vico, 2013)).

The second benefit, about the flexibility of the composition, is a consequence of the fact that the D-bud music box operad is parameterizable with different degree monoids D. By considering different degree monoids, it is possible to drastically change the behavior of the operations on multi-patterns and, as a result, the musical phrases generated by the model. While this section has mainly focused on D-music box operads with $D := (\mathbb{Z}, +, 0)$ as the degree monoid, other monoids may provide interesting operations on multi-patterns and musical phrases.

4 Generation and random generation

Now we exploit the music box operads introduced in the previous section to design three random generation algorithms devoted to generate new musical phrases from a finite set of multi-patterns. This relies on colored operads and bud generating systems, a sort of formal grammars introduced in (Giraudo, 2019).

4.1 Colored operads and bud operads

We provide here the elementary notions about colored operads (Yau, 2016) used in this work. We also explain how to build colored operads from an operad.

4.1.1 Colored operads

A *set of colors* is any nonempty finite set $\mathfrak{C} := \{\mathtt{b}_1, \dots, \mathtt{b}_r\}$ wherein elements are called *colors*. A *$\mathfrak{C}$-colored set* is a set $\mathcal{C}$ decomposing as a disjoint union

$$\mathcal{C} := \bigsqcup_{\substack{a \in \mathfrak{C} \\ u \in \mathfrak{C}^*}} \mathcal{C}(a, u), \tag{86}$$

where the $\mathcal{C}(a, u)$, $a \in \mathfrak{C}$, $u \in \mathfrak{C}^*$, are sets. For any $x \in \mathcal{C}$, there is by definition a unique pair $(a, u) \in \mathfrak{C} \times \mathfrak{C}^*$ such that $x \in \mathcal{C}(a, u)$. The *arity* $|x|$ of x is the length $\ell(u)$ of u as a word, the *output color* $\mathrm{out}(x)$ of x is a, and for any $i \in [|x|]$, the *i-th input color* $\mathrm{in}_i(x)$ of x is the i-th letter $u(i)$ of u. We also denote, for any $n \in \mathbb{N}$, by $\mathcal{C}(n)$ the set of the elements of $\mathcal{C}$ of arity n. Therefore, a colored set is in particular a graded set.

A *$\mathfrak{C}$-colored operad* is a triple $(\mathcal{C}, \circ_i, \mathbf{1})$ such that $\mathcal{C}$ is a $\mathfrak{C}$-colored set, $\circ_i$ is a map

$$\circ_i : \mathcal{C}(a, u) \times \mathcal{C}(u(i), v) \to \mathcal{C}(a, u \circ_i v), \qquad 1 \leq i \leq \ell(u), \tag{87}$$

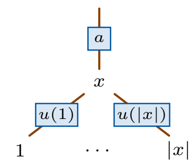
called a *partial composition* map, where $u \circ_i v$ is the word on $\mathfrak{C}$ obtained by replacing the i-th letter $u(i)$ of u by v, and $\mathbf{1}$ is a map

$$\mathbf{1} : \mathfrak{C} \to \mathcal{C}(a, a), \tag{88}$$

such that for any $a \in \mathfrak{C}$, $\mathbf{1}(a) \in \mathcal{C}(a, a)$, called a *colored unit* map. This data has to satisfy Relations (10a) and (10b) when their left and right members are both well-defined, and, for any $x \in \mathcal{C}$, the relation

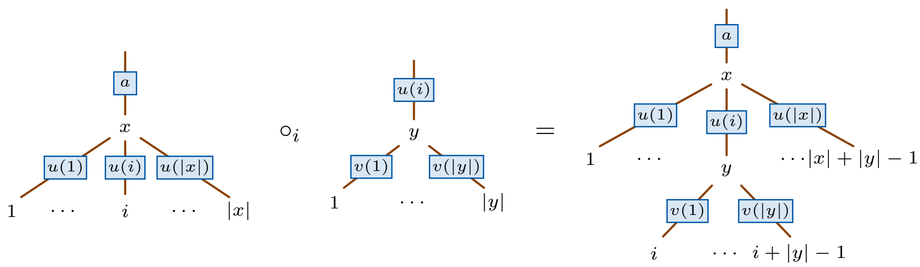
$$\mathbf{1}(\mathrm{out}(x)) \circ_1 x = x = x \circ_i \mathbf{1}(\mathrm{in}_i(x)), \qquad 1 \leq i \leq |x|. \tag{89}$$

Intuitively, an element x of a colored operad having a as output color and $u(i)$ as i-th input color for any $i \in [|x|]$ can be seen as an abstract operator wherein colors are assigned to its output and to each of its inputs. Such an operator is depicted as



, (90)

where the colors of the output and inputs are the squares put on the corresponding edges. The partial composition of two elements x and y of a colored operad expresses pictorially as



. (91)

Most of the definitions about operads recalled in Section 3.1.4 generalize straightforwardly to colored operads. In particular, one can consider the full composition map of a colored operad defined by (14) when its right member is well-defined.

The situation is specific for the homogeneous composition for colored operads. Let $(\mathcal{C}, \circ_i, \mathbf{1})$ be a colored operad. The *homogeneous composition* map of $\mathcal{C}$ is the map

$$\odot : \mathcal{C}(a, u) \times \mathcal{C}(b, v) \to \mathcal{C}, \qquad a, b \in \mathfrak{C}, \quad u, v \in \mathfrak{C}^*, \tag{92}$$

defined, for any $x \in \mathcal{C}(a, u)$ and $y \in \mathcal{C}(b, v)$, by using the full composition map, by

$$x \odot y := x \circ \left[y_1, \dots, y_{|x|}\right], \tag{93}$$

where for any $i \in [|x|]$,

$$y_i := \begin{cases} y & \text{if } \mathrm{in}_i(x) = \mathrm{out}(y), \\ \mathbf{1}(\mathrm{in}_i(x)) & \text{otherwise.} \end{cases} \tag{94}$$

Intuitively, $x \odot y$ is obtained by grafting simultaneously the outputs of copies of y onto the inputs of x having the same color as the output color of y. If the set of colors of $\mathcal{C}$ is a singleton, the homogeneous composition map of $\mathcal{C}$ is the homogeneous composition map of operads described in Section 3.1.4.

4.1.2 Bud operads

Let us describe a general construction building a colored operad from a noncolored one introduced in (Giraudo, 2019). Given a noncolored operad $(\mathcal{O}, \circ_i, \mathbf{1})$ and a set $\mathfrak{C}$ of colors, the *$\mathfrak{C}$-bud operad* of $\mathcal{O}$ is the $\mathfrak{C}$-colored operad $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$ defined in the following way. First, $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$ is the $\mathfrak{C}$-colored set defined, for any $a \in \mathfrak{C}$ and $u \in \mathfrak{C}^*$, by

$$\mathsf{B}_{\mathfrak{C}}(\mathcal{O})(a, u) := \{(a, x, u) : x \in \mathcal{O}(\ell(u))\}. \tag{95}$$

Second, the partial composition maps $\circ_i$ of $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$ are defined, for any $(a, x, u), (u(i), y, v) \in \mathsf{B}_{\mathfrak{C}}(\mathcal{O})$, and $i \in [|u|]$, by

$$(a, x, u) \circ_i (u_i, y, v) := (a, x \circ_i y, u \circ_i v) \tag{96}$$

where the first occurrence of $\circ_i$ in the right member of (96) is the partial composition map of $\mathcal{O}$ and the second one is a substitution of words: $u \circ_i v$ is the word obtained by replacing in u the i-th letter $u(i)$ of u by v. Finally, the colored unit map $\mathbf{1}$ of $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$ is defined by $\mathbf{1}(a) := (a, \mathbf{1}, a)$ for any $a \in \mathfrak{C}$, where $\mathbf{1}$ is the unit of $\mathcal{O}$. The *pruning* $\mathrm{pr}((a, x, u))$ of an element (a, x, u) of $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$ is the element x of $\mathcal{O}$.

Intuitively, this construction consists in forming a colored operad $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$ out of $\mathcal{O}$ by surrounding its elements with an output color and input colors coming from $\mathfrak{C}$ in all possible ways.

For a fixed degree monoid D, we apply this construction to the D-music box operad by setting, for any set $\mathfrak{C}$ of colors,

$$\mathsf{BP}^D_{m,\mathfrak{C}} := \mathsf{B}_{\mathfrak{C}}\left(\mathsf{P}^D_m\right). \tag{97}$$

We call $\mathsf{BP}^D_{m,\mathfrak{C}}$ the *$\mathfrak{C}$-bud D-music box operad*. The elements of $\mathsf{BP}^D_{m,\mathfrak{C}}$ are called *$\mathfrak{C}$-colored multi-patterns*. For instance, for $\mathfrak{C} := \{\mathtt{b}_1, \mathtt{b}_2, \mathtt{b}_3\}$,

$$\left(\mathtt{b}_1, \begin{vmatrix} 1 & \square & 0 & \square & 1 \\ \bar{7} & \square & 0 & 0 & \square \end{vmatrix}, \mathtt{b}_2\mathtt{b}_2\mathtt{b}_1\right) \tag{98}$$

is a $\mathfrak{C}$-colored multi-pattern. Moreover, in the colored operad $\mathsf{BP}^{\mathbb{Z}}_{2,\mathfrak{C}}$, one has

$$\left(\mathtt{b}_3, \begin{vmatrix} 0 & 1 & \square \\ \bar{1} & \square & 0 \end{vmatrix}, \mathtt{b}_2\mathtt{b}_1\right) \circ_2 \left(\mathtt{b}_1, \begin{vmatrix} 1 & 1 & 2 \\ 2 & \bar{1} & \bar{2} \end{vmatrix}, \mathtt{b}_3\mathtt{b}_3\mathtt{b}_2\right) = \left(\mathtt{b}_3, \begin{vmatrix} 0 & 2 & 2 & 3 & \square \\ \bar{1} & \square & 2 & \bar{1} & \bar{2} \end{vmatrix}, \mathtt{b}_2\mathtt{b}_3\mathtt{b}_3\mathtt{b}_2\right). \tag{99}$$

The intuition that justifies the introduction of these colored versions of multi-patterns and of the D-music box operad is that colors restrict the right to perform the composition of two given multi-patterns. In this way, on a small scale, one can for instance forbid some intervals in the musical phrases specified by the multi-patterns of a suboperad of $\mathsf{BP}^D_{m,\mathfrak{C}}$ generated by a given set of $\mathfrak{C}$-colored multi-patterns. On a larger scale, the colors allow us to build complex phrases by imposing some specific parts (for instance, a color might be used for the beginning of a piece, another for the middle, and a third one for the end).

Besides, given a set $\mathfrak{G}$ of $\mathfrak{C}$-colored multi-patterns, the elements of the suboperad of $\mathsf{BP}^D_{m,\mathfrak{C}}$ generated by $\mathfrak{G}$ are obtained by composing elements of $\mathfrak{G}$. Therefore, in some sense, these elements inherit from properties of the multi-patterns of $\mathfrak{G}$. The next section uses these ideas to propose random generation algorithms outputting new multi-patterns from existing ones in a controlled way.

4.2 Bud generating systems

We describe here a sort of generating systems using operads and colored operads introduced in (Giraudo, 2019). Slight variations are considered in this present work.

4.2.1 Bud generating systems

A *bud generating system* (Giraudo, 2019) is a tuple $(\mathcal{O}, \mathfrak{C}, \mathcal{R}, \mathtt{b})$ where

(i) $(\mathcal{O}, \circ_i, \mathbf{1})$ is an operad, called *ground operad*;

(ii) $\mathfrak{C}$ is a set of colors;

(iii) $\mathcal{R}$ is a finite subset of $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$, called *set of rules*;

(iv) $\mathtt{b}$ is a color of $\mathfrak{C}$, called *initial color*.

For any color $a \in \mathfrak{C}$, we shall denote by $\mathcal{R}(a)$ the set of the rules of $\mathcal{R}$ having a as output color.

For instance, $\mathcal{B}_{\mathrm{ex}} := \left(\mathsf{P}^{\mathbb{Z}}_2, \{\mathtt{b}_1, \mathtt{b}_2, \mathtt{b}_3\}, \{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5\}, \mathtt{b}_1\right)$ where

$$\mathbf{c}_1 := \left(\mathtt{b}_1, \begin{vmatrix} 0 & 2 & \square & 1 & \square & 0 & 4 \\ \bar{5} & \square & \square & 0 & 0 & 0 & 0 \end{vmatrix}, \mathtt{b}_3\mathtt{b}_2\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\right), \qquad \mathbf{c}_2 := \left(\mathtt{b}_1, \begin{vmatrix} 1 & \square & 0 \\ 0 & \square & 1 \end{vmatrix}, \mathtt{b}_1\mathtt{b}_1\right),$$
$$\mathbf{c}_3 := \left(\mathtt{b}_2, \begin{vmatrix} \bar{1} \\ \bar{1} \end{vmatrix}, \mathtt{b}_1\right), \qquad \mathbf{c}_4 := \left(\mathtt{b}_2, \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix}, \mathtt{b}_1\mathtt{b}_1\right), \qquad \mathbf{c}_5 := \left(\mathtt{b}_3, \begin{vmatrix} 0 \\ 0 \end{vmatrix}, \mathtt{b}_3\right) \tag{100}$$

is a bud generating system.

Bud generating systems are devices similar to context-free formal grammars (Hopcroft et al., 2006) wherein colors play the role of nonterminal symbols. Each element $(a, x, u) \in \mathcal{R}$ can be seen as a production rule of the form $a \to (x, u)$. The color a plays the role of a nonterminal symbol, u is a word of symbols wherein each letter can be seen as a terminal or a nonterminal symbol, and x denotes additional information. As we shall see, this information is very important because bud generating systems are intended to generate elements of $\mathcal{O}$.

If context-free grammars are devices designed to generate sets of words, bud generating systems are designed to generate more general combinatorial objects. More precisely, a bud generating system $(\mathcal{O}, \mathfrak{C}, \mathcal{R}, \mathtt{b})$ allows us to build elements of $\mathcal{O}$ by following three different operating modes. We describe in the next sections the three corresponding random generation algorithms. These algorithms are in particular intended to work with P^D_m as ground operad in order to generate multi-patterns.

4.2.2 Random generation

In the next three sections, we shall design three random generation algorithms with the goal to randomly generate elements of $\mathcal{O}$ given a bud generating system having $\mathcal{O}$ as ground operad. In these sections, $\mathcal{B} := (\mathcal{O}, \mathfrak{C}, \mathcal{R}, \mathtt{b})$ is a bud generating system. We shall also consider the bud generating system $\mathcal{B}_{\mathrm{ex}}$ introduced in Section 4.2.1 to provide some concrete examples. Besides, for any finite and nonempty set S, $\textsc{random}(S)$ is an element of S picked uniformly at random among the elements of S.

4.2.3 Partial generation

Let $\xrightarrow{\circ_i}$ be the binary relation on $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$ such that $(a, x, u) \xrightarrow{\circ_i} (a, y, v)$ if there is a rule $r \in \mathcal{R}$ and $i \in [|u|]$ such that

$$(a, y, v) = (a, x, u) \circ_i r. \tag{101}$$

An element x of $\mathcal{O}$ is *partially generated* by $\mathcal{B}$ if there is an element $(\mathtt{b}, x, u)$ such that $(\mathtt{b}, \mathbf{1}, \mathtt{b})$ is in relation with $(\mathtt{b}, x, u)$ w.r.t. the reflexive and transitive closure of $\xrightarrow{\circ_i}$.

For instance, by considering the bud generating system $\mathcal{B}_{\mathrm{ex}}$, since

$$\begin{aligned}
\left(\mathtt{b}_1, \begin{vmatrix} 0 \\ 0 \end{vmatrix}, \mathtt{b}_1\right) &\xrightarrow{\circ_i} \left(\mathtt{b}_1, \begin{vmatrix} 1 & \square & 0 \\ 0 & \square & 1 \end{vmatrix}, \mathtt{b}_1\mathtt{b}_1\right) \\
&\xrightarrow{\circ_i} \left(\mathtt{b}_1, \begin{vmatrix} 1 & \square & 0 & 2 & \square & 1 & \square & 0 & 4 \\ 0 & \square & \bar{4} & \square & \square & 1 & 1 & 1 & 1 \end{vmatrix}, \mathtt{b}_1\mathtt{b}_3\mathtt{b}_2\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\right) \\
&\xrightarrow{\circ_i} \left(\mathtt{b}_1, \begin{vmatrix} 1 & \square & 0 & 2 & 2 & \square & 1 & \square & 0 & 4 \\ 0 & \square & \bar{4} & \square & \square & 1 & 1 & 1 & 1 & 1 \end{vmatrix}, \mathtt{b}_1\mathtt{b}_3\mathtt{b}_1\mathtt{b}_1\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\right),
\end{aligned} \tag{102}$$

the multi-pattern

$$\begin{vmatrix} 1 & \square & 0 & 2 & 2 & \square & 1 & \square & 0 & 4 \\ 0 & \square & \bar{4} & \square & \square & 1 & 1 & 1 & 1 & 1 \end{vmatrix} \tag{103}$$

is partially generated by $\mathcal{B}_{\mathrm{ex}}$.

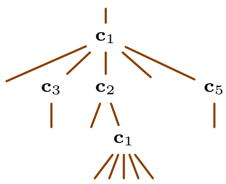
Algorithm 1 returns an element partially generated by $\mathcal{B}$ obtained by applying at most k rules to the initial element $(\mathtt{b}, \mathbf{1}, \mathtt{b})$. The execution of the algorithm builds a syntax tree of elements of $\mathcal{R}$ with at most k internal nodes.

Algorithm 1: Partial random generation.

Data: A bud generating system $\mathcal{B} := (\mathcal{O}, \mathfrak{C}, \mathcal{R}, \mathtt{b})$ and an integer $k \in \mathbb{N}$.
Result: A randomly generated element of $\mathcal{O}$.

```
begin
    x ← (b, 1, b)
    for j ∈ [k] do
        i ← RANDOM([|x|])
        if R(in_i(x)) ≠ ∅ then
            r ← RANDOM(R(in_i(x)))
            x ← x ∘_i r
    return pr(x)
```

For instance, by considering the previous bud generating system $\mathcal{B}_{\mathrm{ex}}$, this algorithm run with $k := 5$ builds the syntax tree of colored multi-patterns



(104)

which produces, when evaluated, the multi-pattern

$$\begin{vmatrix} 0 & 1 & \square & 2 & \square & 1 & 3 & \square & 2 & \square & 1 & 5 & \square & 0 & 4 \\ \bar{5} & \square & \square & \bar{1} & 0 & \square & \bar{4} & \square & \square & 1 & 1 & 1 & 1 & 0 & 0 \end{vmatrix}. \tag{105}$$

4.2.4 Full generation

Let $\xrightarrow{\circ}$ be the binary relation on $\mathrm{B}_{\mathfrak{C}}(\mathcal{O})$ such that $(a, x, u) \xrightarrow{\circ} (a, y, v)$ if there are rules $r_1, \dots, r_{|x|} \in \mathcal{R}$ such that

$$(a, y, v) = (a, x, u) \circ \left[r_1, \dots, r_{|x|}\right]. \tag{106}$$

An element x of $\mathcal{O}$ is *fully generated* by $\mathcal{B}$ if there is an element $(\mathtt{b}, x, u)$ such that $(\mathtt{b}, \mathbf{1}, \mathtt{b})$ is in relation with $(\mathtt{b}, x, u)$ w.r.t. the reflexive and transitive closure of $\xrightarrow{\circ}$.

For instance, by considering the bud generating system $\mathcal{B}_{\mathrm{ex}}$, since

$$\left(\mathtt{b}_1, \begin{vmatrix} 0 \\ 0 \end{vmatrix}, \mathtt{b}_1\right) \xrightarrow{\circ} \left(\mathtt{b}_1, \begin{vmatrix} 0 & 2 & \square & 1 & \square & 0 & 4 \\ \bar{5} & \square & \square & 0 & 0 & 0 & 0 \end{vmatrix}, \mathtt{b}_3\mathtt{b}_2\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\right)$$
$$\xrightarrow{\circ} \left(\mathtt{b}_1, \begin{vmatrix} 0 & 1 & \square & 2 & \square & 1 & \square & 0 & 2 & \square & 1 & \square & 0 & 4 & 4 \\ \bar{5} & \square & \square & \bar{1} & 0 & \square & 1 & \bar{5} & \square & \square & 0 & 0 & 0 & 0 & 0 \end{vmatrix}, \mathtt{b}_3\mathtt{b}_1\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\mathtt{b}_2\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\mathtt{b}_3\right), \tag{107}$$

the multi-pattern

$$\begin{vmatrix} 0 & 1 & \square & 2 & \square & 1 & \square & 0 & 2 & \square & 1 & \square & 0 & 4 & 4 \\ \bar{5} & \square & \square & \bar{1} & 0 & \square & 1 & \bar{5} & \square & \square & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \tag{108}$$

is fully generated by $\mathcal{B}_{\mathrm{ex}}$.

Bud generating systems together with this scheme for generation are very similar to Lindenmayer systems, which are sorts of formal grammars (Lindenmayer, 1968). Such systems lead to frameworks to generate musical phrases (Hudak & Quick, 2018; Mason & Saffle, 1994; McCormack, 1996).

Algorithm 2 returns an element synchronously generated by $\mathcal{B}$ obtained by applying at most k rules to the initial element $(\mathtt{b}, \mathbf{1}, \mathtt{b})$. The execution of the algorithm builds a syntax tree of elements of $\mathcal{R}$ of height at most $k+1$ wherein the leaves are all at the same distance from the root.

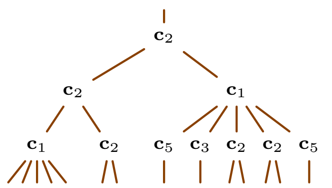
Algorithm 2: Full random generation.

Data: A bud generating system $\mathcal{B} := (\mathcal{O}, \mathfrak{C}, \mathcal{R}, \mathtt{b})$ and an integer $k \in \mathbb{N}$.
Result: A randomly generated element of $\mathcal{O}$.

begin
- $x \leftarrow (\mathtt{b}, \mathbf{1}, \mathtt{b})$
- **for** $j \in [k]$ **do**
  - $R \leftarrow \mathcal{R}(\mathrm{in}_1(x)) \times \cdots \times \mathcal{R}\big(\mathrm{in}_{|x|}(x)\big)$
  - **if** $R \neq \emptyset$ **then**
    - $\big(r_1, \dots, r_{|x|}\big) \leftarrow \mathrm{RANDOM}(R)$
    - $x \leftarrow x \circ \big[r_1, \dots, r_{|x|}\big]$
- **return** $\mathrm{pr}(x)$

Observe that when for all colors $\mathtt{b} \in \mathfrak{C}$, the sets $\mathcal{R}(\mathtt{b})$ have no more than one element, this algorithm is deterministic.

For instance, by considering the previous bud generating system $\mathcal{B}_{\mathrm{ex}}$, this algorithm run with $k := 2$ builds the syntax tree of colored multi-patterns



(109)

which produces, when evaluated, the multi-pattern

$$\begin{vmatrix} 2 & 4 & \square & 3 & \square & 2 & 6 & \square & 2 & \square & 1 & \square & 0 & 1 & \square & 2 & \square & 1 & \square & 1 & \square & 0 & 4 \\ \bar{5} & \square & \square & 0 & 0 & 0 & 0 & \square & 1 & \square & 2 & \square & \bar{4} & \square & \square & 0 & 1 & \square & 2 & 1 & \square & 2 & 1 \end{vmatrix}. \tag{110}$$

4.2.5 Homogeneous generation

Let $\xrightarrow{\odot}$ be the binary relation on $\mathsf{B}_{\mathfrak{C}}(\mathcal{O})$ such that $(a, x, u) \xrightarrow{\odot} (a, y, v)$ if there is a rule $r \in \mathcal{R}$ such that

$$(a, y, v) = (a, x, u) \odot r. \tag{111}$$

An element x of $\mathcal{O}$ is *homogeneously generated* by $\mathcal{B}$ if there is an element $(\mathtt{b}, x, u)$ such that $(\mathtt{b}, \mathbf{1}, \mathtt{b})$ is in relation with $(\mathtt{b}, x, u)$ w.r.t. the reflexive and transitive closure of $\xrightarrow{\odot}$.

For instance, by considering the bud generating system $\mathcal{B}_{\mathrm{ex}}$, since

$$\begin{aligned}
\left(\mathtt{b}_1, \left|\begin{matrix}0\\0\end{matrix}\right|, \mathtt{b}_1\right) &\xrightarrow{\odot} \left(\mathtt{b}_1, \left|\begin{matrix}0\;2\;\square\;1\;\square\;0\;4\\ \bar{5}\;\square\;\square\;0\;0\;0\;0\end{matrix}\right|, \mathtt{b}_3\mathtt{b}_2\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\right)\\
&\xrightarrow{\odot} \left(\mathtt{b}_1, \left|\begin{matrix}0\;2\;\square\;2\;\square\;1\;\square\;1\;\square\;0\;4\\ \bar{5}\;\square\;\square\;0\;0\;\square\;1\;0\;\square\;1\;0\end{matrix}\right|, \mathtt{b}_3\mathtt{b}_2\mathtt{b}_1\mathtt{b}_1\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\right)\\
&\xrightarrow{\odot} \left(\mathtt{b}_1, \left|\begin{matrix}0\;1\;\square\;2\;\square\;1\;\square\;1\;\square\;0\;4\\ \bar{5}\;\square\;\square\;\bar{1}\;0\;\square\;1\;0\;\square\;1\;0\end{matrix}\right|, \mathtt{b}_3\mathtt{b}_1\mathtt{b}_1\mathtt{b}_1\mathtt{b}_1\mathtt{b}_1\mathtt{b}_3\right),
\end{aligned} \tag{112}$$

the multi-pattern

$$\left|\begin{matrix}0\;1\;\square\;2\;\square\;1\;\square\;1\;\square\;0\;4\\ \bar{5}\;\square\;\square\;\bar{1}\;0\;\square\;1\;0\;\square\;1\;0\end{matrix}\right| \tag{113}$$

is homogeneously generated by $\mathcal{B}_{\mathrm{ex}}$.

Algorithm 3 returns an element homogeneously generated by $\mathcal{B}$ obtained by applying at most k rules to the initial element $(\mathtt{b}, \mathbf{1}, \mathtt{b})$. The execution of the algorithm builds a syntax tree of elements of height at most $k + 1$.

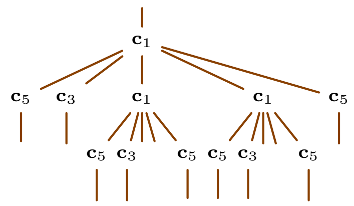
Algorithm 3: Homogeneous random generation.

Data: A bud generating system $\mathcal{B} := (\mathcal{O}, \mathfrak{C}, \mathcal{R}, \mathtt{b})$ and an integer $k \in \mathbb{N}$.
Result: A randomly generated element of $\mathcal{O}$.

begin
    $x \leftarrow (\mathtt{b}, \mathbf{1}, \mathtt{b})$
    for $j \in [k]$ **do**
        if $\mathcal{R} \neq \emptyset$ **then**
            $r \leftarrow \mathrm{RANDOM}(\mathcal{R})$
            $r \leftarrow x \odot r$
    return $\mathrm{pr}(x)$

Observe that when the set of rules $\mathcal{R}$ has no more than one element, this algorithm is deterministic.

For instance, by considering the previous bud generating system $\mathcal{B}_{\mathrm{ex}}$, this algorithm run with $k := 3$ builds the syntax tree of colored multi-patterns



(114)

which produces, when evaluated, the multi-pattern

$$\left|\begin{matrix}0\;1\;\square\;1\;2\;\square\;2\;\square\;1\;5\;\square\;0\;1\;\square\;1\;\square\;0\;4\;4\\ \bar{5}\;\square\;\square\;\bar{1}\;\bar{5}\;\square\;\square\;\bar{1}\;0\;0\;0\;\bar{5}\;\square\;\square\;\bar{1}\;0\;0\;0\;0\end{matrix}\right|. \tag{115}$$

5 Applications

In this part, we introduce some basic notions about an implementation of the bud music box model. Next, we construct three particular bud generating systems devoted to work with the algorithms introduced in Section 4.2. They generate variations of multi-patterns placed at input, with possibly some auxiliary data. All these constructions are illustrated with some small examples. Finally, we use these constructions to propose a bud generating system for the random generation of a complete musical piece.

5.1 The *Bud Music Box* system

The music box model, the related operads, and the related random generation algorithms have been implemented by the author. The program, named *Bud Music Box*, as well as its complete documentation are available at (Giraudo, 2023). We review some of the main instructions of the language thus designed, useful to understand the examples of the next sections.

5.1.1 Context instructions

Here are the main instructions of the language used to specify the context in which the multi-patterns will be interpreted:

- **scale** *i_1 ... i_k* sets the scale specified by the sequence *i_1 ... i_k* designating the distances in semitones between two consecutive notes. The multi-patterns of the program are interpreted through this scale.
- **root** *note* sets the root note specified by the MIDI code *note*. The multi-patterns of the program are interpreted so that the degree 0 corresponds with the specified MIDI note.
- **tempo** *v* sets the tempo specified by the value *v* designating a tempo in beats per minute. The multi-patterns of the program are interpreted through this specified tempo.
- **sounds** *s_1 ... s_k* sets the sounds specified by the sequence *s_1 ... s_k* of General MIDI programs (which are nonnegative integers). The code *s_i* designates the MIDI instrument for the i-th voice of each multi-pattern of the program.
- **monoid** *dm* set *dm* as the degree monoid on which the multi-patterns of the program are defined. The possible values for *dm* are **add** for the degree monoid $\mathbb{Z}$, **cyclic** *k* for the degree monoid $\mathbb{C}_k$, and **max** *z* for the degree monoid $\mathbb{M}_Z$ where Z is the set of the integers greater than or equal to z.

5.1.2 Multi-pattern operation instructions

Here are the main instructions of the language used to define, modify, and generate multi-patterns:

- **multi-pattern** *mpat description* creates a multi-pattern identified by *mpat*. It is specified by *description* which encodes a multi-pattern by the sequence of its patterns separated by `+` symbols, where degrees are signed integers and rests are encoded by `.` symbols.
- **stack** *mpat mpat_1 ... mpat_r* creates a multi-pattern identified by *mpat* obtained by stacking the existing multi-patterns identified by *mpat_1*, ..., *mpat_r*.
- **mirror** *mpat mpat'* creates a multi-pattern identified by *mpat* and whose value is the retrograde of an existing multi-pattern identified by *mpat'*.
- **inverse** *mpat mpat'* creates a multi-pattern identified by *mpat* and whose value is the inverse image of an existing multi-pattern identified by *mpat'*.
- **colorize** *cmpat %col_out mpat %col_in_1 ... %col_in_n* creates a colored multi-pattern identified by *cmpat* and whose value is the triple having *%col_out* as output color, the existing multi-pattern identified by *mpat* as multi-pattern, and the sequence *%col_in_1 ... %col_in_n* as input colors.
- **mono-colorize** *cmpat %col_out mpat %col_in* creates a colored multi-pattern identified by *cmpat* and whose value is the triple having *%col_out* as output color, the existing multi-pattern identified by *mpat* as multi-pattern, and the sequence made of the single color *%col_in* as input colors. This sequence has automatically the right size according to the arity of the multi-pattern identified by *mpat*.
- **generate** *mpat mode k %col_init cmpat_1 ... cmpat_r* creates a multi-pattern identified by *mpat* and whose value is a multi-pattern randomly generated by the partial (resp. full,

homogeneous) generation algorithm if *mode* is equal to **partial** (resp. **full**, **homogeneous**), run with its parameter k equal to *k* on the bud generating system $\left(\mathsf{P}_m^D, \mathfrak{C}, \mathcal{R}, \mathtt{b}\right)$ defined as follows. The degree monoid D is the one which has been specified in the context. The set of rules $\mathcal{R}$ consists of the existing colored multi-patterns identified by *cmpat_1*, ..., *cmpat_r*, m is the common multiplicity of the multi-patterns appearing in $\mathcal{R}$, $\mathtt{b}$ is the color *%col_init*, and $\mathfrak{C}$ is the smallest set of colors containing the involved colors.

5.2 Particular bud generating systems

We introduce here monochrome bud generating systems, which are basically bud generating systems where colors do not play any role, and use these to construct a bud generating system devoted to randomly generate mixes of multi-patterns. We describe also three bud generating systems devoted to randomly generate multi-patterns from a single pattern and some auxiliary data.

5.2.1 Monochrome bud generating systems

A bud generating system $\mathcal{B}$ is *monochrome* if its set of colors is a singleton $\{\mathtt{b}\}$. In this case, the fact that there is only one color implies that there is no constraint for the application of the rules of $\mathcal{B}$. Any pair $(\mathcal{O}, R)$ where $\mathcal{O}$ is an operad and R is a finite subset of $\mathcal{O}$ specifies the monochrome bud generating system $(\mathcal{O}, \{\mathtt{b}\}, \mathcal{R}, \mathtt{b})$ where $\mathcal{R}$ is the set of the rules $\left(\mathtt{b}, x, \mathtt{b}^{|x|}\right)$ for all $x \in R$.

5.2.2 Style emulation

Given a set M of multi-patterns all of the same multiplicity m, the *mix bud generating system* $\mathcal{B}_M^{\mathrm{mix}}$ of M is the monochrome bud generating system specified by the pair $\left(\mathsf{P}_m^{\mathbb{Z}}, M\right)$. The partial, full, and homogeneous random generation algorithms run with $\mathcal{B}_M^{\mathrm{mix}}$ randomly generate multi-patterns obtained by composing elements of M together. Such a multi-pattern $\mathbf{m}$ potentially inherits characteristics from the multi-patterns of M. Hence, $\mathcal{B}_M^{\mathrm{mix}}$ can be seen as a style emulation device.

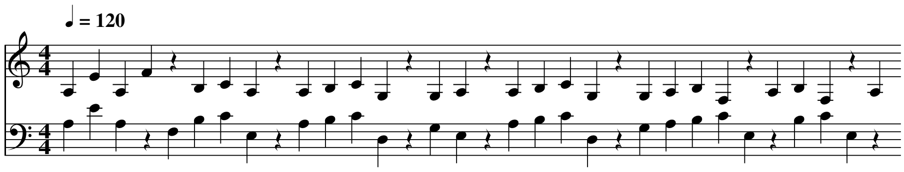
For instance, consider the multi-patterns

$$\mathbf{m}_1 := \left|\begin{array}{ccccccccccccc} 0 & \square & 0 & 1 & 2 & \bar{1} & \square & \bar{1} & 0 & 1 & \bar{2} & \square & 0 \\ \bar{3} & \square & 0 & 1 & 2 & \bar{4} & \square & \bar{1} & 0 & 1 & 2 & \bar{3} & \square \end{array}\right| \quad \text{and} \quad \mathbf{m}_2 := \left|\begin{array}{cccccccc} 0 & 4 & 0 & 5 & \square & 1 & 2 & 0 \\ 0 & 4 & 0 & \square & \bar{2} & 1 & 2 & 0 \end{array}\right|. \tag{116}$$

The partial random generation algorithm run with $\mathcal{B}_M^{\mathrm{mix}}$ where $M := \{\mathbf{m}_1, \mathbf{m}_2\}$ and $k := 3$ as inputs produces the multi-pattern

$$\left|\begin{array}{cccccccccccccccccccccccccccccccc} 0 & 4 & 0 & 5 & \square & 1 & 2 & 0 & \square & 0 & 1 & 2 & \bar{1} & \square & \bar{1} & 0 & \square & 0 & 1 & 2 & \bar{1} & \square & \bar{1} & 0 & 1 & \bar{2} & \square & 0 & 1 & \bar{2} & \square & 0 \\ 0 & 4 & 0 & \square & \bar{2} & 1 & 2 & \bar{3} & \square & 0 & 1 & 2 & \bar{4} & \square & \bar{1} & \bar{3} & \square & 0 & 1 & 2 & \bar{4} & \square & \bar{1} & 0 & 1 & 2 & \bar{3} & \square & 1 & 2 & \bar{3} & \square \end{array}\right|. \tag{117}$$

This multi-pattern interprets as the musical phrase



Here is the *Bud Music Box* program associated with this example:

```
scale 2 1 2 2 1 2 2
root 57
tempo 120
sounds 0 0
monoid add
multi-pattern m1 0 . 0 1 2 -1 . -1 0 1 -2 . 0 + -3 . 0 1 2 -4 . -1 0 1 2 -3 .
multi-pattern m2 0 4 0 5 . 1 2 0 + 0 4 0 . -2 1 2 0
mono-colorize c1 %b m1 %b
mono-colorize c2 %b m2 %b
generate res partial 3 %b c1 c2
```

5.2.3 Horizontal transformations

Given a pattern $\mathbf{p}$ and a sequence Ψ of length $q \geq 1$ of maps on the set of the patterns, the *Ψ-horizontal bud generating system* of $\mathbf{p}$ and Ψ is the monochrome bud generating system $\mathcal{B}^{\mathrm{hor}}_{\mathbf{p},\Psi} := \mathcal{B}^{\mathrm{mix}}_{M}$ where M is the set of multi-patterns

$$\{\Psi(1)(\mathbf{p}), \dots, \Psi(q)(\mathbf{p})\}. \tag{118}$$

For instance, consider the pattern $\mathbf{p} := |0\ 1\ \square\ \bar{1}\ 0\ \square\ 2\ 0\ \square|$ and the sequence Ψ of length 3 of maps such that $\Psi(1)$ is the identity map, $\Psi(2)$ is the retrograde map, and $\Psi(3)$ is the inverse map (see Section 3.3.3). The partial random generation algorithm run with $\mathcal{B}^{\mathrm{hor}}_{\mathbf{p},\Psi}$ and $k := 4$ as inputs produces the multi-pattern

$$|0\ \bar{1}\ \square\ 1\ 2\ \square\ 0\ 1\ \square\ 3\ 1\ \square\ 0\ \square\ \bar{2}\ 0\ 1\ \square\ \bar{1}\ \square\ 0\ 2\ \square\ 0\ \bar{1}\ \square\ 1\ 0\ \square\ 2\ 0\ \square\ \square|. \tag{119}$$

This multi-pattern interprets as the musical phrase

Here is the *Bud Music Box* program associated with this example:

```
scale 2 1 2 2 1 2 2
root 57
tempo 120
sounds 0 0
monoid add
multi-pattern p 0 1 . -1 0 . 2 0 .
mirror p1 p
inverse p2 p
mono-colorize c %b p %b
mono-colorize c1 %b p1 %b
mono-colorize c2 %b p2 %b
generate res partial 4 %b c c1 c2
```

5.2.4 Vertical transformations

Given a pattern $\mathbf{p}$ and a sequence Λ of length $q \geq 1$ of homogeneous operations (see Section 3.3.3) on the set of the patterns, the *Λ-vertical bud generating system* of $\mathbf{p}$ and Λ is the monochrome bud generating system $\mathcal{B}^{\mathrm{ver}}_{\mathbf{p},\Lambda} := \mathcal{B}^{\mathrm{mix}}_{M}$ where M is the singleton containing the multi-pattern

$$\Lambda(\mathbf{p}) := \Lambda(1)(\mathbf{p}) \centerdot \cdots \centerdot \Lambda(q)(\mathbf{p}). \tag{120}$$

For instance, consider the pattern $\mathbf{p} := |0\ 1\ \square\ \bar{1}\ 0\ \square\ 2\ 0\ \square|$ and the sequence Λ of length 2 of homogeneous operations such that $\Lambda(1)$ is the identity map and $\Lambda(2)$ is the retrograde inverse map. The partial random generation algorithm run with $\mathcal{B}^{\mathrm{ver}}_{\mathbf{p},\Lambda}$ and $k := 4$ as inputs produces the multi-pattern

$$\left|\begin{array}{ccccccccccccccccccccccccccccccccc} 0 & 1 & 2 & \square & 0 & 1 & 2 & \square & 0 & 1 & \square & 3 & 4 & \square & 2 & 3 & \square & 5 & 3 & \square & 1 & \square & \square & 3 & 1 & \square & \square & \bar{1} & 0 & \square & 2 & 0 & \square \\ \square & 0 & \square & \bar{2} & \bar{4} & \square & \bar{2} & \square & \bar{1} & \bar{3} & \square & \bar{1} & 0 & \square & \square & \bar{2} & \bar{4} & \square & \bar{2} & \bar{1} & \square & \bar{3} & \bar{2} & \bar{1} & \square & \bar{3} & \bar{2} & \square & 0 & 1 & \square & \bar{1} & 0 \end{array}\right|. \tag{121}$$

This multi-pattern interprets as the musical phrase

Here is the *Bud Music Box* program associated with this example:

```
scale 2 1 2 2 1 2 2
root 57
tempo 120
sounds 0 0
monoid add
multi-pattern p 0 1 . -1 0 . 2 0 .
mirror p1 p
inverse p2 p
mirror p3 p2
stack m p p3
mono-colorize c %b m %b
generate res partial 4 %b c
```

5.2.5 Local variations

Given a pattern $\mathbf{p}$ and a set $M := \{\mathbf{m}_1, \dots, \mathbf{m}_q\}$, $q \geq 0$, of multi-patterns of multiplicity m, the *variation bud generating system* of $\mathbf{p}$ and M is the bud generating system $\mathcal{B}^{\mathrm{var}}_{\mathbf{p},M} := \left(\mathsf{P}^{\mathbb{Z}}_m, \mathfrak{C}, \mathcal{R}, \mathtt{b}_1\right)$ where $\mathfrak{C}$ is the set of colors $\{\mathtt{b}_1, \mathtt{b}_2, \mathtt{b}_3\}$ and $\mathcal{R}$ is the set of rules containing the colored multi-patterns

$$\left(\mathtt{b}_1, \mathrm{cp}_m(\mathbf{p}), \mathtt{b}_2^{|\mathbf{p}|}\right), \left(\mathtt{b}_2, \mathrm{cp}_m(\mathbf{p}), \mathtt{b}_2^{|\mathbf{p}|}\right), \left(\mathtt{b}_2, \mathbf{m}_1, \mathtt{b}_3^{|\mathbf{m}_1|}\right), \dots, \left(\mathtt{b}_2, \mathbf{m}_q, \mathtt{b}_3^{|\mathbf{m}_q|}\right), (\mathtt{b}_3, \mathrm{cp}_m(0), \mathtt{b}_3). \tag{122}$$

The partial, full, and homogeneous random generation algorithms run with $\mathcal{B}^{\mathrm{var}}_{\mathbf{p},M}$ randomly generate multi-patterns of multiplicity m obtained by self-composing $\mathrm{cp}_m(\mathbf{p})$ and then, by transforming some beats of the obtained multi-pattern by composing them with elements of M. The color $\mathtt{b}_1$ is the initial color in order to start the generation with $\mathrm{cp}_m(\mathbf{p})$, $\mathtt{b}_2$ is an intermediate color, and the color $\mathtt{b}_3$ prevents compositions of patterns of M with $\mathrm{cp}_m(\mathbf{p})$ or with other patterns of M in order to not create degenerated results. The rule $(\mathtt{b}_3, \mathrm{cp}_m(0), \mathtt{b}_3)$ is important since this rule admits $\mathtt{b}_3$ as output color, what is required in the case where this bud generating system is used with the full random generation algorithm.

Observe that a chord of M acts by changing one beat of the current generated pattern by an harmonized version of it. Similarly, a flat multi-pattern of M acts by changing one beat of the current generated pattern by a scheme of beats having the same degree. Finally, an arpeggio of M acts by changing one beat of the current generated pattern by an arpeggiated version of it.

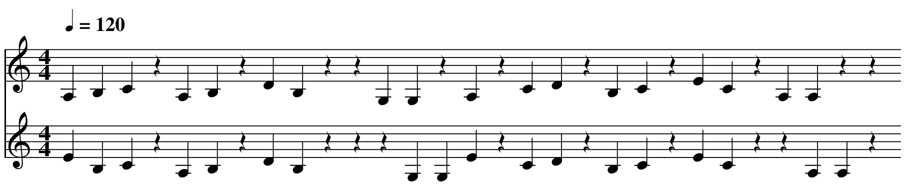
For instance, consider the multi-patterns

$$\mathbf{p} := \begin{vmatrix} 0 & 1 & \square & \bar{1} & 0 & \square & 2 & 0 & \square \end{vmatrix}, \qquad \mathbf{m}_1 := \begin{vmatrix} 0 & 0 & \square \\ \square & 0 & 0 \end{vmatrix}, \quad \text{and} \quad \mathbf{m}_2 := \begin{vmatrix} 0 \\ 4 \end{vmatrix}. \tag{123}$$

The full random generation algorithm run with $\mathcal{B}^{\mathrm{var}}_{\mathbf{p},M}$ where $M := \{\mathbf{m}_1, \mathbf{m}_2\}$ and $k := 2$ as inputs produces the multi-pattern

$$\begin{vmatrix} 0 & 1 & 2 & \square & 0 & 1 & \square & 3 & 1 & \square & \square & \bar{1} & \bar{1} & \square & 0 & \square & 2 & 3 & \square & 1 & 2 & \square & 4 & 2 & \square & 0 & 0 & \square & \square \\ 4 & 1 & 2 & \square & 0 & 1 & \square & 3 & 1 & \square & \square & \square & \bar{1} & \bar{1} & 4 & \square & 2 & 3 & \square & 1 & 2 & \square & 4 & 2 & \square & \square & 0 & 0 & \square \end{vmatrix}. \tag{124}$$

This multi-pattern interprets as the musical phrase



Here is the *Bud Music Box* program associated with this example:

```
scale 2 1 2 2 1 2 2
root 57
tempo 120
sounds 0 0
monoid add
multi-pattern p 0 1 . -1 0 . 2 0 .
stack m p p
multi-pattern m1 0 0 . + . 0 0
multi-pattern m2 0 + 4
multi-pattern u 0 + 0
mono-colorize c1 %b1 m %b2
mono-colorize c2 %b2 m %b2
mono-colorize cm1 %b2 m1 %b3
colorize cm2 %b2 m2 %b3
colorize cu %b3 u %b3
generate res full 2 %b1 c1 c2 cm1 cm2 cu
```

5.3 Generating structured pieces

With the aim to use bud generating systems to produce complete pieces, we first define a notion of composition of bud generating systems. Then, we use the constructions presented in Section 5.2 to propose an example of a piece generated by the tools presented by this paper.

5.3.1 Composition of bud generating systems

Let $\mathcal{O}$ be an operad, x be an element of arity $n \in \mathbb{N}$ of $\mathcal{O}$, and $\mathcal{B}_i := (\mathcal{O}, \mathfrak{C}_i, \mathcal{R}_i, \mathtt{b}_i)$, $i \in [n]$, be bud generating systems such that for all $i \neq i' \in [n]$, $\mathfrak{C}_i$ and $\mathfrak{C}_{i'}$ are disjoint. The *composition* of x with $\mathcal{B}_1, \dots, \mathcal{B}_n$ is the bud generating system $x \circ [\mathcal{B}_1, \dots, \mathcal{B}_n] := (\mathcal{O}, \mathfrak{C}, \mathcal{R}, \mathtt{b})$ such that

$$\mathfrak{C} := \left(\bigsqcup_{i \in [n]} \mathfrak{C}_i \right) \sqcup \{\mathtt{b}\} \quad \text{and} \quad \mathcal{R} := \left(\bigsqcup_{i \in [n]} \mathcal{R}_i \right) \sqcup \{(\mathtt{b}, x, \mathtt{b}_1 \dots \mathtt{b}_n)\}. \tag{125}$$

It is easy to see that any element partially (resp. fully, homogeneously) generated by $x \circ [\mathcal{B}_1, \dots, \mathcal{B}_n]$ is of the form $x \circ [x_1, \dots, x_n]$ where for any $i \in [n]$, x_i is an element partially (resp. fully, homogeneously) generated by $\mathcal{B}_i$.

In this context of bud generating systems having $\mathsf{P}_m^{\mathbb{Z}}$, $m \geq 1$, as ground operad, this construction is useful to generate complete musical pieces. Indeed, given a multi-pattern $\mathbf{m} \in \mathsf{P}_m^{\mathbb{Z}}$ of arity $n \in \mathbb{N}$ and bud generating systems $\mathcal{B}_i$, $i \in [n]$, having $\mathsf{P}_m^{\mathbb{Z}}$ as ground operads, the partial, full, and homogeneous random generation algorithms run with $\mathbf{m} \circ [\mathcal{B}_1, \dots, \mathcal{B}_n]$ generate multi-patterns formed of n parts such that each $\mathcal{B}_i$ directs the i-th part, and such that $\mathbf{m}$ connects all the parts together.

5.3.2 A complete example

Let us use the composition of bud generating systems and some of the previous constructions to provide a complete example of the generation of a musical piece. Let Λ be the sequence of length 2 of homogeneous operations such that $\Lambda(1)$ is the identity map and $\Lambda(2)$ is the retrograde inverse map. Let the multi-patterns

$$\mathbf{m} := \begin{vmatrix} 7 & \bar{1} & 7 & 14 & 2 & 14 \\ 0 & \bar{8} & 0 & 0 & \bar{5} & 0 \end{vmatrix}, \qquad \mathbf{p}_1 := |0 \ \square 0 \ \bar{2} \ 2 \ \square \ \bar{1} \ 0|, \qquad \mathbf{p}_2 := |0 \ 2 \ \square \ \bar{2} \ 0 \ 2 \ 0 \ \square|, \tag{126}$$

and, by using the notation introduced in (120), $\mathbf{p}'_1 := \Lambda(\mathbf{p}_1)$ and $\mathbf{p}'_2 := \Lambda(\mathbf{p}_2)$. Let also the bud generating systems $\mathcal{B}_1 := \mathcal{B}_2 := \mathcal{B}^{\mathrm{ver}}_{\mathbf{p}_1, \Lambda}$, $\mathcal{B}_3 := \mathcal{B}_4 := \mathcal{B}^{\mathrm{mix}}_{\{\mathbf{p}'_1, \mathbf{p}'_2\}}$, $\mathcal{B}_5 := \mathcal{B}_6 := \mathcal{B}^{\mathrm{ver}}_{\mathbf{p}_2, \Lambda}$, and $\mathcal{B} := \mathbf{m} \circ [\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6]$.

Let us discuss about the structure of a multi-pattern generated by any of the three random generation algorithm run with $\mathcal{B}$. Since $\mathbf{m}$ is of arity 6, such a multi-pattern consists of six parts, having the following properties:

1. The first one consists in self-composing $\mathbf{p}'_1$, where the first voice is transposed one octave up (since $\mathbf{m}(1)(1) = 7$).
2. The second one consists in self-composing $\mathbf{p}'_1$ where the first voice is transposed one degree down (since $\mathbf{m}(2)(1) = \bar{1}$) and the second voice is transposed one degree and one octave down (since $\mathbf{m}(2)(2) = \bar{8}$).
3. The third one consists in composing freely $\mathbf{p}'_1$ and $\mathbf{p}'_2$ together where the first voice is transposed one octave up (since $\mathbf{m}(3)(1) = 7$).
4. The fourth one consists in composing freely $\mathbf{p}'_1$ and $\mathbf{p}'_2$ together where the first voice is transposed two octaves up (since $\mathbf{m}(4)(1) = 14$).
5. The fifth one consists in self-composing $\mathbf{p}'_2$ where the first voice is transposed two degrees up (since $\mathbf{m}(5)(1) = 2$) and the second voice is transposed five degrees down (since $\mathbf{m}(5)(2) = \bar{5}$).
6. The sixth one consists in self-composing $\mathbf{p}'_2$ where the first voice is transposed two octaves up (since $\mathbf{m}(6)(1) = 14$).

Here is the *Bud Music Box* program associated with this example:

```
scale 2 2 1 2 2 2 1
root 60
tempo 180
sounds 0 0
monoid add
multi-pattern p1 0 . 0 -2 2 . -1 0 + 0 1 . -2 2 0 . 0
multi-pattern p2 0 2 . -2 0 2 0 . + . 0 -2 0 2 . -2 0
multi-pattern p3 7 -1 7 14 2 14 + 0 -8 0 0 -5 0
mono-colorize c1 %ph_1 p1 %ph_1
mono-colorize c2 %ph_2 p1 %ph_2
mono-colorize c3 %ph_2 p2 %ph_2
mono-colorize c4 %ph_3 p2 %ph_3
colorize c5 %start p3 %ph_1 %ph_1 %ph_2 %ph_2 %ph_3 %ph_3
generate res full 3 %start c1 c2 c3 c4 c5
```

6 Evaluation of the generation algorithms

We have developed a listening study including a questionnaire, intended to gain a clearer understanding of the musical scope of our generation algorithms. We begin by describing the questionnaire, then present its results, and finally provide an interpretation.

6.1 Description of the study

Let us now offer a detailed description of the questionnaire, outline its modalities, and present some pertinent data about its participants.

6.1.1 Practical details, objectives, and participants

The questionnaire was made available for a month starting from December 2023, and was hosted on Google Forms. It was broadcast through several channels, including the *Creative Code Paris* community, the *International Society for Music Information Retrieval* community, the *Society for Mathematics and Computation in Music* community, the *Institut de recherche et coordination acoustique/musique*, and the *musiSorbonne* mailing list. The duration of the questionnaire is approximately ten minutes. After giving their consent for this study, the participants were asked to specify their level of expertise and then their areas of expertise. Subsequently, ten tracks were presented to them for listening.

We have conceived five *Bud Music Box* programs generating some tracks, and selected five works composed by humans. Following the listening of each of these pieces, we ask the following four identical questions to evaluate how they are received:

(Q1) *How would you rate the aesthetic appeal of this track? Range: 0 is "very ugly" and 10 is "very beautiful".*

(Q2) *How would you rate the general complexity of this track? Range: 0 is "extremely simplistic" and 10 is "extremely complex".*

(Q3) *Would you have liked the track to have lasted longer? For instance, 4 or 5 means that listening was a good experience over the proposed duration, but that a longer duration might have been less interesting. Range: 0 is "absolutely not" and 10 is "definitively".*

(Q4) *How likely do you think this track was composed by a human (instead of an algorithm)? Range: 0 is "most likely by an algorithm" and 10 is "most likely by a human".*

In addition, a comment space is provided to collect the comments of each participant for each piece. The scores out of 10 obtained for each track provide us with an indication of their perceived quality according to various aspects. The participants, not knowing which work is composed according to our method or by a human, results in the scores obtained on all five human works serving as a benchmark to evaluate the works generated by our method.

The questionnaire was filled out by 70 participants. Among them, 29 declared having a level of expertise equal to or greater than 8 out of 10. Our analysis of the results is based solely on the opinions of these 29 experts. Upon analyzing the declared areas of expertise, the most frequently occurring fields of expertise are piano practice (nine times), electronic music (six times), and popular music (four times).

6.1.2 Musical tracks

Each offered track lasts 30 seconds and in any case, the sound begins with a gradual increase in the first few seconds, then stabilizes, and diminishes in the last few seconds. In order to standardize the interpretation, each track is played from a MIDI file. Here is the list of the considered tracks:

- T1 Computer-generated, `P_2_4.bmb --seed 8`.
- T2 Human-composed, Erik Satie, *Vexations*, 1893.
- T3 Computer-generated, `H_2_5-2-1-1.bmb --seed 4`.
- T4 Computer-generated, `F_2_6-6-6.bmb --seed 0`.
- T5 Human-composed, Johann Sebastian Bach, *Duetto in E minor, BWV 802*, 1739.
- T6 Computer-generated, `P_2_15-3-1.bmb --seed 5`.
- T7 Human-composed, Radan Papezik, *12-tone blues*, 2006.
- T8 Human-composed, Erik Satie, *Españaña, Croquis et Agaceries d'un gros bonhomme en bois*, 1913.
- T9 Computer-generated, `H_2_3-2-1-1.bmb --seed 3`.
- T10 Human-composed, Johann Sebastian Bach, *Fugue in C minor, BWV 906b*, 1738.

They are presented in this exact order to all participants. This order was selected at random when the questionnaire was designed.

As explained in Section 4, the bud music box algorithms work partly in generating musical phrases through performing some mechanical operations on them, reminiscent of some works by Bach. For this reason, we chose two excerpts T5 and T10 from this composer. Additionally, since some generated phrases sometimes recall passages of minimal music, we have chosen T2 and T8, extracted from works of Satie who is often considered one of the precursors of this trend, although this is certainly debatable. We have also included T7, a more recent and royalty-free composition of Papezik which falls within the serialism movement. Under particular settings, our generation method can produce passages that are close to this style.

Our five computer-generated tracks are based upon small sets of small multi-patterns. For instance, T4 results from the program presented in Section 5.3.2. We made this paradigmatic choice because, in the extreme, with very long multi-patterns taken from existing compositions, our algorithms would generate pieces too close to the original ones. This approach, which prioritizes distinctiveness over replication, is intentional for the scope of our current evaluation, although this stance is open to reconsideration, as will be discussed in Section 6.2.3. Consequently, we aimed to find a middle ground between the simplicity of the multi-patterns and our subjective perception of the outcomes. Additionally, it is important to note that our expertise does not lie in the field of musical composition.

Here is a very important remark: our objective is not to claim that our method can, even remotely, produce pieces comparable to those ones of human composers. Rather, our aim is to have standard works that can be used to detect the quality of responses, and whose metrics can be used as benchmarks to estimate the quality of the computer-generated creations.

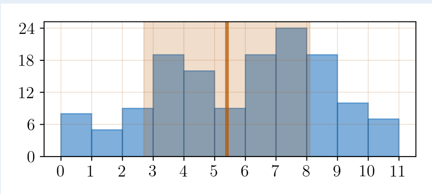
6.2 Results and interpretation

Here we present some significant data collected during the study. We then offer an interpretation of this data before presenting the limitations of the study and a conclusion.

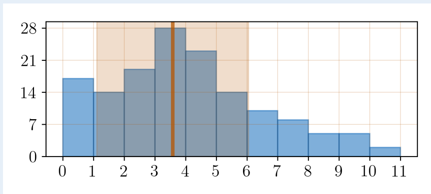
6.2.1 some statistics

The following histograms present some distributions of responses, ranging from 0 to 10, for each question among (Q1), (Q2), (Q3), and (Q4) merged for the human-composed pieces on the left and for the computer-generated tracks on the right. Orange thick lines are positioned on the means of the distributions. Approximations of these values are denoted by “m” under the histograms. Transparent orange rectangles around the means, having widths of twice the standard deviations, are also depicted. Approximations of these values are denoted by “σ” under the histograms.

- Distributions of the aesthetic appeal evaluation:

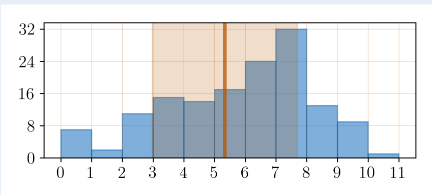


$m \simeq 5.40$, $\sigma \simeq 2.70$

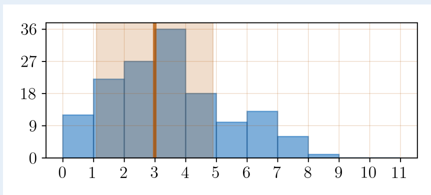


$m \simeq 3.58$, $\sigma \simeq 2.47$

- Distributions of the complexity evaluations:

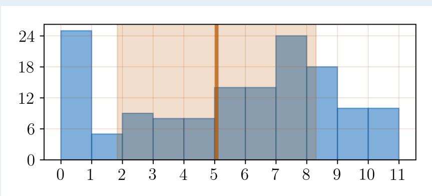


$m \simeq 5.33$, $\sigma \simeq 2.36$

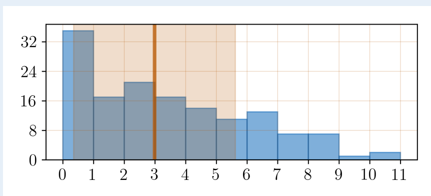


$m \simeq 2.99$, $\sigma \simeq 1.90$

- Distributions of the willingness to listen evaluation:

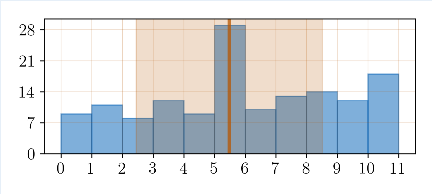


$m \simeq 5.07$, $\sigma \simeq 3.24$

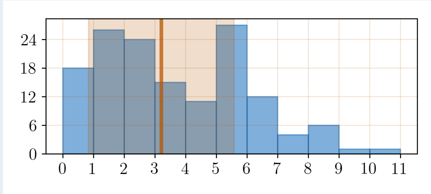


$m \simeq 2.99$, $\sigma \simeq 2.64$

- Distributions of the human composition likelihood evaluation:



$m \simeq 5.48$, $\sigma \simeq 3.03$

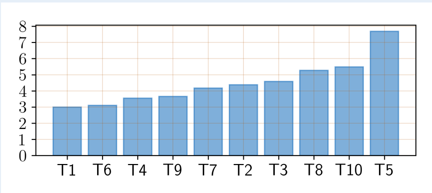


$m \simeq 3.21$, $\sigma \simeq 2.37$

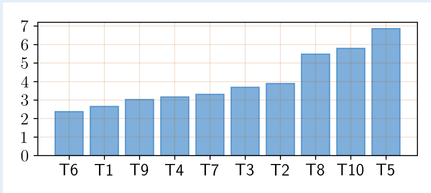
Let us consider the mean absolute error regarding Question (Q4), which asks to classify the tracks according to their origin. By interpreting it as an accuracy rate, this value is about $55\% \simeq 5.48/10$ on human-composed pieces (since the expected correct response is $10/10$) and about $68\% \simeq 1 - 3.21/10$ on computer-generated tracks (since the expected correct response is $0/10$).

The following histograms present the ten tracks ranked w.r.t. the average rating assigned by participants for each question among (Q1), (Q2), (Q3), and (Q4):

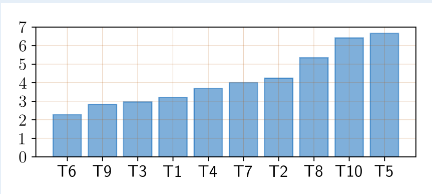
- Aesthetic appeal ranking:



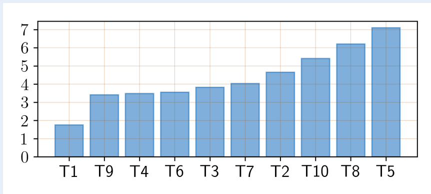
- Willingness to listen ranking:



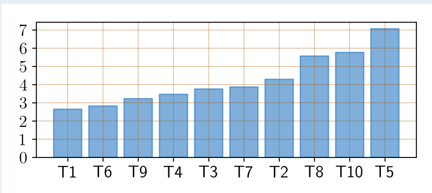
- Complexity ranking:



- Human composition likelihood ranking:



Here is the ranking w.r.t. the means of the sums of the evaluations of Questions (Q1), (Q2), (Q3), and (Q4):



Let us now present a synthesis of the observations of the participants. The total number of comments left for human-composed pieces is 69 (approximately 48% of participants commented on average on each such track) and the total number of comments left for computer-generated tracks is 61 (approximately of 42% participants commented on average on each such track).

In evaluating the five human-composed pieces, participants noted contrasts between simple rhythms and complex harmonies. There was frequent speculation about whether the pieces were human-composed or algorithmically generated, particularly due to their mechanical tones and lack of expressive variation. Certain pieces were appreciated for their clear direction and contrapuntal styles, drawing comparisons to baroque composers like Bach. Opinions remained divided regarding the authenticity of the compositions, with some tracks being praised for their sophistication and others critiqued as generic or excessively mechanical in their emulation of classical styles.

The evaluations of the five tracks generated by our method indicate a general perception of simplicity and predictability in melody and harmonic structure. The mechanical and synthetic quality was often highlighted, with listeners expressing a need for more variety and depth, often finding the compositions too repetitive. The repetitive and predictable nature of some pieces led to speculation about the possibilities for algorithmic composition. However, positive elements such as interesting rhythmic patterns, minimalist aspects, gamelan-like elements, and intriguing intervals were also noted.

6.2.2 Interpretation

Let us provide an interpretation of the previous results.

About classification accuracy The classification accuracy rate on computer-generated tracks suggests that roughly one out of every three tracks generated by the bud music box algorithms is perceived as having a reasonable chance of being composed by a human. Similarly, one out of every two tracks composed by a human is perceived as having been generated by a computer. This relatively high rate might be explained by the choice of the sounds forming the tracks, as discussed further in Section 6.2.3. Therefore, although the classification accuracy is not exceptional, it is encouraging for our method that nearly a third of the tracks generated by the machine could be reasonably perceived as human-composed. This outcome is particularly significant considering the limitations in sound quality.

About the general aesthetic appeal of the tracks We observed that the mean aesthetic appeal for all five computer-generated tracks is about $3.6/10$. Conversely, the aesthetic appeal for all five human-composed tracks is about $5.4/10$. This notable quasi two-points difference demonstrates consistency in the responses since the human-composed tracks served as benchmarks for evaluating the perceived quality of the tracks generated by our algorithms. This relatively modest rating of $5.4/10$ reflects a level of severity and stringency in the responses to Question (Q1). Based on participant comments, this severity can be partly attributed to the quality of the MIDI sounds and automatic interpretations. Nevertheless, the aesthetic appeal of the tracks generated by our method is reasonably good. It was particularly noteworthy to find T3 in the fourth position in the overall rankings of tracks w.r.t. their aesthetic appeals.

About the general ranking of the tracks To derive a general evaluation of a track, we calculated a score based on the mean of the sums of the responses received for Questions (Q1), (Q2), (Q3), and (Q4). This approach is deemed relevant as all four dimensions highlight positive attributes of the tracks. Particularly for the last question, it was considered coherent to assign a value to a piece proportional to the impression it creates of being human-composed. All computer-generated tracks were ranked lower than human-composed pieces. Track T1 ranked the lowest, but notably, T3 was very close to the human-composed track at fifth position. This appears to be influenced by the relatively low perceived complexity in response to Question (Q2). Overall, the best tracks generated by our method are T3 and T4.

6.2.3 Limitations

Let us state some of the limitations of this study.

On the order of the presented tracks A primary concern pertains to the questionnaire design, especially concerning the order in which the tracks were presented. As mentioned above, the order was randomly selected once the questionnaire was designed. However, to more effectively mitigate potential bias arising from the order in which tracks are presented, it would have been preferable to provide each participant with a uniquely randomized order of tracks.

On the identification of expert participants We paid particular attention to the responses from experts in various fields of music, which was crucial to get significant responses. The identification of experts was based on self-assessment of their expertise. This approach, however, introduces at least two significant issues. Firstly, self-assessment is inherently subjective and can be influenced by an individual level of humility or self-confidence. Consequently, it is probable that some individuals with expert-level skills were categorized as non-experts. Additionally, setting a minimum threshold of 8 out of 10 for expert classification is arbitrary. While efforts were made to calibrate responses to this question, the process was not foolproof.

On the sound quality and interpretation of the tracks A significant challenge in this study was the quality of the interpretation of the tracks. As outlined in the questionnaire description, we opted to use MIDI-generated tracks to ensure that the quality of their interpretation did not overshadow their intrinsic compositional qualities. However, as highlighted in numerous participant comments, this decision significantly impacted the reception of the pieces. It is important to note that due to this limitation, the scores obtained from the study should not be interpreted in absolute terms, but rather in relative terms. Despite this constraint, the approach was adequate for the objectives of the current study.

On the musical quality of the computer-generated tracks The tempered reception of the participants towards the tracks generated by our algorithms was previously discussed. However, as highlighted in earlier sections, some characteristics of the computer-generated pieces were better received than certain human-composed works (like for instance for T3 which is ranked ahead of T2 and T7 for Question (Q1), and ahead of T7 for Question (Q3)). Of course, this does not necessarily imply intrinsic value in the algorithmically generated pieces. A preferable approach would have been to offer tracks of significantly higher quality. As mentioned, the pieces produced by our algorithms were constructed from small sets of multi-patterns. An advantageous modification could have involved relaxing this somewhat arbitrary constraint slightly, allowing the inclusion of phrases from works in the styles we aimed to emulate as multi-patterns. This approach, being easily implementable, would have yielded a distinctly different set of pieces.

6.2.4 Conclusion

The objective of this evaluation study was successfully achieved. We garnered very consistent responses from participants, which provided a clearer understanding of the quality of music generated by the bud music box algorithms. As anticipated, the reception of the algorithm-generated pieces was less favorable compared to those composed by humans. However, we now possess a valuable point of comparison, highlighting that certain rhythmic and harmonic elements were effectively captured. Interestingly, the results revealed not only expected elements of minimal music but also surprising instances of gamelan music influences. The primary criticism revolved around the simplicity and repetitiveness of the pieces. To further advance this research, it would be beneficial for a composer to directly engage with our algorithm. By merging their expertise with the capabilities of the algorithm, we could explore high-quality compositions and fully realize the potential of the method.

7 Conclusion and perspectives

In this work, we have introduced the music box model, a framework to represent musical phrases as multi-patterns, perform computations on these, and various random generation algorithms. This framework has some strengths and weaknesses as discussed in Sections 2.2.4, 2.2.5, and 3.3.3. Our generative algorithms, derived from this framework, were also the subject of an evaluation, reported in Section 6. Here are some perspectives raised by this work.

The first one consists in the description of a minimal generating set for the operad P_m^D, $m \geq 2$, even just for the additive degree monoid $D = \mathbb{Z}$. The knowledge of such a minimal generating set would provide a way to decompose a multi-pattern as a syntax tree decorated by generators. This means we can parse a musical phrase into primitive parts. By using some algorithms coming from operad theory, this would lead to applications such as the automatic discovery of repeated parts of musical phrases (up to some elementary transformations like transpositions or more complex ones involving rhythmic motives), or the compression of musical data.

Another perspective is to consider variations of the operad RP controlling the rhythmic part of the multi-patterns. Such a variation can potentially produce very different results from the present ones when used with bud generating systems. In the present work, the operad RP is constructed as the image by the construction $\mathbf{U}$ of the additive monoid on $\mathbb{N}$. It is envisaged to see if it is possible to use other monoids and to what extent this can lead to new ways of composing rhythm patterns.

Acknowledgments We would like to express our gratitude to all the participants who contributed to the study. Special thanks go to Moreno Andreatta, Grégory Châtel, Chiara Giraudo, and Grégory Jarrige for their help in distributing the questionnaire.

References

Ames, C. (1989). The Markov Process as a Compositional Model: A Survey and Tutorial. *Leonardo*, *22*(2), 175–187.

Andreatta, M. (2018). From music to mathematics and backwards: introducing algebra, topology and category theory into computational musicology. In *Imagine math. 6* (pp. 77–88). Springer, Cham.

Bel, B. (1989). Pattern grammars in formal representations of musical structures. *International Joint Conference on Artificial Intelligence, Workshop on AI & Music*, 118–146.

Boardman, J. M., & Vogt, R. M. (1973). Homotopy invariant algebraic structures on topological spaces. *Lect. Notes Math.*, *347*.

Briot, J.-P., Hadjeres, G., & Pachet, F.-D. (2020). *Deep Learning Techniques for Music Generation*. Springer Nature Switzerland AG.

Cambouropoulos, E., Crawford, T., & Iliopoulos, C. S. (2001). Pattern Processing in Melodic Sequences: Challenges, Caveats and Prospects. *Computers and the Humanities*, *35*, 9–21.

Collins, N. (2008). The Analysis of Generative Music Programs. *Organ. Sound*, *13*, 237–248.

Diaz-Jerez, G. (2000). *Algorithmic Music Using Mathematical Models in Music Composition* [Doctoral dissertation, The Manhattan School of Music].

Eibensteiner, L. (2018). Procedural Music Generation with Grammars. *Central European Seminar on Computer Graphics*.

Fernández, J. D., & Vico, F. (2013). AI Methods in Algorithmic Composition: A Comprehensive Survey. *J. Artif. Intell. Res.*, *48*, 513–582.

Garcia Salas, H. A., Gelbukh, A., Calvo, H., & Soria, F. (2011). Automatic Music Composition with Simple Probabilistic Generative Grammars. *Polibits*, *44*, 59–65.

Gartland-Jones, A., & Copley, P. (2003). The suitability of genetic algorithms for musical composition. *Contemp. Music Rev.*, *22*(3), 43–55.

Giraudo, S. (2015). Combinatorial operads from monoids. *J. Algebr. Comb.*, *41*(2), 493–538.

Giraudo, S. (2018). *Nonsymmetric Operads in Combinatorics*. Springer Nature Switzerland AG.

Giraudo, S. (2019). Colored operads, series on colored operads, and combinatorial generating systems. *Discrete Math.*, *342*(6), 1624–1657.

Giraudo, S. (2020). Generation of musical patterns through operads. *Journées d'informatique musicale*.

Giraudo, S. (2023). Bud Music Box [Version `1.101` (November 2023), https://github.com/SamueleGiraudo/Bud-Music-Box]. *GitHub repository*.

Holtzman, S. R. (1981). Using Generative Grammars for Music Composition. *Comput. Music J.*, *5*(1), 51–64.

Hopcroft, J. E., Matwani, R., & Ullman, J. D. (2006). *Introduction to Automata Theory, Languages, and Computation* (3rd). Pearson.

Hudak, P. (2004). An Algebraic Theory of Polymorphic Temporal Media. *Lect. Notes Comput. Sc.*, *3057*.

Hudak, P., & Quick, D. (2018). *The Haskell School of Music: From Signals to Symphonies*. Cambridge University Press.

Jedrzejewski, F. (2019). *Hétérotopies musicales: Modèles mathématiques de la musique*. Hermann.

Komosinski, M., & Szachewicz, P. (2015). Automatic species counterpoint composition by means of the dominance relation. *J. Math. Music*, *9*, 75–94.
Lerdhal, F., & Jackendoff, R. (1996). *A Generative Theory of Tonal Music*. MIT Press.
Lindenmayer, A. (1968). Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs. *J.Theor. Biol.*, *18*(3), 300–315.
Loday, J.-L., & Vallette, B. (2012). *Algebraic Operads* (Vol. 346) [xxiv+634]. Springer.
Loy, G., & Abbott, C. (1985). Programming Languages for Computer Music Synthesis, Performance, and Composition. *ACM Comput. Surv.*, *17*(2), 235–265.
Mason, S., & Saffle, M. (1994). L-Systems, Melodies and Musical Structure. *Leonardo Music J.*, *4*, 31–38.
Matić, D. (2010). A genetic algorithm for composing music. *Yugosl. J. Oper. Res.*, *20*(1), 157–177.
May, J. P. (1972). *The geometry of iterated loop spaces* [Lectures Notes in Mathematics, Vol. 271]. Springer-Verlag, Berlin-New York.
McCormack, J. (1996). Grammar Based Music Composition. *Complex Syst.*, *96*, 320–336.
Méndez, M. A. (2015). *Set operads in combinatorics and computer science*. Springer, Cham.
Morris, R. D. (1987). *Composition With Pitch-Classes: A Theory of Compositional Design*. Yale University Press.
Quick, D., & Hudak, P. (2013). Grammar-based music composition in Haskell. *Workshop on Functional Art, Music, Modeling and Design*, 59–71.
Roads, C. (1979). Grammars as Representations for Music. *Comput. Music J.*, *3*(1), 48–55.
Shapiro, I., & Huber, M. (2021). Markov Chains for Computer Music Generation. *Journal of Humanistic Mathematics*, *11*(2).
Steedman, M. J. (1984). A Generative Grammar for Jazz Chord Sequences. *Music Percept.*, *2*(1), 52–77.
Tymoczko, D. (2011). *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford University Press.
Warrack, G. (1945). Music and Mathematics. *Music Lett.*, *26*(1), 21–27.
Yau, D. (2016). *Colored Operads*. American Mathematical Society.