A Model of Scores as Abstract Syntactic Trees

Gonzalo Romero-García, Carlos Agón, Isabelle Bloch

A Model of Scores as Abstract Syntactic Tree

Gonzalo Romero-García[1][0000−0003−1971−6204], Carlos Agón[1][0000−0001−7926−2537], and Isabelle Bloch[2][0000−0002−6984−1532]

[1] Sorbonne Université, CNRS, IRCAM, STMS, Paris, France
{romero,agonc}@ircam.fr
[2] Sorbonne Université, CNRS, LIP6, Paris, France
isabelle.bloch@sorbonne-universite.fr

Abstract. This paper deals with the structure of a musical piece. The score is modeled as an Abstract Syntactic Tree (AST) to account for the hierarchy of its elements. Formal definitions of harmony, texture and instrumentation are proposed and constitute the main components of the model. Concatenation and parallelization operators are then proposed to combine these components and organize them in a tree structure. This approach is illustrated on some examples.

Keywords: Harmony · Texture · Instrumentation · Abstract Syntactic Tree · Hierarchical score modeling.

1 Introduction

Through centuries, Western classical music has been engraved in scores with staff notation. This way of laying down music is very well suited for sharing music and allows mutual comprehension between the composer and the interpreter. However, staff notation does not account, in general, for the hierarchical nature of music, nor for the structure that underlies a musical piece.

In this article, we propose to model a score as an *Abstract Syntactic Tree* (AST). This way, we can express the structure of a musical piece in a hierarchical way, which has proven to be a very useful way of analyzing and composing music [12,13,14,15,11,5].

To organize a score into a tree has already been proposed, for instance through the MusicXML [8] format that uses XML notation and thus a tree structure. However, this format is a pure translation of the score information, and the hierarchy is mainly structured in bars. In our model, we aim to go beyond the bars decomposition into a more high level one.

Modeling a score as an AST (as in [16]) requires making several key decisions, and two fundamental questions must be answered: What are the most basic materials? And, how can they be combined? In computational terms, these questions translate to: What are the leaves of the tree, and what are the nodes? These questions have profound philosophical implications, as their answers are essentially equivalent to asking: What is music made of, and how is it constructed?

It is a common approach to say that music is made by combining *frequency* with *time*. To this, we add a third component: *timbre*. Using these three axes, we will construct our computational model. More precisely, we will identify three musical notions—*harmony*, *texture*, and *instrumentation*—and provide them with formal mathematical definitions. These elements will be shown to be the essential materials of our model.

To combine these materials, we introduce two fundamental operators: *concatenation* and *parallelization*. These operators are analogous to those proposed in [9,10]. They complete our grammar and allow for the construction of ASTs.

Once we have constructed an AST, we can compile it into various musical representations such as a MIDI file, MusicXML, piano roll, audio file, and others. In this article, we focus on the compilation into a MIDI file. Note that, audio and piano rolls can be generated from the MIDI file.

The strength of our model is as follows: constructing our AST allows for a direct interpretation of the structure and hierarchy of a musical piece and the relationships between its components. Successfully expressing common repertoire pieces as ASTs enables us to assert that we have conducted an analysis of the piece. This has deep implications, since it transforms the analysis problem into the construction of an AST. In this sense, our work is close to the one proposed in [18].

Additionally, our approach offers an alternative to traditional score notation for music composition. While engraving music on staffs is the standard method for composing Western classical music, this process typically requires a learning curve that can span years. By contrast, composing through an AST may be more intuitive for newcomers, especially when presented interactively, as in a Computer Assisted Composition (CAC) tool, like OpenMusic [1,4]. Currently, we offer a Python library named `harmtex` (that can be found in the code that accompanies this article) and an XML Schema for expressing the AST as an XML document.

The article is structured as follows. We first expose the main three materials: harmony, texture and instrumentation. These are introduced in mathematical terms to provide a robust framework that extends beyond mere computational implementation. Then, we define the three operators: tensor contraction, parallelization and concatenation, which complete the grammar necessary for our AST model. Finally, we present a computational implementation in the form of a Python library and XML Schema.

All the code and examples from this article are accessible in the public repository that accompanies this article: https://github.com/Manza12/MCM-2024.

2 Harmony, Texture and Instrumentation

The concepts of *harmony*, *texture*, and *instrumentation* are subject to various interpretations and hold different meanings across different domains. In this article, we provide mathematical definitions for each, tailored specifically to our model yet capturing a significant part of their traditional meaning.

2.1 Harmony

Harmony is a term that is very used in musicology, commonly referring to the distribution and relation between pitches.

In this article, we consider that a *harmony* is simply a list of chords. In our framework, a *chord* is a finite set of pitches. More formally, if we identify the musical pitches (..., C3, C♯3, D3, ..., C4, ...) with the integer numbers[3] denoted by $\mathbb{Z}$, a **chord** is a finite subset of $\mathbb{Z}$, that we denote by $C \in \mathcal{P}_{\mathrm{F}}(\mathbb{Z})$, where $\mathcal{P}_{\mathrm{F}}(\mathbb{Z})$ denotes the finite subsets of $\mathbb{Z}$. For instance, $\{\mathrm{C3}, \mathrm{E3}, \mathrm{G3}\} = \{60, 64, 67\}$ is a C major chord. Notice that, with this definition, the empty set is a chord that corresponds to the absence of pitch and thus to rest, and so is a single note (a chord of cardinal 1), an interval (a chord of cardinal 2), a scale (a chord with varying cardinal, usually from 5 to 8), etc. We denote by $\mathcal{C}$ the set of all chords.

Since a **harmony** is a list of chords, we can formally say that $H \in \mathcal{C}^N$ where N is a finite set of indices, i.e. $|N| < \infty$. In general, we choose $N = \{0, 1, ..., n-1\}$, for some $n \in \mathbb{N}$, and think of H as and ordered list.

This definition of harmony is simple yet powerful; for instance, when we talk about the *harmony* of a piece, me may use the list of chords of the piece[4]. In a similar way, the *harmony* of a jazz standard is the ordered list of chords that forms this standard.

In this work, we use harmonies in a particular way, which is in combination with *textures*. Let us then define textures.

2.2 Texture

The notion of *texture* has a huge amount of interpretations, depending on the domain. In this article, we refer to what is commonly called *symbolic* or *compositional texture* [7], which is rather recent compared to the study of harmony [6].

In this framework, we aim to give a precise definition of symbolic texture; it is the time analogous to harmony: instead of pitches, we use what we call *hits*, which are pairs of onsets and durations; instead of chords, we use *rhythms*, that are sets of hits; and a *texture* is a list of rhythms.

While saying that a texture is simply a list of rhythms might appear as oversimplification, we will in the next section that combining a single texture with different harmonies captures the essence of a symbolic texture paired with a chord progression.

In order to be mathematically precise, we need to choose a model for time and define hits, rhythms and textures. Since our main target is to model scores, we consider that time inside a score is modeled through the use of rational numbers, denoted by $\mathbb{Q}$, where we associate 1 to 𝅝, $\frac{1}{2}$ to 𝅗𝅥, $\frac{1}{4}$ to ♩, and so on[5].

[3] The standard way to do that is to associate each pitch with its corresponding MIDI number, i.e., C3 → 60, C♯3 → 61, D3 → 62, C4 → 72, etc.

[4] The question that naturally arises is how these chords are chosen, but this will be the subject of next sections.

[5] We may model ties by the sum of two rational numbers and tuplets by using different numbers in the denominator.

Then, the duration of a hit can be modeled by a positive rational number. It is a bit more delicate how to model the onset; we say that there is a 0 that acts as origin and then the number associated to the onset is the time (expressed in rationals) elapsed between this origin and the onset. In the case of notes of a piece, the origin would be the start of the piece[6]. With this formalization, a **hit** $h = (o, d)$ is an element of $\mathbb{Q} \times \mathbb{Q}^+$.

Let us move on to the definition of rhythms. We define a **rhythm** R as a finite set of hits, thus an element of $\mathcal{P}_{\mathrm{F}}(\mathbb{Q} \times \mathbb{Q}^+)$. For instance, the rhythm | ♩𝄽 ♫ ♩, where the | symbol accounts for the origin, is associated with the set $\{(0, \frac{1}{4}), (\frac{1}{2}, \frac{1}{8}), (\frac{5}{8}, \frac{1}{8}), (\frac{3}{4}, \frac{1}{4})\}$. We denoted by $\mathcal{R}$ the set of all rhythms.

Similarly to the case of harmony, a **texture** is a list of rhythms, that is and element of $\mathcal{R}^N$, where N is a finite set.

2.3 Instrumentation

Instrumentation has been an important domain of music composition and musicology already developed by Berlioz [3], and is today an essential skill for music composition. As in the case of harmony and texture, we expose it here in a rigorous mathematical way.

Let us consider a set of instruments I. We call a **group** G a finite subset of instruments, i.e. an element of $\mathcal{P}_{\mathrm{F}}(I)$. We call $\mathcal{G}$ the set of all groups[7]. Now, we can define an **instrumentation** O[8] as a list of groups, i.e. $O = (G_n)_{n \in N}$, where N is a finite set and G_n is a group for each n in N.

We now have a formal setting for harmony, texture and instrumentation, that is summarized in Table 1.

Table 1: Main terminology employed through this article for frequency, time and timbre. We consider that N is a finite set, i.e. $|N| < \infty$.

Frequency elements

Term	Notation	Belonging to
pitch	p	$\mathbb{Z}$
chord	C	$\mathcal{C} = \mathcal{P}_{\mathrm{F}}(\mathbb{Z})$
harmony	H	$\mathcal{C}^N$

Time elements

Term	Notation	Belonging to
hit	h	$\mathbb{Q} \times \mathbb{Q}^+$
rhythm	R	$\mathcal{R} = \mathcal{P}_{\mathrm{F}}(\mathbb{Q} \times \mathbb{Q}^+)$
texture	T	$\mathcal{R}^N$

Timbre elements

Term	Notation	Belonging to
instrument	i	I
group	G	$\mathcal{G} = \mathcal{P}_{\mathrm{F}}(I)$
instrumentation	O	$\mathcal{G}^N$

[6] We may even allow for negative numbers for the case of anacrusis.

[7] The dependence on I is omitted to simplify notations.

[8] We use the O for instrumentation since I is already used and it refers to the term *orchestration*, usually taken as synonym.

3 Composition Operators

We have now our three main mathematical objects (harmony, texture and instrumentation) that we are using as leaves of our AST. Let us now expose the three main operators that we are using to combine them: tensor contraction, parallelization and concatenation.

3.1 Tensor contraction

Music in a score is made of *notes*. Each note has a lot of parameters attached to it, but we are keeping four of them: the *pitch*, the *onset*, the *duration* and the *instrument*. This are, as presented before, elements of $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Q}^+$ and I, which makes a note an element of $\mathbb{Z} \times \mathbb{Q} \times \mathbb{Q}^+ \times I$, that we will call the space of notes and denote it by $\mathcal{N}$.

We call the finite power set of $\mathcal{N}$, $\mathcal{P}_{\mathrm{F}}(\mathcal{N})$, the **musical space** and we denote it by $\mathcal{M}$. With these definitions, a musical piece might be defined as an element of $\mathcal{M}$.

The aim of tensor contraction is to merge a harmony, a texture and an instrumentation into a set of notes belonging to $\mathcal{M}$, and thus forming a musical piece (or fragment). Let us present the mathematical formalism.

Let N be a finite set of indices. Let $H = (C_n)_{n\in N} \in \mathcal{C}^N$ be a harmony, $T = (R_n)_{n\in N} \in \mathcal{R}^N$ be a texture and $O = (G_n)_{n\in N} \in \mathcal{G}^N$ be an instrumentation. We define the **tensor contraction**[9] generated by H, T and O, denoted by $H \otimes T \otimes O$ as the union of the Cartesian products between C_n, R_n and G_n, i.e.,

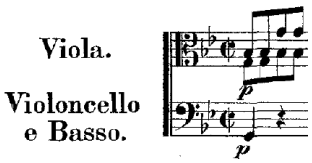
$$H \otimes T \otimes O = \bigcup_{n\in N} C_n \times R_n \times G_n \tag{1}$$

$$= \bigcup_{n\in N} \{(p, o, d, i) \in \mathbb{Z} \times \mathbb{Q} \times \mathbb{Q}^+ \times I : p \in C_n,\, (o, d) \in R_n,\, i \in G_n\}\,. \tag{2}$$

Notice that we require H, T and O to have the same set of indices N, such that we can pair the chords with the rhythms and with the groups.

Let us show how we can use this operator to build music fragments. To this end, we use as example the first 7 measures of Mozart's Symphony No.40 in G minor, K.550, depicted in Figure 1.

Let us focus first in the accompaniment, that is, the pattern



[9] The term *tensor contraction* comes from the similarity of this operation with tensor contraction, where the contraction operator is the union instead of the sum.

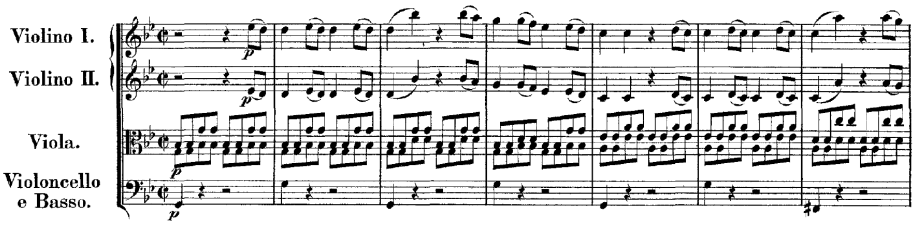


Fig. 1: First 7 measures of Mozart's Symphony No.40 in G minor, K.550.

that can be expressed as the set of notes

$$\begin{aligned} A_0^{\mathrm{a}} = \{&(\mathrm{G2}, 0, \tfrac{1}{4}, \mathbf{Vlc.}), (\mathrm{G2}, 0, \tfrac{1}{4}, \mathbf{Cb.}),\\ &(\mathrm{G3}, 0, \tfrac{1}{8}, \mathbf{Vla.}), (\mathrm{B}\flat 3, 0, \tfrac{1}{8}, \mathbf{Vla.}), (\mathrm{G3}, \tfrac{1}{8}, \tfrac{1}{8}, \mathbf{Vla.}), (\mathrm{B}\flat 3, \tfrac{1}{8}, \tfrac{1}{8}, \mathbf{Vla.}),\\ &(\mathrm{B}\flat 3, \tfrac{2}{8}, \tfrac{1}{8}, \mathbf{Vla.}), (\mathrm{G4}, \tfrac{2}{8}, \tfrac{1}{8}, \mathbf{Vla.}), (\mathrm{B}\flat 3, \tfrac{3}{8}, \tfrac{1}{8}, \mathbf{Vla.}), (\mathrm{G4}, \tfrac{3}{8}, \tfrac{1}{8}, \mathbf{Vla.})\}\,. \end{aligned}$$

We can use the following harmony, texture and instrumentation:

$$\begin{aligned} H_0^{\mathrm{a}} &= \big(\{\mathrm{G2}\}, \{\mathrm{G3}, \mathrm{B}\flat 3\}, \{\mathrm{B}\flat 3, \mathrm{G4}\}\big) = \big(\{43\}, \{55, 58\}, \{58, 67\}\big)\\ T^{\mathrm{a}} &= \big(|\!\!\downarrow, |\!\!\downarrow\!\!\downarrow, |\!\!\downarrow\!\!\downarrow\!\!\downarrow\big) = \big(\{(0, \tfrac{1}{4})\}, \{(0, \tfrac{1}{8}), (\tfrac{1}{8}, \tfrac{1}{8})\}, \{(\tfrac{2}{8}, \tfrac{1}{8}), (\tfrac{3}{8}, \tfrac{1}{8})\}\big)\\ O^{\mathrm{a}} &= \big(\{\mathbf{Vlc.}, \mathbf{Cb.}\}, \{\mathbf{Vla.}\}, \{\mathbf{Vla.}\}\big) \end{aligned}$$

and we have

$$A_0^{\mathrm{a}} = H_0^{\mathrm{a}} \otimes T^{\mathrm{a}} \otimes O^{\mathrm{a}}\,. \tag{3}$$

We can actually express the rest of the accompaniments by only changing the harmony, that is by choosing

$$\begin{aligned} H_1^{\mathrm{a}} &= \big(\emptyset, \{\mathrm{G3}, \mathrm{B}\flat 3\}, \{\mathrm{B}\flat 3, \mathrm{G4}\}\big) & H_2^{\mathrm{a}} &= \big(\{\mathrm{G3}\}, \{\mathrm{G3}, \mathrm{B}\flat 3\}, \{\mathrm{B}\flat 3, \mathrm{G4}\}\big)\\ H_4^{\mathrm{a}} &= \big(\{\mathrm{G2}\}, \{\mathrm{G3}, \mathrm{B}\flat 3\}, \{\mathrm{D4}, \mathrm{G4}\}\big) & H_5^{\mathrm{a}} &= \big(\emptyset, \{\mathrm{G3}, \mathrm{B}\flat 3\}, \{\mathrm{D4}, \mathrm{G4}\}\big)\\ H_6^{\mathrm{a}} &= \big(\{\mathrm{G3}\}, \{\mathrm{G3}, \mathrm{B}\flat 3\}, \{\mathrm{B}\flat 3, \mathrm{D4}\}\big) & H_8^{\mathrm{a}} &= \big(\{\mathrm{G2}\}, \{\mathrm{A3}, \mathrm{E}\flat 4\}, \{\mathrm{E}\flat 4, \mathrm{A4}\}\big)\\ H_9^{\mathrm{a}} &= \big(\emptyset, \{\mathrm{A3}, \mathrm{E}\flat 4\}, \{\mathrm{E}\flat 4, \mathrm{A4}\}\big) & H_{10}^{\mathrm{a}} &= \big(\{\mathrm{G3}\}, \{\mathrm{A3}, \mathrm{E}\flat 4\}, \{\mathrm{E}\flat 4, \mathrm{A4}\}\big)\\ H_{12}^{\mathrm{a}} &= \big(\{\mathrm{F}\sharp 2\}, \{\mathrm{A3}, \mathrm{D4}\}, \{\mathrm{D4}, \mathrm{C5}\}\big) & H_{13}^{\mathrm{a}} &= \big(\emptyset, \{\mathrm{A3}, \mathrm{D4}\}, \{\mathrm{D4}, \mathrm{C5}\}\big) \end{aligned}$$

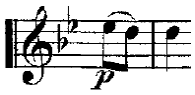
and $H_7^{\mathrm{a}} = H_3^{\mathrm{a}} = H_1^{\mathrm{a}}$, $H_{11}^{\mathrm{a}} = H_9^{\mathrm{a}}$.

This means that each half bar[10] A_i can be expressed as

$$A_i^{\mathrm{a}} = H_i^{\mathrm{a}} \otimes T^{\mathrm{a}} \otimes O^{\mathrm{a}}\,. \tag{4}$$

A similar approach could be used to model the melody; let us consider the main motive:

[10] A_i corresponds to the bar $\lfloor \frac{i}{2} \rfloor + 1$ and to the first half if i is even and to the second half if i is odd.



We can model it through the texture

$$T^{\mathrm{m}} = \left(\text{𝅘𝅥𝅮}\flat\gamma|, \text{𝅘𝅥𝅮}|, |\text{𝅘𝅥}, |\text{𝅘𝅥}\right). \tag{5}$$

The harmonies will depend on the instrument we consider; in order to avoid a too large AST in Figure 3, we consider that both violins are a single instrument[11], **Vln.**, and then we have the instrumentation $O^{\mathrm{m}} = \left(G_{\mathbf{Vln.}}, G_{\mathbf{Vln.}}, G_{\mathbf{Vln.}}, G_{\mathbf{Vln.}}\right)$ with $G_{\mathbf{Vln.}} = \{\mathbf{Vln.}\}$. The corresponding harmonices would be

$$\begin{aligned}
H_0^{\mathrm{m}} &= \left(\{\mathrm{E}\flat 4, \mathrm{E}\flat 5\}, \{\mathrm{D}4, \mathrm{D}5\}, \{\mathrm{D}4, \mathrm{D}5\}, \emptyset\right)\\
H_2^{\mathrm{m}} &= \left(\{\mathrm{E}\flat 4, \mathrm{E}\flat 5\}, \{\mathrm{D}4, \mathrm{D}5\}, \{\mathrm{D}4, \mathrm{D}5\}, \{\mathrm{B}\flat 4, \mathrm{B}\flat 5\}\right)\\
H_3^{\mathrm{m}} &= \left(\{\mathrm{B}\flat 4, \mathrm{B}\flat 5\}, \{\mathrm{A}4, \mathrm{A}5\}, \{\mathrm{G}4, \mathrm{G}5\}, \emptyset\right)\\
H_4^{\mathrm{m}} &= \left(\{\mathrm{G}4, \mathrm{G}5\}, \{\mathrm{F}4, \mathrm{F}5\}, \{\mathrm{E}\flat 4, \mathrm{E}\flat 5\}, \emptyset\right)\\
H_5^{\mathrm{m}} &= \left(\{\mathrm{E}\flat 4, \mathrm{E}\flat 5\}, \{\mathrm{D}4, \mathrm{D}5\}, \{\mathrm{C}4, \mathrm{C}5\}, \{\mathrm{C}4, \mathrm{C}5\}\right)\\
H_6^{\mathrm{m}} &= \left(\{\mathrm{D}4, \mathrm{D}5\}, \{\mathrm{C}4, \mathrm{C}5\}, \{\mathrm{C}4, \mathrm{C}5\}, \emptyset\right)\\
H_8^{\mathrm{m}} &= \left(\{\mathrm{D}4, \mathrm{D}5\}, \{\mathrm{C}4, \mathrm{C}5\}, \{\mathrm{C}4, \mathrm{C}5\}, \{\mathrm{A}4, \mathrm{A}5\}\right)
\end{aligned}$$

and $H_1^{\mathrm{m}} = H_0^{\mathrm{m}}$, $H_7^{\mathrm{m}} = H_6^{\mathrm{m}}$.

We have now a way of combining harmonies, textures and instrumentation through the tensor contraction. Let us now expose how we can combine these tensor contractions to build bigger ones. To that end, we propose two operators: *parallelization* and *concatenation*.

3.2 Parallelization

Parallelization is a way of combining tensor contractions such that they start at the same time. To do that, we align the origins of the textures, represented by the 0. This means that the operation is equivalent to the union. In mathematical terms, let X_1 and X_2 be two tensor contractions. Then, the **parallelization** of them, denoted by $X_1 \parallel X_2$, is the set

$$X_1 \parallel X_2 = X_1 \cup X_2\,. \tag{6}$$

In fact, the parallelization of two tensor contractions is a tensor contraction: indeed, if we have $X_1 = H_1 \otimes T_1 \otimes O_1$ and $X_2 = H_2 \otimes T_2 \otimes O_2$ with set of indices N_1 and N_2 respectively, we can choose $H = (H_1, H_2) := (C_n)_{n\in N_1 \sqcup N_2}$, $T = (T_1, T_2) := (R_n)_{n\in N_1\sqcup N_2}$ and $O = (G_1, G_2) := (G_n)_{n\in N_1\sqcup N_2}$, where $\sqcup$ is

[11] This is only a simplification, the model can handle several instances of an instrument like **Vln. I** and **Vln. II**.

the disjoint union[12] and then we have

$$X_1 \parallel X_2 = H \otimes T \otimes O \tag{7}$$
$$= (H_1, H_2) \otimes (T_1, T_2) \otimes (O_1, O_2) \tag{8}$$
$$= \big(H_1 \otimes T_1 \otimes O_1\big) \cup \big(H_2 \otimes T_2 \otimes O_2\big)\,. \tag{9}$$

Notice that this operators inherits all its properties from union, and in particular it is associative and commutative. The graphic representation of the parallelization is shown in Figure 2b.

3.3 Concatenation

The case of the concatenation requires slightly more work; first, we need to define the endpoint of a texture; we define the **endpoint** of a texture as the maximum of the ends of the hits, i.e., if T is a texture with indices N, its endpoint $e(T)$ is defined by

$$e(T) = \max_{n \in N} \{\max\{o + d \in \mathbb{Q} : (o, d) \in R_n\}\}\,. \tag{10}$$

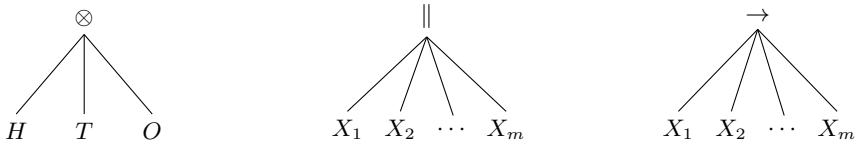
Now, to concatenate two tensor contractions, we shift all the onsets of the second one by the endpoint of the texture of the first one and then we take the parallelization. Mathematically, let X_1 and X_2 be two tensor contractions with $X_j = H_j \otimes T_j \otimes H_j$. The **concatenation** of them, denoted by $X_1 \to X_2$, is the set

$$X_1 \to X_2 = X_1 \parallel \big(H_2 \otimes (T_2 + e(T_1)) \otimes O_2\big) \tag{11}$$

where $T_2 + e(T_1) = (R_n + e(T_n))_{n \in N_2}$ with $R_n + e(T_1) = \{(o + e(T_1), d) \in \mathbb{Q} \times \mathbb{Q}^+ : (o, d) \in R_n\}$.

Similarly as in the case of parallelization, the concatenation of two tensor contractions is also a tensor contraction.

The concatenation is also associative, but it is not commutative. The graphic representation of the concatenation is shown in Figure 2c.



(a) Tensor contraction (b) Parallelization (c) Concatenation

Fig. 2: Graphic representation of the operators of the AST.

[12] We need to use the disjoint union for the case where there are common members between N_1 and N_2.

Let us show how to model the previous example as an AST. We use the harmonies, textures and instrumentations previously introduced. Since the size of the AST grows very fast, we only expose the first three measures.

We chose to split the first bar apart since it is an introduction. We concatenate that with the parallelization of the accompaniment and the melody; the accompaniment is made up from four tensor contractions but the melody has only three repetitions of the motive.

If we wanted to include the subsequent bars, we can map the same distribution of the parallelization simply changing the harmonies.

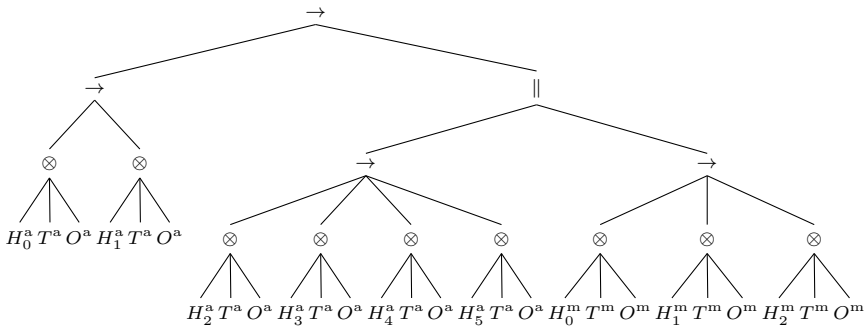


Fig. 3: A proposition of AST of the first three measures of Mozart's Symphony No.40 in G minor, K.550.

4 Computational implementation

In order to compose through this model, we have made an implementation as a Python library that allows us to create harmonies, textures and instrumentations, and use the three operators $\otimes$, $\|$ and $\rightarrow$ to build ASTs.

In addition to this library, we have specified a XML Schema that allows us to write XML files that can be compiled with the Python library into MIDI files. The compilation is done by transforming the AST into a Python object representation, and then by converting this representation into a MIDI file using `pretty_midi` [17].

The interface and data structures used are shown in the following listing; the tensor contraction is an object and the parallelization and concatenation operators are denoted by `|` and `-`, respectively. They are implemented as overridden operator and thus presented with the `__or__` and `__sub__` methods, respectively.

```
Hit: {
    onset: frac
    duration: frac
}
```

```
Rhythm: {
    hits: Set[Hit]
}
Texture: {
    rhythms: List[Rhythm]
}
Pitch: {
    number: int
}
Chord: {
    pitches: Set[Pitch]
}
Harmony: {
    chords: List[Chord]
}
Instrument: {
    name: str
}
Group: {
    instruments: Set[Instrument]
}
Instrumentation: {
    groups: List[Group]
}
TensorContraction: {
    texture: Texture
    harmony: Harmony
    __or__(self, other: TensorContraction) -> TensorContraction
    __sub__(self, other: TensorContraction) -> TensorContraction
}
```

Regarding the XML Schema, we used as root a tag `score` that includes some metadata, and that has an `ast` tag inside. We also allowed in the metadata section the declaration of harmonies, textures and instrumentations, such that they can be reused in the AST through an id.

5 Conclusions and future work

Through this article, we have seen how we can model a score as an Abstract Syntactic Tree (AST). This representation makes the structure of the music explicit and organizes it in a hierarchical way. Structuring musical information in such a way sheds light on the different relations between elements and might be useful both for analysis and composition.

One of the main contributions of this article is the formal definition of key musical notions, namely harmony, texture and instrumentation. While these def-

initions shall not to be taken as the full description of these notions, they capture a significant part of their essence, especially when combined together.

For combining these three elements and building bigger musical structures, we presented three operators; the tensor contraction allows for the mix of harmony, texture and instrumentation, and transforms them into a set of notes, and parallelization and concatenation are the necessary operators to arrange tensor contractions in time.

These six concepts are the elements that constitute the basic grammar of our language and enable the construction of an AST. While the AST can be compiled into a MIDI file, it might serve several other purposes; for instance, we can change some harmonies or textures from the AST for performing harmony or texture transfer. Even more practically, we can change some instrumentations to make arrangements.

We think that one of the most promising way of using this model is for pedagogical purposes; an exercise for a student might be to provide an AST for a music fragment. This can be followed by a transformation of certain elements showing up their impact on the global result.

Lastly, we want to say that, in our opinion, music pieces are essentially conceived as ASTs; while this opinion might seem bold, we think that a lot of composers operate implicitly by choosing textures, harmonies and instrumentations, and aggregating them through tensor contraction, parallelization and concatenation to build their pieces.

Several avenues of research could be explored in order to enrich the expressiveness of the AST. First of all, there are important parameters contained inside a score that we have drop, mainly the dynamic and articulation. It is a question whether we could include them in the texture or rather as a separate component of the contraction. We could also extend the number of operators available in the language: it would be interesting to have a transposition operator that can be applied to harmonies, or even include some other operators like the octave duplication. Another important improvement would be to implement a compiler that transforms the AST into a MusicXML file. Note that we do not use the word compiler as a metaphor but we talk about a real compiler as defined in [2]. This will require to make several choices, for instance the way of engraving the different rhythms of a texture into a single voice. Regarding the interactive part, we aim to implement this model in the Computer Assisted Composition software OpenMusic, as we think that it is well suited to its functional approach and most of the elements are already implemented (like the harmony, that can be mapped to the `ChordSeq` object).

References

1. Agón, C.: OpenMusic : un langage visuel pour la composition musicale assistée par ordinateur. Phd., Paris 6 (1998)
2. Aho, A.V., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques, and Tools. Addison-Wesley, Boston Munich (1986)

3. Berlioz, H.: Grand traité d'instrumentation et d'orchestration modernes. Schonenberger, Paris (1844)
4. Bresson, J., Agon, C., Assayag, G.: OpenMusic: visual programming environment for music composition, analysis and research. In: Proceedings of the 19th ACM international conference on Multimedia. pp. 743–746. New York, NY, USA (2011)
5. Carnovalini, F., Rodà, A., Harley, N., Homer, S.T., Wiggins, G.A.: A New Corpus for Computational Music Research and a Novel Method for Musical Structure Analysis. In: Proceedings of the 16th International Audio Mostly Conference. pp. 264–267. New York, NY, USA (2021)
6. Couturier, L., Bigo, L., Levé, F.: Annotating Symbolic Texture in Piano Music: a Formal Syntax. In: Proceedings of Sound and Music Computing 2022. pp. 577–584 (2022)
7. Couturier, L., Bigo, L., Levé, F.: A Dataset of Symbolic Texture Annotations in Mozart Piano Sonatas. In: Proceedings the 23rd International Society for Music Information Retrieval Conference. pp. 509–516 (2022)
8. Good, M.: MusicXML for notation and analysis. The virtual score: representation, retrieval, restoration **12**(113–124), 160 (2001), publisher: MIT Press, Cambridge, MA
9. Hudak, P.: An algebraic theory of polymorphic temporal media. In: International Symposium on Practical Aspects of Declarative Languages. pp. 1–15. Springer (2004)
10. Hudak, P., Janin, D.: Tiled Polymorphic Temporal Media. In: 2nd ACM SIGPLAN international workshop on Functional art, music, modeling & design (FARM). pp. 49–60. Gothenburg, Sweden (2014)
11. Koelsch, S., Rohrmeier, M., Torrecuso, R., Jentschke, S.: Processing of Hierarchical Syntactic Structure in Music. Proceedings of the National Academy of Sciences **110**(38), 15443–15448 (2013), publisher: Proceedings of the National Academy of Sciences
12. Lerdahl, F.: A Generative Theory of Tonal Music. MIT Press, Cambridge (1983)
13. Lerdahl, F., Jackendoff, R.: An Overview of Hierarchical Structure in Music. Music Perception: An Interdisciplinary Journal **1**(2), 229–252 (1983), publisher: University of California Press
14. Marsden, A., Hirata, K., Tojo, S.: Towards Computable Procedures for Deriving Tree Structures in Music : Context Dependency in GTTM and Schenkerian Theory. In: Proceedings of the Sound and Music Computing Conference 2013. pp. 360–367 (2013)
15. Orio, N., Roda, A.: A Measure of Melodic Similarity based on a Graph Representation of the Music Structure. In: Proceedings of the 10th International Society for Music Information Retrieval Conference. pp. 543–548. Kobe, Japan (2009), pages: 548
16. Orlarey, Y., Fober, D., Letz, S., Bilton, M.: Lambda calculus and music calculi. In: International Conference on Mathematics and Computing (1994)
17. Raffel, C., Ellis, D.P.: Intuitive analysis, creation and manipulation of MIDI data with pretty_midi. In: 15th international society for music information retrieval conference late breaking and demo papers. pp. 84–93 (2014)
18. Tymoczko, D.: Tonality: An Owner's Manual. Oxford Studies in Music Theory, Oxford University Press, Oxford, New York (2023)