Chapter 10
An Algebraic Approach to Time-Span Reduction

Keiji Hirata, Satoshi Tojo, and Masatoshi Hamanaka

Abstract In this chapter, we present an algebraic framework in which a set of simple, intuitive operations applicable to music can be flexibly combined to realize a target application and generate music. We formalize the concept of time-span tree introduced by Lerdahl and Jackendoff (1983) in their *Generative Theory of Tonal Music* (GTTM) and define the distance between time-span trees, on the hypothesis that this might coincide with the psychological resemblance between melodies heard by human listeners. To confirm the feasibility of the proposed framework, we conduct an experiment to determine whether the distance calculated on the basis of the framework reflects cognitive distance in human listeners. To demonstrate that the algebraic framework is computationally tractable, we present the implementation of a musical morphing system that, given two original melodies, generates an intermediate melody at any internally dividing point between them (i.e., at any ratio).

10.1 Introduction

The analogy between music and natural language has long been discussed (Aiello, 1994; Cook, 1994; Jackendoff, 2009; Molino, 2000; Sloboda, 1985). Our short-term memory plays an important role in understanding music as well as language (Baroni et al., 2011). Since short-term memory is used to realize a push-down stack, it can

Keiji Hirata
Future University Hakodate, Hakodate, Hokkaido, Japan
e-mail: hirata@fun.ac.jp

Satoshi Tojo
Japan Advanced Institute of Science and Technology (JAIST), Nomi, Ishikawa, Japan
e-mail: tojo@jaist.ac.jp

Masatoshi Hamanaka
Kyoto University, Clinical Research Center, Kyoto University Hospital, Kyoto, Japan
e-mail: masatosh@kuhp.kyoto-u.ac.jp

accept a context-free grammar (CFG) language. It is commonly accepted that human language is mostly generated by a CFG in Chomsky's hierarchy; at the same time, we often encounter linguistic phenomena that are context-sensitive (Stabler, 2004). Most sentences can be generated by CFGs, which have long distance dependency and a *tree* structure. Thus, we may assume that music is also governed by a CFG-like grammar. Many natural language researchers have tried to implement music parsers with CFG-like grammars (Steedman, 1996; Tojo et al., 2006; Winograd, 1968). For another example of the importance of short-term memory in music, we consider melodic recognition. In a piece of music, the identical *motif* or *phrase* appears repeatedly in time and/or in other voices. When we recognize such a motif/phrase, this suggests that we possess an ability to group consecutive notes or parallel phrases together with the help of short-term memory; this psychological phenomenon is called *Gestalt*.

Influenced by Noam Chomsky's framework of transformational generative grammar (Chomsky, 1957, 1965), Lerdahl and Jackendoff (1983) proposed their *Generative Theory of Tonal Music* (GTTM). GTTM consists of modules for grouping-structure analysis, metrical-structure analysis, time-span reduction, and prolongational reduction. The grouping structure analysis segments a piece of music into nested groups of varying sizes. The metrical structure analysis identifies the positions of strong and weak beats at the levels of a quarter note, half note, whole note, and so on.

The time-span tree is constructed on the basis of the results of the grouping structure and metrical structure analyses in a bottom-up manner: parts come together to form wholes, in accordance with the Gestalt principle. Time-span reduction represents the intuitive idea, originating from Schenkerian analysis, that, if we remove ornamental notes from a long melody, we obtain a simple melody that sounds similar. By time-span reduction, an entire piece of music can eventually be reduced to an important note or a tonic triad. Hence, the time-span tree stands for the progression of this time-span reduction.

Prolongational reduction represents musical intuitions relating to both the harmonic and melodic aspects of the global structure of a piece. In contrast to the time-span reduction, a prolongational tree is constructed in a top-down manner, by recognizing that parts of a piece—or even entire pieces—exhibit patterns of tension and relaxation. That is, given a homophonic (i.e., homorhythmic) sequence, an important note or chord is first selected, and the sequence is then split at that note or chord.

The rules of GTTM comprise well-formedness rules for specifying all the possible tree structures on the basis of analyses, along with preference rules for designating which of the possible tree structures to adopt. As described above, the time-span and prolongational trees represent aspects of the underlying structure of a piece. The theory attempts to look for a unique underlying structure by applying the preference rules. However, a piece can be interpreted in various ways, and, of course, the analysis occasionally derives more than one time-span tree and prolongational tree.

To understand the relationships between GTTM and Chomsky's generative grammar more precisely, let us compare the analysis process of GTTM with the derivation of a sentence using a generative grammar. In Fig. 10.1(a), the meaning of an utter-

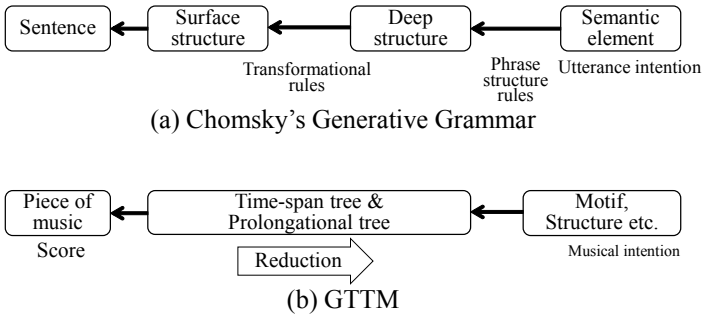


Fig. 10.1 Framework for giving meaning to a sentence and a piece of music

ance is represented by its semantic content, which is transformed into deep and then surface structures by applying the phrase structure rules and the transformational grammar. These grammar rules give meaning to a transformed tree structure. The direction of giving meaning is the same as that of producing a sentence.

The time-span tree and prolongational tree are generated from a motif and a global structure by an elaboration that is the opposite of reduction (Fig. 10.1(b)). The rules and the roles of tree structures in GTTM are different from those they have in linguistic generative grammars. Thus, Lerdahl and Jackendoff (1983, p. 9) state that

> we have found that a generative theory, unlike a generative linguistic theory, must not only assign structural descriptions to a piece, but must also differentiate them along a scale of coherence, weighting them as more or less "preferred" interpretations... The preference rules, which do the major portion of the work of developing analyses within our theory, have no counterpart in linguistic theory; their presence is a prominent difference between the forms of the two theories...

Thus, a generative grammar usually assigns different derivational trees to different sentences, mostly in a one-to-one manner (of course, there are exceptions). Accordingly, in language, the surface structure (a sentence) typically carries enough information to allow direct manipulation and/or calculation of a derivational tree. In contrast, in music, the relationship between the surface structure (a score) and a time-span tree/prolongational tree is more ambiguous due to the preference rules. The time-span/prolongational tree conveys more precise information of musical meaning than the surface structure.

This chapter is structured as follows. In Sect. 10.2, we describe an algebraic framework that formalizes the concept of a time-span tree. We introduce the concepts of reduction and maximal time-span, define the time-span tree operations *join* and *meet*, and provide a theoretical distance between time-span trees. On the basis of this development, in Sect. 10.3, we conduct an experiment to confirm the feasibility of the proposed framework. We compare the cognitive distances of human listeners, measured experimentally, with those calculated by the framework, and determine whether the theoretical distances correctly reflect our cognitive reality. Next, in Sect. 10.4, to illustrate that the combination of primitive operators straightforwardly

realizes a more complicated operation, we implement a musical morphing system. As in the previous section, we employ human listeners to determine whether the morphed melodies generated by the system properly correspond to internally dividing points between the two original melodies given.

10.2 Formal Treatment of Time-Span Trees

In this section, we will explain our approach and introduce some fundamental definitions and properties relating to time-span trees.

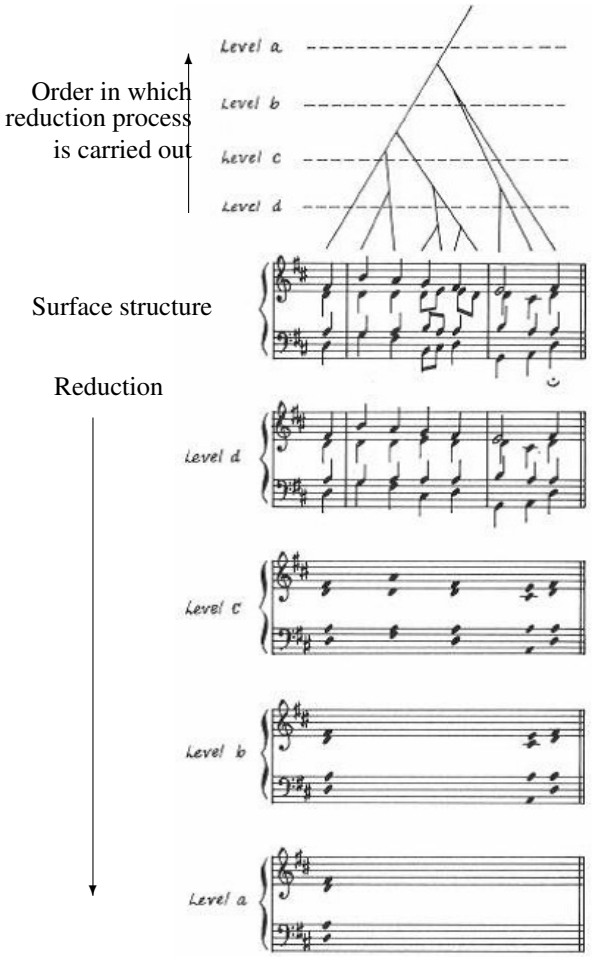


Fig. 10.2 Reduction hierarchy of the chorale, 'O Haupt voll Blut und Wunden' from the St. Matthew Passion by J. S. Bach (from Lerdahl and Jackendoff, 1983, p. 115)

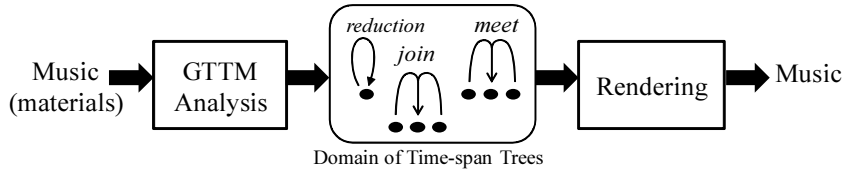


Fig. 10.3 Proposed framework for composition and creation

10.2.1 The Time-Span Tree as a Domain for Modification

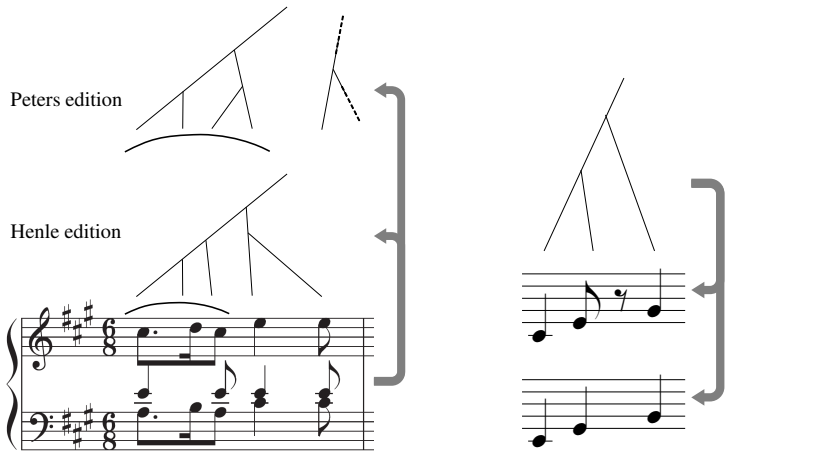
We consider that in computational composition or arrangement, it is more promising to modify the time-span tree than the score itself for the following two reasons. First, the tree is more *meaningful*. The time-span tree is organized on the basis of the *reduction hypothesis* so that neighbouring pitch events[1] are compared in a bottom-up way in terms of importance, and the less important notes are absorbed into more significant ones in a hierarchical manner. As a result, we can obtain the fundamental skeleton of the music (Marsden, 2005). We illustrate this process in Fig. 10.2.[2] We can also use the reduction process to obtain a hypothesis regarding the original intent of the music, on an analogy with a Chomskyan analysis of natural language (Fig. 10.1(b)). This relates to the Schenkerian notion that we can retrieve the underlying structure of a piece by selecting pitch events that represent its tonality (Cadwallader and Gagné, 1998). This selection process exactly corresponds to the reduction hypothesis. In both theories, as each note is classified according to its rhythmic and/or tonal significance, it contributes to the formation of a specific *interpretation* of the music, and, for this reason, we claim that a hierarchical tree is more meaningful than a raw score.

Second, we can distinguish the realm of formal modification (i.e., composition and arrangement) from that of listening. In the former, we need to introduce a *rendering* process which is the reverse of music analysis. In a time-span reduction analysis, a tree is constructed on the basis of the reduction hypothesis; whereas, in the rendering process, a concrete piece of music is created—that is, a musical score is externalized and made performable and audible (Fig. 10.3). Rendering can be viewed as playing the role of resolving ambiguity in the musical surface, which relates to the raw score being less meaningful.

In general, the mapping from musical surfaces to time-span reductions is many-to-many: for a given piece, there is typically more than one possible time-span tree; and for a given time-span tree, there is typically more than one possible surface that has that tree (Marsden et al., 2013). Fig. 10.4(a) shows that two possible time-span

[1] A pitch event originally means a single note or a chord. In this work, we restrict our interest to homophonic analysis as the method of polyphonic recognition is not included in the original theory.

[2] Once a piece of music is reduced, each note with onset-offset and duration becomes a virtual note; it is only meant to be a pitch event that is salient during the corresponding time-span. Therefore, to listen to a reduced piece of music, we need a rendering process that compensates for this onset-offset/duration information.



(a) 1-to-many mapping in time-span reduction analysis (from score to time-span trees)

(b) 1-to-many mapping in rendering (from a time-span tree to scores)

Fig. 10.4 Samples of ambiguity in time-span reduction analysis and rendering. In (a), the time-span reduction depends on slurring, which is different in different editions of the piece. In (b), two different surfaces have the same time-span reduction analysis

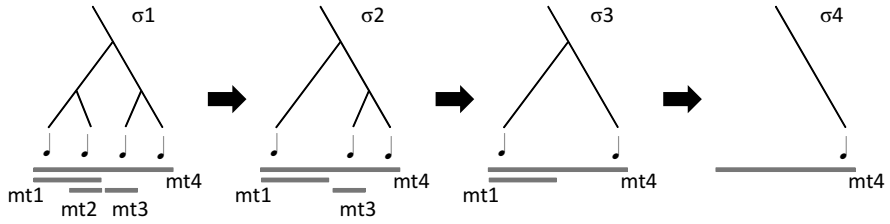
trees may exist, depending on the edition of a score that is used. Conversely, in Fig. 10.4(b), we show that one time-span tree can be rendered in multiple ways, as the time-span tree does not include rests and the occurrences of a rest in a score have various realizations.

10.2.2 Maximal Time-Span

The *head* pitch event of a tree is the most salient event in the tree—i.e., the salient event dominates the whole tree. As the situation is the same in each subtree, we consider that each pitch event has its maximal length of saliency, called its *maximal time-span*. For example, let us think of two maximal time-spans such that one's temporal interval is subsumed by the other's. Since the longer maximal time-span dominates a longer interval, we assume that the longer maximal time-span conveys more information and that the amount of information is proportional to the length of the maximal time-span. Then, we hypothesize that, if a branch with a single pitch event is reduced, the amount of information corresponding to the length of its maximal time-span is lost.

Figure 10.5(a) contains four contiguous pitch events: e1, e2, e3 and e4. Each has its own temporal span (duration on the surface score) denoted by thin lines: s1, s2, s3

(a) Sequence of pitch events and their durations



(b) Reduction proceeds by removing reducible maximal time-spans

Fig. 10.5 Reduction of time-span tree and maximal time-span hierarchy; thick grey lines denote maximal time-spans while thin ones denote pitch durations

and s4. Figure 10.5(b) depicts time-span trees and corresponding maximal time-span hierarchies, denoted by thick grey lines. The relationship between spans in (a) and maximal time-spans in (b) is as follows: at the lowest level in the hierarchy, a span is the same length as a maximal time-span: $\mathrm{mt2} = \mathrm{s2}$, $\mathrm{mt3} = \mathrm{s3}$; at the other levels, $\mathrm{mt1} = \mathrm{s1} + \mathrm{mt2}$, and $\mathrm{mt4} = \mathrm{mt1} + \mathrm{mt3} + \mathrm{s4} = \mathrm{s1} + \mathrm{s2} + \mathrm{s3} + \mathrm{s4}$. In the figure, if the duration of a quarter note is 12 ticks, then $\mathrm{s1} = \mathrm{s2} = \mathrm{s3} = \mathrm{s4} = 12$, $\mathrm{mt2} = \mathrm{mt3} = 12$, $\mathrm{mt1} = 24$, and $\mathrm{mt4} = 48$. That is, every span extends itself by concatenating the span at a higher level along the configuration of a time-span tree. When all subordinate spans are concatenated into one span, the span reaches the maximal time-span.

10.2.3 Lattice and Join/Meet

Here we consider a sequence of reductions from a tree. First, the relation between two trees on the sequence becomes the *subsumption relation*, which is the most fundamental mereological relation among real-world objects in knowledge representation. Since the reduction is generally made in a different order, the sequence bifurcates, and the set of reduced time-span trees becomes a partially ordered set (*poset*).[3] Moreover, if we can define *join* and *meet* in the set, the set becomes a *lattice*.

For the base case, we define *join* and *meet* of two time-spans (Fig. 10.6). If τ_A and τ_B are separated from each other (that is, they do not temporally overlap), *join* does not exist, while *meet* becomes empty, denoted by $\perp$. Next, we consider the inductive case for a time-span tree. Let σ_1 and σ_2 be time-span trees. σ_1 is subsumed by σ_2,

[3] Reflexive, anti-symmetric, and transitive set.

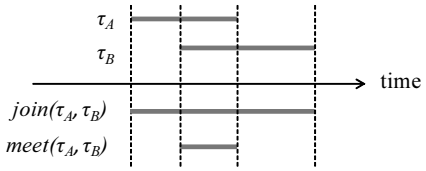


Fig. 10.6 *join* and *meet* operators applied to maximal time-spans

denoted by $\sigma_1 \sqsubseteq \sigma_2$, if and only if for any branch in σ_1 there is a corresponding branch in σ_2.[4] Now let σ_A and σ_B be time-span trees for pieces A and B, respectively.

join: If there is a smallest unique y such that $\sigma_A \sqsubseteq y$ and $\sigma_B \sqsubseteq y$, we call such y the *join* of σ_A and σ_B, denoted by $\sigma_A \sqcup \sigma_B$.
meet: If there is a largest unique x such that $x \sqsubseteq \sigma_A$ and $x \sqsubseteq \sigma_B$, we call such x the *meet* of σ_A and σ_B, denoted by $\sigma_A \sqcap \sigma_B$.

We illustrate *join* and *meet* in a simple example in Fig. 10.7. The '$\sqcup$' (*join*) operation takes eighth notes in the scores to fill sub-time-span trees so that a missing note in one side is complemented. On the other hand, the '$\sqcap$' (*meet*) operation takes $\perp$ for possibly mismatching sub-time-span trees, and thus only the common notes appear as a result.

In the process of unification between σ_A and σ_B, when a single branch is unifiable with a tree, $\sigma_A \sqcup \sigma_B$ chooses the tree while $\sigma_A \sqcap \sigma_B$ chooses the branch recursively. Because there is no alternative action in these procedures, $\sigma_A \sqcup \sigma_B$ and $\sigma_A \sqcap \sigma_B$ exist uniquely. Then, the partially ordered set of time-span trees becomes a lattice, as mentioned above, where $\sigma_A \sqcup x = \sigma_A$ and $\sigma_A \sqcap x = x$ if $x \sqsubseteq \sigma_A$. Moreover, if $\sigma_A \sqsubseteq \sigma_B$, then $x \sqcup \sigma_A \sqsubseteq x \sqcup \sigma_B$ and $x \sqcap \sigma_A \sqsubseteq x \sqcap \sigma_B$ for any x. In an algebraic lattice where

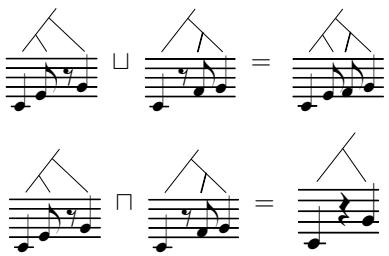


Fig. 10.7 Samples of *join* and *meet*

[4] Currently, we are concentrating on the theory for handling the configurations of trees and the time-spans based on the subsumption relation introduced above, ignoring pitch events. When we become able to define the proper subsumption relation between pitch events and integrate both subsumption relations into a coherent framework, the total theory for handling melodies will be realized. We consider Lerdahl's (2001) tonal pitch space theory to be a valid starting point for developing the subsumption relation between pitch events.

meet and *join* exist uniquely, we can easily confirm the absorption law as follows: $(\sigma_A \sqcup \sigma_B) \sqcap \sigma_A = \sigma_A$ and $(\sigma_A \sqcap \sigma_B) \sqcup \sigma_A = \sigma_A$.

Tojo and Hirata (2012) provided the data representation of a time-span tree in a feature structure and mentioned the algorithms for *join* and *meet*. The framework we propose can be considered algebraic because the set of time-span trees works as a domain and *join* and *meet* are operators defined on this set. Moreover, we consider this algebraic approach to be an implementation of Donald Norman's (1999, p. 67) design principle of 'Simplicity':

> *Simplicity: The complexity of the information appliance is that of the task, not the tool. The technology is invisible.*

That is, Norman argued that a user should be provided with a framework in which a set of simple, intuitive primitives can be flexibly combined to realize an intended function.

10.2.4 Reduction Distance

We call a sequence of reductions of a piece of music a *reduction path*. We regard the sum of the lengths of maximal time-spans lost in going from one tree to another in the reduction path as the distance between the two trees. We generalize the notion to be applicable not only between trees in the same reduction path, but also in any direction in the lattice. We presuppose that branches are reduced only one-by-one, for convenience in summing up distances. A branch is *reducible* only in the bottom-up direction, i.e., a reducible branch possesses no other sub-branches except a single pitch event as a leaf of a tree.

Let $\varsigma(\sigma)$ be a set of pitch events in a time-span tree σ and let $\#\varsigma(\sigma)$ be its cardinality. We denote by s_e the maximal time-span of event e. The distance $d_{\sqsubseteq}$ of two time-span trees such that $\sigma_A \sqsubseteq \sigma_B$ in a reduction path is defined as follows

$$d_{\sqsubseteq}(\sigma_A, \sigma_B) = \sum_{e \in \varsigma(\sigma_B) \setminus \varsigma(\sigma_A)} s_e.$$

For example in Fig. 10.5(b), the distance between $\sigma 1$ and $\sigma 4$ becomes mt1 + mt2 + mt3. Note that, if e3 is first reduced and e2 is subsequently reduced, the distance is the same. Although the distance appears at a glance to be a simple summation of maximal time-spans, there is a latent order in the addition, because the reducible branches are different in each reduction step. To give a constructive procedure to this summation, we introduce the notion of total sum of maximal time-spans as:

$$tmts(\sigma) = \sum_{e \in \varsigma(\sigma)} s_e \quad ,$$

which we call the *total maximal time-span*. When $\sigma_A \sqsubseteq \sigma_B$, $d_{\sqsubseteq}(\sigma_A, \sigma_B) = tmts(\sigma_B) - tmts(\sigma_A)$. As a special case of the above, $d_{\sqsubseteq}(\bot, \sigma) = tmts(\sigma)$.

We now consider the requirements for the distance between two trees to be a true metric. As there is a reduction path between $\sigma_A \sqcap \sigma_B$ and $\sigma_A \sqcup \sigma_B$, it follows that

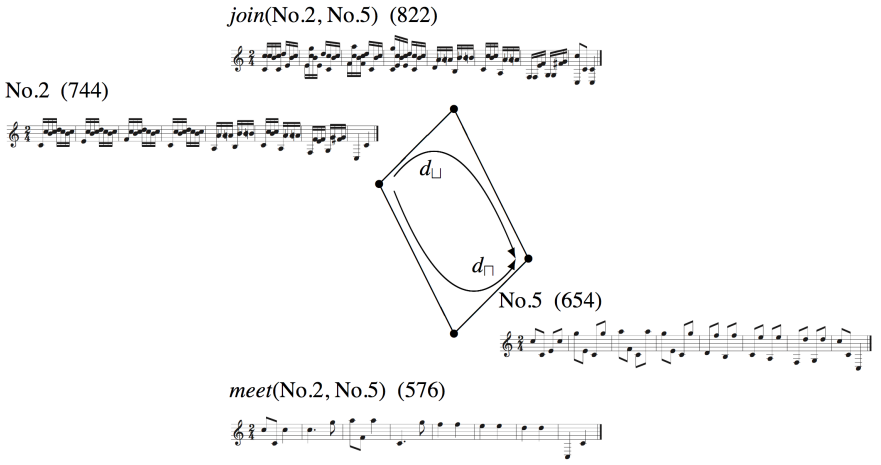


Fig. 10.8 Parallelogram composed of variations No. 2 and No. 5, *join* and *meet*. The values in parentheses are total maximal time-spans

$\sigma_A \sqcap \sigma_B \sqsubseteq \sigma_A \sqcup \sigma_B$ and that $d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_A \sqcup \sigma_B)$ is unique. Suppose we define the following two distance metrics:

$$d_{\sqcap}(\sigma_A, \sigma_B) \equiv d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_A) + d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_B) \ ,$$
$$d_{\sqcup}(\sigma_A, \sigma_B) \equiv d_{\sqsubseteq}(\sigma_A, \sigma_A \sqcup \sigma_B) + d_{\sqsubseteq}(\sigma_B, \sigma_A \sqcup \sigma_B) \ .$$

We immediately obtain the lemma, $d_{\sqcup}(\sigma_A, \sigma_B) = d_{\sqcap}(\sigma_A, \sigma_B)$, by the uniqueness of reduction distance (see Tojo and Hirata (2012) for the outline of a proof). From here on, we therefore omit $\{\sqcap, \sqcup\}$ from $d_{\{\sqcap,\sqcup\}}$, and simply express it as 'd'. Here, $d(\sigma_A, \sigma_B)$ is unique among the shortest paths between σ_A and σ_B. Finally, we obtain the triangle inequality:

$$d(\sigma_A, \sigma_B) + d(\sigma_B, \sigma_C) \geq d(\sigma_A, \sigma_C) \ .$$

For more details on the theoretical background, see Tojo and Hirata (2012).

We show an example in which, given two pieces, the *join* and *meet* are calculated (Fig. 10.8). The two pieces are taken from Mozart's variations K.265/300e '*Ah, vous dirai-je, maman*', variations No. 2 and No. 5. The value in parentheses shows the total maximal time-span of each time-span tree, as defined above. In Fig. 10.8, if we let the duration of a quarter note be 12 ticks, the total maximal time-span of variation No. 2 amounts to 744 ticks, which is the sum of the maximal time-spans of all notes contained in variation No. 2. Similarly, the total maximal time-span of variation No. 5 is 654 ticks. According to the definition of distance, we obtain $d_{\sqcap} = (744 - 576) + (654 - 576) = 246$, and $d_{\sqcup} = (822 - 744) + (822 - 654) = 246$. Notice that the four time-span trees form a parallelogram because the lengths of the opposite sides are equal. Then, we have confirmed the lemma on uniqueness of reduction distance in the proposed framework.

In general, *join* and *meet* of the time-span trees in Fig. 10.8 are possible as long as the left-/right-branching coincides in every subtree. However, we have enhanced the algorithm to tolerate the matching between two different directions of branching. In the current implementation, the *join* and *meet* operations have already been improved to handle unmatched-branching trees so that they preserve the results of *join* and *meet* in the matched-branching trees and satisfy the absorption law, $(\sigma_A \sqcup \sigma_B) \sqcap \sigma_A = \sigma_A$ and $(\sigma_A \sqcap \sigma_B) \sqcup \sigma_A = \sigma_A$, and the lemma, $d_{\sqcup}(\sigma_A, \sigma_B) = d_{\sqcap}(\sigma_A, \sigma_B)$, even for the unmatched-branching trees. For more details, see Hirata et al. (2014).

As described in footnote 4, since the subsumption relation between pitch events is not given, at present, the distance between pitch events is not calculated. Thus, we suppose that every pitch event occurring in a time-span tree is identical. Therefore, within the calculation of the distance between time-span trees, *join* and *meet* neither generate a homophonic pitch event nor a chord; that is, let e be such a pitch event, so we have $\mathit{join}(e,e) = \mathit{meet}(e,e) = e$.

10.3 Verification: Distance and Cognitive Similarity

In this section, we investigate whether the definition of distance correctly reflects cognitive reality. For this purpose, we employ human listeners to compare the distance with intuitive similarity.

The target set of pieces was Mozart's variations K.265/300e '*Ah, vous dirai-je, maman*' (known in English as 'Twinkle, twinkle little star'). The piece consists of the theme and 12 variations. In our experiment, we used the first eight bars of the theme and each variation (Fig. 10.9). Although the original piece includes multiple voices, our framework can only handle monophony; therefore, the original pieces were reduced to monophonic melodies. We did this by extracting salient pitch events from each of two voices, choosing a prominent note from a chord, and disregarding the difference in octave so that the resultant melody sounded fluid. In total, we used 8-bar excerpts from the theme and 12 variations and thus obtained 78 pairs to be compared (${}_{13}C_2 = 78$).

For the similarity assessment by human listeners, 11 university students participated in our study, seven of whom had some experience in playing musical instruments. Each participant listened to all pairs of excerpts, $\langle m_1, m_2 \rangle$, in a random order without duplication, and ranked each pair for similarity on a 5-point scale, ranging from -2 (very different) through to 2 (very similar). To counteract a potential cold start bias, each participant first heard all 8-bar excerpts without ranking them. To avoid order effects, each pair of excerpts was presented in both possible orders on separate trials The average rankings were calculated for each participant and then for all participants. Finally, we computed a distance matrix based on the participants' responses.

For the theoretical estimation by the proposed theory, we used the reduction distance introduced in Sect. 10.2.4. In order to calculate the reduction distance, a unit of duration must be defined. We set this unit to be one-third of a sixteenth note so

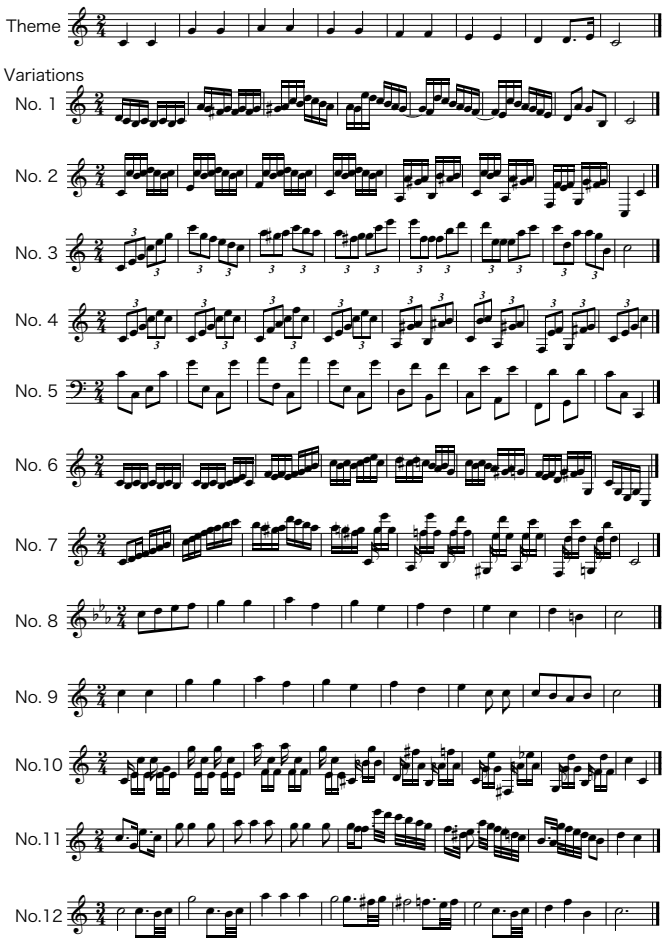
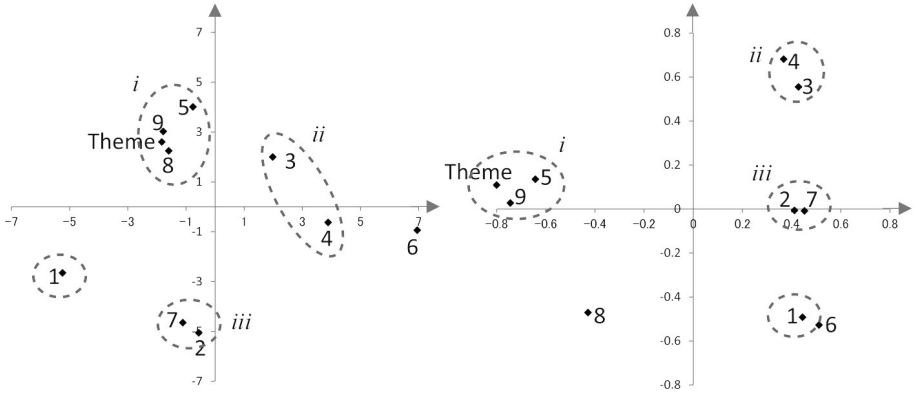


Fig. 10.9 Monophonic melodies used in the experiment

that pieces in both duple and triple time could be represented (this is the same unit as used in the examples in Figs. 10.5 and 10.8). The correct time-span trees of the theme and 12 variations were first created by the authors and cross-checked against each other.

It was not easy to examine the correspondence between the results calculated by $d(\sigma_A, \sigma_B)$ and the psychological resemblance obtained by participants in the distance matrix. We thus employed multidimensional scaling (MDS) to visualize the correspondence. MDS takes a distance matrix containing dissimilarity values or distances among items, identifies the axes to discriminate items most prominently, and plots items on the coordinate system of these axes. Therefore, the more similar items are, the closer together they are in the resulting MDS solution.



(a) Theoretical estimation (b) Human listener

Fig. 10.10 Relative distances among melodies in multidimensional scaling

First, we used Torgerson's (1952) traditional method of scaling in MDS to plot the proximity among the 13 melodies. However, it was still difficult to find a clear correspondence between the results calculated by the reduction distance and the psychological resemblance obtained by participants. We then removed the results for variations 10–12 (Fig. 10.10). The contributions in MDS are as follows: (a) Theoretical estimation: the first axis (horizontal) = 0.21 and the second = 0.17; (b) Human listeners: the first axis (horizontal) = 0.32 and the second = 0.17.

In Fig. 10.10, we can see an interesting correspondence between (a) and (b) in terms of positional relationships among the 10 melodies. In both (a) and (b), we find that the Theme and variations 5 and 9 are clustered together (cluster *i*), that variations 3 and 4 form a cluster (*ii*) and that variations 2 and 7 form a cluster (*iii*). The positional relationships among clusters *i*, *ii* and *iii* resemble each other. The positional relationships between variation 1 and the others in (a) and (b) (except for variation 6) show a similar tendency. Since the contribution in the first axis of (a) is considered close to the second, by rotating the axes of (a) by 90 degrees anticlockwise, a more intuitive correspondence between (a) and (b) emerges. On the other hand, the discrepancy between them is quite apparent; the positional relationship between No.6 and the others is significantly different.

These results suggest a correspondence between our calculated reduction distance and intuitive similarity, if we focus on the rhythmic structure (Hirata et al., 2013). However, in order to claim that our methodology was adequate, we would need to include pitch similarity (see footnote 4). In addition, we need to carry out further comparisons with other distance measures, such as Levenshtein (edit) distance and Earth mover's distance.

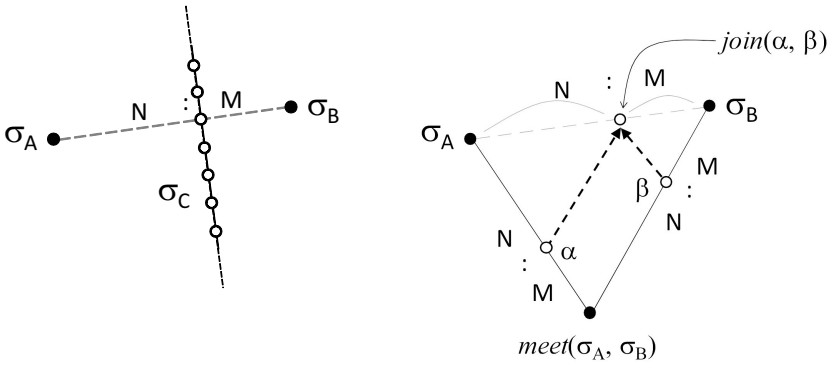


Fig. 10.11 There are infinitely many σ_Cs (left). On the right, the proposed morphing algorithm

10.4 Application: Melodic Morphing

In image processing, a morphing algorithm takes two images and finds an intermediate image. In a similar way, we now propose a new method for composing an intermediate piece of music, given two existing variations with a common theme. Let σ_A and σ_B be two time-span trees of music, and σ_C be the expected result of morphing; we require σ_C to reside at a point between σ_A and σ_B that internally divides the distance between these two in the ratio $M : N$, calculated in terms of the total sum of maximal time-spans (denoted as *tmts* in Sect. 10.2.4). Note that there are uncountably many σ_Cs such that the ratio of the distance between σ_A and σ_C to that between σ_C and σ_B is $M : N$. This is because σ_C resides at any point on the straight line that crosses at such an internally dividing point of $M : N$ and forms an angle of 90 degree with the line segment between σ_A and σ_B (the left-hand side of Fig. 10.11). Thus, we should restrict σ_C to the one that resides on the line segment between σ_A and σ_B, respectively.

Our morphing algorithm, shown on the right-hand side of Fig. 10.11 (Hirata et al., 2014), consists of the following steps:

1. Given the time-span trees of two melodies σ_A and σ_B, calculate *meet*(σ_A, σ_B).
2. Find a time-span tree α that divides the line between σ_A and *meet*(σ_A, σ_B) in the ratio of $N : M$ by removing pitch events in order from σ_A.
3. Similarly, find β that divides the line between σ_B and *meet*(σ_A, σ_B) in the ratio of $M : N$.
4. Calculate *join*(α, β).
5. Obtain a real piece of music by rendering the result of *join*(α, β).

We see that the four time-span trees, $\{\alpha, \beta, meet(\sigma_A, \sigma_B), join(\alpha, \beta)\}$, form a parallelogram, as in Fig. 10.8. Clearly, in terms of the distance between σ_A and σ_B,

we have $d(\sigma_A, \sigma_B) = d(\sigma_A, join(\alpha, \beta)) + d(join(\alpha, \beta), \sigma_B)$. Moreover, $tmts(\sigma_A) \leq tmts(join(\sigma_A, \sigma_B)) \leq tmts(\sigma_B)$ holds if $tmts(\sigma_A) \leq tmts(\sigma_B)$.[5]

Here, we add two more comments on the morphing algorithm. The first concerns the unmatched-branching in *join*, i.e., the unification of left- and right-branching trees. In the current implementation, we interpret the value of *join* as the superimposition of the differently branching nodes of two time-span trees. Thus, the result of *join* simply becomes a chord of two notes sounding simultaneously. Otherwise, for instance, it could be rendered as a transformation of the superimposed time-spans.[6]

The second issue concerns the rendering algorithm itself. The current rendering algorithm works in a top-down manner so that a maximal time-span is basically regarded as a horizontal line segment in a piano-roll representation, and the time-spans at lower levels (closer to the leaves) overwrite those at higher levels. Thus, the entirety of the maximal time-span may be overwritten by the lower-level time-spans; that is, even though a pitch event is quite salient, it may become inaudible, or its assigned duration in a real score may become very short. Consequently, there are cases where the simple top-down algorithm does not generate a proper melody. Thus, we are considering algorithms that, for example, integrate some bottom-up process with the current top-down one; alternatively, we may employ a new process, based on GTTM, for determining whether the rendering process generates a correct melody.

The morphing algorithm is implemented in SWI-Prolog (SWI, 1987). The target set of pieces was again Mozart's variations K.265/300e '*Ah, vous dirai-je, maman*'. In this experiment, we took the first 8 bars of each of the variations 1, 2, and 5 as the sources for morphing (Fig. 10.12). We have chosen these three variations because, for every pair of these three variations, we can calculate *join*—that is, the maximal time-spans are all correctly concatenated. The morphed melodies are shown in Fig. 10.12 between the scores of the variations. For example, "No.2&No5 1:1" means the morphed melody at the midpoint of variations 2 and 5. Ratio "1:3" indicates the position of the internally dividing point, e.g., "No.2&No5 1:3" means the internally dividing point is closer to variation 2 than it is to variation 5. Thus, the bottom three melodies in Fig. 10.12 are formed by morphing variations 1 and 5, with different ratios of internal division. We see that the melodic patterns are gradually changed in accordance with the ratio in distance.

Next, taking "No.2&No.5 1:1" as an example, we examine the morphing calculation in more detail. For convenience of explanation, we show only the first bars of variations 2 and 5 and intermediate time-span trees α and β (Fig. 10.13). In the figure, the intermediate time-span trees are shown in the rendered melodies. Time-span tree α is generated by dividing σ_A (variation 2) and $meet(\sigma_A, \sigma_B)$ in the ratio of 1 : 1. Then, α is generated by removing some reducible branches in σ_A one-by-one so that $tmts(\alpha) = (tmts(\sigma_A) + tmts(meet(\sigma_A, \sigma_B)))/2$. This condition means that $tmts(\alpha)$ is positioned at the centre of $tmts(\sigma_A)$ and $tmts(meet(\sigma_A, \sigma_B))$. Similarly, β is generated by removing some reducible pitch events in σ_B. Finally, by *join*ing α and β, we obtain the first bar of "No.2&No.5 1:1" in Fig. 10.12.

---

[5] *tmts* means total maximal time-span, as introduced in Sect. 10.2.4.

[6] This resembles the notion of a transformation head (Lerdahl and Jackendoff, 1983, p. 155).

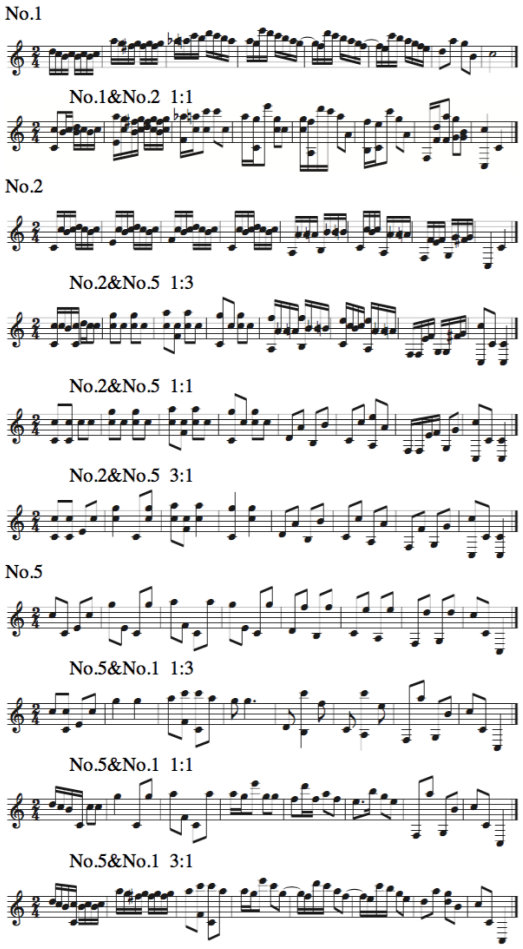


Fig. 10.12 Variations 1, 2, and 5, and morphed melodies between them

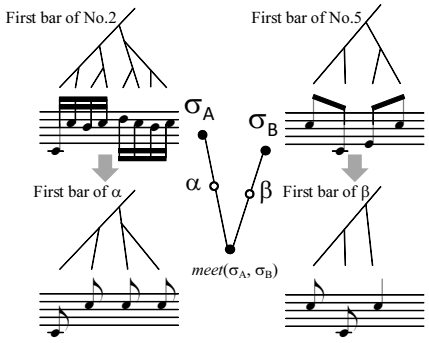


Fig. 10.13 Detailed morphing calculation of first bars of No.2&No.5 1:1

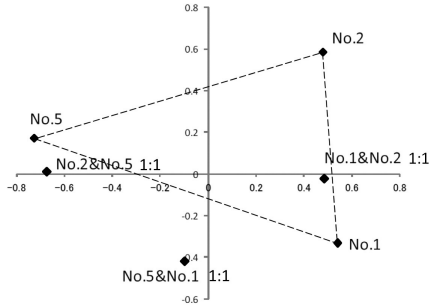


Fig. 10.14 Relative distance among variations and morphed melodies according to the impression of human listeners

For the similarity assessment of the morphed melodies by human listeners, six university students participated in our study, four of whom had played musical instruments for five years or more. We used the same experimental method as described in Sect. 10.3. A participant listened to all pairs $\langle m_1, m_2 \rangle$ in a random order without duplication, where m_i ($i \in \{1,2\}$) were either variations 1, 2 or 5 or the morphed melodies between them, such as "No.2&No.5 M: N". The experimental results were used to construct a distance matrix between these three variations and the morphed melodies between them. We then visualized the results using multidimensional scaling (MDS) (Fig. 10.14).

As can be seen in Fig. 10.14, for variation pairs, $\langle 1,2 \rangle$ and $\langle 1,5 \rangle$, the morphed melodies lie near the midpoints between the original variations, as expected. On the other hand, the position of "No.2&No.5 1:1" is problematic. As can be seen in Fig. 10.12, the number of notes in "No.2&No.5 1:1", which is supposed to be at the midpoint between variations 2 and 5, seems to be the average of the numbers of notes in variations 2 and 5. However, "No.2&No.5 1:1" is almost entirely made of eighth notes, and, as a result, many notes co-occur temporally. This may help to explain why this intermediate melody was perceived by participants as being more similar to variation 5 than variation 2.

10.5 Conclusions

In this chapter, we began by focusing on the structural information provided by a time-span reduction tree produced in accordance with Lerdahl and Jackendoff's (1983) *Generative Theory of Tonal Music* (GTTM), where the process of reduction reflects the hierarchical abstraction of the music. Then, we introduced the concept of *maximal time-span* and formalized the time-span tree; thus, as the subsumption relation exists between trees, the set of trees is a partially ordered set (*poset*) and is qualified as a domain for computational processing and modification. Next, we defined such primitive operations as *join*/*meet* on this domain, thus generating a *distributive lattice*

from this poset. We are then able to define more complicated algebraic operations, combining *join/meet* operations. As we can numerically measure the length of a maximal time-span, we can define the notion of a distance between trees on a reduction path as being the sum of reduced time-spans. Extending this idea, we were able to define the distance between any two arbitrarily chosen trees in the lattice.

To assess the feasibility of the proposed framework, we conducted two experiments. In the first, we focused on the similarity between variations, and compared the reduction distance of our framework with the psychological distance of human intuition. As we discussed in Sect. 10.3, we found a correspondence between the computed reduction distance and experimentally determined intuitions of similarity, when we focused on rhythmic structure. However, further experiments need to be carried out on distance measures that take pitch structure into account, and these measures need to be compared with other metrics that have been proposed in the literature. Next, we implemented a music morphing system in order to illustrate that a combination of primitive operators realizes a more complicated operation. Since the distance between time-span trees defined in our framework satisfies the proper geometric properties, we could locate the internally dividing point on a line segment with a simple ratio. We also found that such geometric positioning coincides to some extent with the cognitive intuition of human listeners.

In order to develop and deploy our proposed framework, we need to consider the following issues. The first concerns *music rendering*, which is the process of realizing a musical score from a time-span tree as we discussed in Sect. 10.2.1. In fact, the applicability of our framework seems to depend strongly on the quality of the rendering. There are many possible algorithms for the rendering process besides the one described in Sect. 10.4. Ideally, a rendering algorithm would restore the original pitch and duration of each note, since the algorithm can be viewed as the reverse of the analysis process shown in Fig. 10.3. However, this is rarely the case in practice. One practical strategy might be to employ machine learning on a large database of pieces paired with their time-span trees. Here, we considered the "round-trip" scenario in which a time-span tree, obtained by carrying out a time-span reduction analysis, can be rendered as a real score which can then be re-analysed to generate a time-span tree. Conversely, we could first have considered the process of generating a tree from a musical surface, and then rendering the tree again to produce a (possibly different) piece of real music. In this way, we would be able to assess the fidelity of the analysis and rendering processes.

Thus far, we have provided only *meet* and *join* as primitive operators and shown an example of music morphing by the combination of these operations. Indeed, if we can extend the notion of such simple arithmetic operations in the domain of time-span trees, we will be able to benefit from richer music manipulation systems. An even more expressive algebra might be achieved by introducing a complement or inverse element to make the set a *group*. As *join* behaves intuitively as addition and *meet* as multiplication, introducing a complement could enhance the algebra by allowing operations analogous to subtraction and division. As the current lattice we have obtained is distributive, in our future work, we intend to employ the *relative pseudo-complement* for each tree and apply it to a new arithmetic operation in a

pseudo-Boolean (Heyting) algebra. This would provide us with much more expressive methods for arranging time-span trees.

Although we have selected time-span trees as the semantic domain in our framework, there are other possibilities. For example, we could incorporate the concept of reduction into the implication–realization theory (Narmour, 1990). This is another direction that we intend to explore in our future work.

Acknowledgements The authors thank the anonymous reviewers for their valuable and essential comments and suggestions, which greatly contributed to improving the quality of this chapter. This work was supported by JSPS KAKENHI Grant Numbers 20300035, 23500145, 25330434 and 26280089.

References

Aiello, R. (1994). Music and language: Parallels and contrasts. In Aiello, R. and Sloboda, J., editors, *Musical Perceptions*, pages 40–63. Oxford University Press.

Baroni, M., Dalmonte, R., and Caterina, R. (2011). Salience of melodic tones in short-term memory: Dependence on phrasing, metre, duration, register tonal hierarchy. In Deliège, I. and Davidson, J., editors, *Music and the Mind: Essays in honour of John Sloboda*, pages 139–160. Oxford University Press.

Cadwallader, A. and Gagné, D. (1998). *Analysis of Tonal Music: A Schenkerian Approach*. Oxford University Press.

Chomsky, N. (1957). *Syntactic Structures*. Mouton de Gruyter.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Cook, N. (1994). Perception—A perspective from music theory. In Aiello, R. and Sloboda, J., editors, *Musical Perceptions*, pages 64–95. Oxford University Press.

Hirata, K., Tojo, S., and Hamanaka, M. (2013). Cognitive similarity grounded by tree distance from the analysis of K.265/300e. In Aramaki, M., e. a., editor, *Music and Motion: Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research (CMMR 2013)*, volume 8905 of *Lecture Notes in Computer Science*, pages 589–605, Marseille, France.

Hirata, K., Tojo, S., and Hamanaka, M. (2014). Algebraic Mozart by tree synthesis. In *Proceedings of the Joint 40th International Computer Music Conference and 11th Sound and Music Computing Conference (ICMC/SMC 2014)*, pages 991–997, Athens, Greece.

Jackendoff, R. (2009). Parallels and nonparallels between language and music. *Music Perception*, 26(3):195–204.

Lerdahl, F. (2001). *Tonal Pitch Space*. Oxford University Press.

Lerdahl, F. and Jackendoff, R. S. (1983). *A Generative Theory of Tonal Music*. MIT Press.

Marsden, A. (2005). Generative structural representation of tonal music. *Journal of New Music Research*, 34(4):409–428.

Marsden, A., Hirata, K., and Tojo, S. (2013). Towards computable procedures for deriving tree structures in music: Context dependency in GTTM and Schenkerian theory. In *Proceedings of the 10th Sound and Music Computing Conference (SMC 2013)*, pages 360–367, Stockholm, Sweden.

Molino, J. (2000). Toward an evolutionary theory of music and language. In Wallin, N. L., Merker, B., and Brown, S., editors, *The Origins of Music*, pages 165–176. MIT Press.

Narmour, E. (1990). *The Analysis and Cognition of Basic Melodic Structure: The Implication–Realization Model*. University of Chicago Press.

Norman, D. (1999). *The Invisible Computer*. MIT Press.

Sloboda, J. (1985). *The Musical Mind: The Cognitive Psychology of Music*. Oxford University Press.

Stabler, E. P. (2004). Varieties of crossing dependencies: Structure dependence and mild context sensitivity. *Cognitive Science*, 28(4):699–720.

Steedman, M. (1996). The blues and the abstract truth: Music and mental models. In Garnham, A. and Oakhill, J., editors, *Mental Models In Cognitive Science*, pages 305–318. Psychology Press.

SWI (1987). SWI-Prolog. `http://www.swi-prolog.org/` Accessed on 20 December 2014.

Tojo, S. and Hirata, K. (2012). Structural similarity based on time-span tree. In *Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR 2012)*, pages 645–660, London, UK.

Tojo, S., Oka, Y., and Nishida, M. (2006). Analysis of chord progression by HPSG. In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2006)*, Innsbruck, Austria.

Torgerson, W. S. (1952). Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419.

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