Structural Similarity Based on Time-span Tree

Satoshi Tojo[1] and Keiji Hirata[2]

[1] Japan Advanced Institute of Science and Technology tojo@jaist.ac.jp
[2] Future University Hakodate hirata@fun.ac.jp

Abstract. Time-span tree is a stable and consistent representation of musical structure since most experienced listeners deliver the same one, almost independently from context and subjectivity. In this paper, we pay attention to the reduction hypothesis of the tree structure, and introduce the notion of distance as a promising candidate of stable and consistent metric of similarity. First, we design a feature structure to represent a time-span tree. Next, we regard that when a branch is removed from the tree the information corresponding to its time-span is lost, and suggest that the sum of the length of those removed spans is the distance between two trees. We will show that the distance preserves uniqueness in multiple shortest paths, as well as the triangle inequality. Thereafter, we illustrate how the distance works as a metric of similarity, and then, we discuss the feasibility and the problem of our methodology.

Keywords: Similarity, time-span reduction, feature structure, join, meet

1 Introduction

As is remarked in [26], *an ability to assess similarity lies close to the core of cognition*. Musical similarity is multi-faceted as well [15], and this property inevitably raises a context-dependent, subjective behavior [14]. As to context dependency, similarity cannot be perceived in isolation from the musical context in which it occurs. Volk stated in [22]: *Depending on the context, similarity can be described using very different features*. For instance, the impact of cultural knowledge may degrade a stable similarity assessment. As to subjectivity, similarity is likely perceived differently between subjects and even within a subject, depending on listening style, preference, and so on. For instance, [23] revealed that the inconsistency in the annotations by experts is caused by the divergence of four musical dimensions (rhythm, contour, motif, and mode).

Thus far, many researches have explored stable and consistent musical similarity metrics as a central topic in music modelling and music information retrieval [9, 4]. Some of them are motivated by engineering demands such as musical retrieval, classification, and recommendation [15, 7, 18], and others are by modelling the cognitive processes of musical similarity [5, 6]. In this paper, we also seek for a stable and consistent similarity, postponing context-dependency and subjectivity later. We regard that similarity is stable in the sense that similarity assessment is performed only on a score of music, disregarding such context-dependent factors as timber, artist, subject matter of lyrics, and cultural factors. Also, we regard that similarity assessment is consistent in the sense that most experienced listeners can deliver same results as long as the western-tonal-classical style of music is targeted.

9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)
19-22 June 2012, Queen Mary University of London

To propose a stable and consistent similarity, we rely on the cognitive reality or perceptual universality of music theory. As addressed in [24], *systems which aim to encode musical similarity must do so in a human-like way*. Now, we take the stance that *tree* structure underlies such cognitive reality. Bod claimed in his DOP model [1] that there lies cognitive plausibility in combining a rule-based system with a fragment memory when a listener parses music and produces a relevant tree structure, like a linguistic model. Lerdahl and Jackendoff presumed that perceived musical structure is internally represented in the form of hierarchies, which means time-span tree and strong reduction hypothesis in Generative Theory of Tonal Music (GTTM, hereafter) [16, p.2, pp.105-112, p.332]. Dibben argued that the experimental results show that pitch events in tonal music are heard in a strict hierarchical manner and provide evidence for the internal cognitive representation of time-span tree of GTTM [3]. Wiggins et al. deployed discussions on the tree structures and argued that they are more about semantic grouping than about syntactic grouping [25]. We basically follow their view, under which we assume the time-span tree of a melody represents its meaning. Here, we need to admit that GTTM has its innate problem, that is, those ambiguous preference rules may result in multiple time-span analyses; [8] has solved this issue, assigning a parametric weight to each rule, and has implemented an automatic tree analyzer.

In effect, tree representation has contributed to the study on similarity. Marsden began with conventional tree representations and allowed joining of branches in the limited circumstances with preserving the directed acyclic graph (DAG) property for expressing information dependency [13]. As a result, high expressiveness was achieved, while it was difficult to define consistent similarity between melodies. Valero proposed a representation method dedicated to a similarity comparison task, called metrical tree [21]. Valero used a binary tree representing the metrical hierarchy of music and avoided the necessity of explicitly encoding onsets and duration; only pitches needed to be encoded. As a measure to compare metrical trees, Valero adopted the tree edit distance with many parameters, which were justified only by the best performance in experiments, but not by cognitive reality.

Among the properties of time-span tree, in particular, we consider the concept of *reduction* essential, when a time-span tree subsumes a reduced one. Selfridge-Field also claimed that a relevant way of taking deep structures (meaning) into account is to adopt the concept of reduction [19]. Since the subsumption relation between time-span trees can be defined as a partial order, the above consideration may imply a possibility for treating time-span tree (i.e., the meaning of a melody) as a mathematical entity. Our objective is to derive the notion of distance from the reduction and the subsumption relation, to employ it as a metric of similarity. At this time, our attitude toward the design is strictly computational; that is, there must lie a reliable logical and algebraic structure so that we will be able to implement the similarity onto computers.

In the following Section 2, we translate a time-span tree into a feature structure, carefully preventing the other factors from slipping into the structure, to guarantee stability. In Section 3, we define a notion of distance between time-span trees and then show that the notion enjoys several desirable mathematical properties, including the triangle inequality. In Section 4, we illustrate our analysis. In Section 5 we discuss open

problems concerning how we can apply our notion of distance to music similarity, and in Section 6 we summarize our contribution.

2 Time-Span Tree in Feature Structure

In this section, we develop the representation method for time-span tree in [11, 10], in terms of feature structure. First we introduce the general notion of feature structure, and then we propose a set of necessary features to represent a time-span tree. As the set of feature structures are partially ordered, we define such algebraic operations as *meet* and *join* and show that the set becomes a *lattice*. Since this section and the following section include mathematical foundation, those who would like to see examples first may jump to Section 4 and come back to technical details afterward.

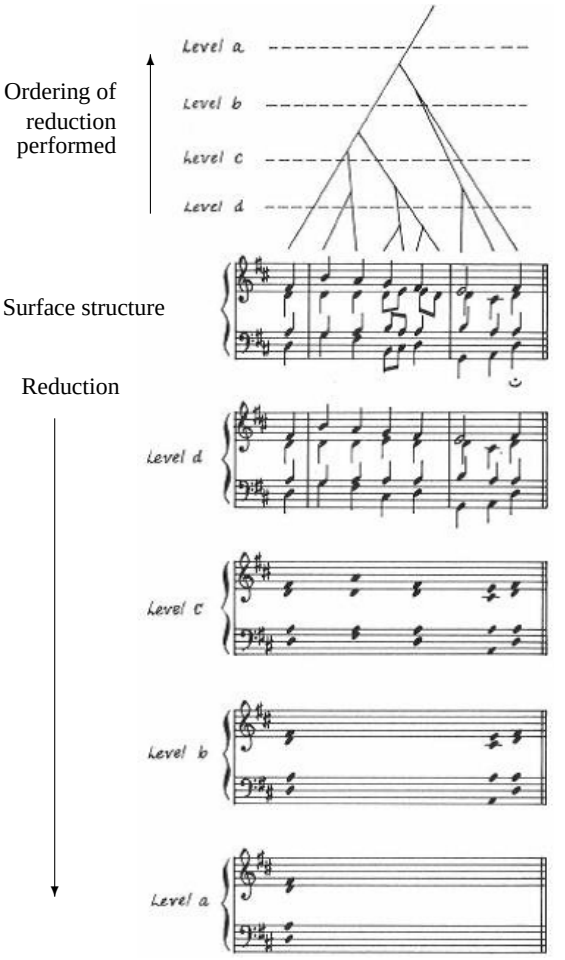


Fig. 1. Time-span reduction in GTTM (Lerdahl and Jackendoff [16, page 115])

2.1 Time-Span Tree and Reduction

A melody is considered to be a sequence of pitch events in temporal order, consisting of a single note and a chord. Time-span reduction [16] assigns structural importance to each pitch events in the hierarchical way. The structural importance is derived from the grouping analysis, in which multiple notes compose a short phrase called a group, and from the metrical analysis, where the regular alternation of strong and weak beats affects. As a result, a time-span tree becomes a binary tree constructed in bottom-up and top-down manners by comparison between the structural importance of adjacent pitch events at different hierarchical levels.

Fig. 1 shows an excerpt from [16] demonstrating the concept of reduction. In the sequence of reductions, each level should sound like a natural simplification of the previous level.[3] The alternative omission of notes must make the successive levels sound less like the original. Hence, reduction can be regarded as rewriting an expression to an equivalent simpler one; it often has the same meaning as abstraction. Since reduction is designed based on Gestalt grouping, the reduction successfully associates a melody with another one that sounds quite similar. The key idea of our framework is that reduction is identified with the subsumption relation, which is the most fundamental relation in knowledge representation.

2.2 Feature Structure and Subsumption Relation

Feature structure (*f-structure*, hereafter) has been mainly studied for applications to linguistic formalism based on unification and constraint, such as Head-driven Phrase Structure Grammar (HPSG)[2, 17]. An f-structure is a list of feature-value pairs where a value may be replaced by another f-structure recursively. Below is an f-structure in attribute-value matrix (AVM) notation where σ is a structure, the label headed by '~ ' (tilde) is the *type* of the whole structure, and f_i's are feature labels and v_i's are their values:

$$\sigma = \begin{bmatrix} \tilde{}type \\ f_1\ v_1 \\ f_2\ v_2 \end{bmatrix} .$$

A type requires its indispensable features. When all these intrinsic features are properly valued, the f-structure is said to be *full-fledged*.

Now we define the notion of *subsumption*. Let σ_1 and σ_2 be f-structures. σ_2 subsumes σ_1, that is, $\sigma_1 \sqsubseteq \sigma_2$ if and only if for any $(f\ \ v_1) \in \sigma_1$ there exists $(f\ \ v_2) \in \sigma_2$ and $v_1 \sqsubseteq v_2$. Since we suppose an f-structure is considered to be the conjunctive set of feature-value pairs, '$\sqsubseteq$' corresponds to the so-called Hoare order of sets (e.g., $\{b, d\} \sqsubseteq \{a, b, c, d\}$). For example, by assuming $v_1 \sqsubseteq [f_3\ \ v_3]$, σ_1 below is subsumed both by the following σ_2 and σ_3.

$$\sigma_1 = \begin{bmatrix} \tilde{}type1 \\ f_1\ v_1 \end{bmatrix}, \qquad \sigma_2 = \begin{bmatrix} \tilde{}type1 \\ f_1\ v_1 \\ f_2\ v_2 \end{bmatrix}, \qquad \sigma_3 = \begin{bmatrix} \tilde{}type1 \\ f_1 \begin{bmatrix} \tilde{}type2 \\ f_3\ v_3 \end{bmatrix} \end{bmatrix} .$$

[3] Once a melody is reduced, each note with onset and duration properties becomes a virtual note that is just a pitch event dominating a corresponding time-span, omitting onset and duration. Therefore, to listen to a reduced melody, we assume that it can be rendered by regarding a time-span as a real note with such onset timing and duration.

Since both σ_2 and σ_3 are elaborations of σ_1, which are differently elaborated, ordering '$\sqsubseteq$' is a partial order, not a total order like integers and real numbers. Equivalence $a = b$ is defined as $a \sqsubseteq b \wedge b \sqsubseteq a$.

To denote value v of feature f in structure σ, we write $\sigma.f = v$. Thus, $\sigma_1.f_1 = v_1$ and $\sigma_1.f_2$ is undefined while $\sigma_3.f_1.f_3 = v_3$. We call a sequence of features $f_1.f_2.\cdots.f_n$ a *feature path*. Structure sharing is indicated by boxed tags such as $\boxed{\text{i}}$ or $\boxed{\text{j}}$. The set value $\{x, y\}$ means the choice either of x or y, and $\bot$ means that the value is empty. Even for $\bot$, any feature f_i is accessible though $\bot.f_i = \bot$.

2.3 Time-Span Trees in F-Structures

We name the type of an f-structure corresponding a time-span tree $\tilde{}tree$.

Definition 1 (Tree Type F-structure) *A full-fledged $\tilde{}tree$ f-structure possesses the following features.*

- *head represents the most salient pitch event in the tree.*
- *span represents the length of the time-span of the whole tree, measured by the number of quarter notes.*
- *dtrs (daughters) are subtrees, whose left and right are recursively $\tilde{}tree$. This dtrs feature is characterized by the following two conditions.*
  - *The value of span must be the addition of two spans of the daughters.*
  - *The value of $head$ is chosen from either that of $left$ or of $right$ daughter.*

If $head = dtrs.left.head$, the node has the right-hand elaboration of shape $\diagup\!\!\diagdown$, while if $head = dtrs.right.head$, the left-hand elaboration $\diagdown\!\!\diagup$. If $dtrs = \bot$ then the tree consists of a single branch with a single pitch event at its leaf.

Fig. 2 shows the examples. Such bold-face letters as **C4**, **E4** and **G4** are pitch events.

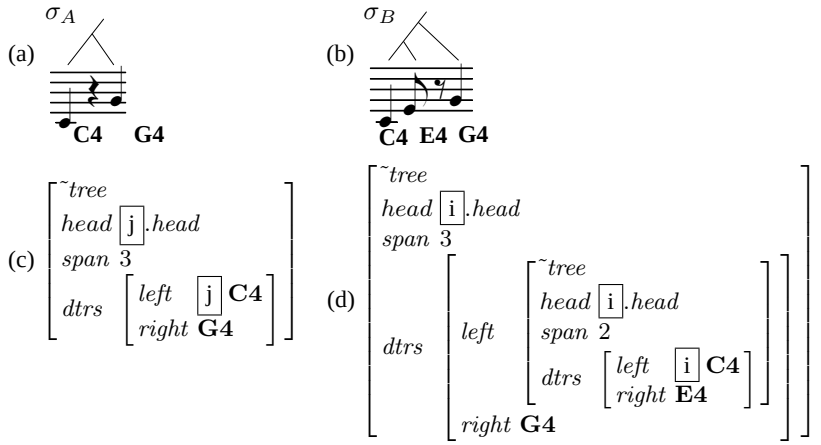


Fig. 2. Melodies (a) and (b) and their f-structures (c) and (d), respectively.

The value of *head* feature is occupied by $\tilde{}event$ f-structure; a full-fledged one should include *pitch*, *onset*, and *duration* features. For example,

$$\mathbf{C4} = \begin{bmatrix} \tilde{}tree & \\ head & \begin{bmatrix} \tilde{}event & \\ pitch & C4 \\ onset & \ldots \\ duration & 1 \end{bmatrix} \\ span & \ldots \\ dtrs & \bot \end{bmatrix}.$$

2.4 Unification, *Join* and *Meet*

Intuitively, unification is a process of information conjunction. We introduce the set notation of an f-structure using the set of feature-path-value pairs: $\{(f_{11}.\cdots.f_{1n}\ v_1), (f_{21}.\cdots.f_{2m}\ v_2), \cdots\}$. Unification is the consistent union of f-structures in the set notation, results in another f-structure. Unification fails only if there exists an inconsistency in any feature-path-value pair.

The set of f-structures are partially ordered as there is the subsumption relation. Here, we can introduce *join* and *meet* operations; *Join* corresponds to a union of sets or a consistent overlay while *meet* does to intersection or the common part.

Definition 2 (*Join*) *Let σ_A and σ_B be full-fledged f-structures representing the time-span trees of melodies A and B, respectively. If we can fix the least upper bound of σ_A and σ_B, that is, the least y such that $\sigma_A \sqsubseteq y$ and $\sigma_B \sqsubseteq y$ is unique, we call such y the join of σ_A and σ_B, denoted as $\sigma_A \sqcup \sigma_B$.*

Theorem 3.13 in Carpenter [2] provides that the unification of f-structures A and B is the least upper bound of A and B, which is equivalent to *join* in this paper. Similarly, we regard the intersection of the unifiable f-structures as *meet*.

Definition 3 (*Meet*) *Let σ_A and σ_B be full-fledged f-structures representing the time-span trees of melodies A and B, respectively. If we can fix the greatest lower bound of σ_A and σ_B, that is, the greatest x such that $x \sqsubseteq \sigma_A$ and $x \sqsubseteq \sigma_B$ is unique, we call such x the meet of σ_A and σ_B, denoted as $\sigma_A \sqcap \sigma_B$.*

We show a musical example in Fig. 3.

Obviously from Definitions 2 and 3, we obtain the absorption laws: $\sigma_A \sqcup x = \sigma_A$ and $\sigma_A \sqcap x = x$ if $x \sqsubseteq \sigma_A$. Moreover, if $\sigma_A \sqsubseteq \sigma_B$, for any x $x \sqcup \sigma_A \sqsubseteq x \sqcup \sigma_B$ and $x \sqcap \sigma_A \sqsubseteq x \sqcap \sigma_B$.

We can define $\sigma_A \sqcup \sigma_B$ and $\sigma_A \sqcap \sigma_B$ in recursive functions. In the process of unification between σ_A and σ_B, when we are to match a subtree with a single branch in the counterpart, if we always choose the subtree the result becomes $\sigma_A \sqcup \sigma_B$ and if we always choose the single branch we obtain $\sigma_A \sqcap \sigma_B$. Because there is no alternative action in these procedures, $\sigma_A \sqcup \sigma_B$ and $\sigma_A \sqcap \sigma_B$ exist uniquely. Thus, the partially ordered set of time-span trees becomes a *lattice*.

Since time-span tree T is rigidly corresponds to f-structure σ, we identify T with σ and may call σ a tree in the following sections as long as no confusion.

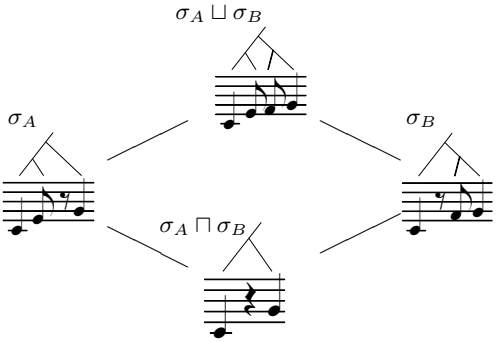


Fig. 3. *Join* and *Meet* operations of time-span trees

3 Strict Distance in Time-Span Reduction

In this section, we introduce the notion of distance between two time-span trees. We propose that:

> *If a branch with a single pitch event is reduced, the information corresponding to the length of its time-span is lost.*

Thus, we regard the accumulation of such lost time-spans as the distance of two trees in the sequence of reductions, called *reduction path*. Thereafter, we generalize the notion to be feasible, not only in a reduction path but in any direction in the lattice. Finally in this section, we show the distance suffices the triangle inequality. Again as this section includes technical details, those who would like to see examples earlier may skip this section and can come back later.

We restrict that branches are reduced only one by one, for the convenience to sum up distances. A branch is *reducible* only when there exists no other lower branch than its junction (attaching point); thus, a reducible branch possesses a single pitch event at its leaf. In the similar way, we restrict that a branch can be an elaboration of some tree only when it consists of a single event and can be attached to a junction under which there is no other branch.

By the way, the *head* pitch event of a tree structure is the representative of the whole tree, whose length appears at *span* feature. Though the event itself retains its original shorter duration, we may regard its supremacy is extended to the tree length. The situation is the same as each subtree. Thus, we consider that each pitch event has the maximal length of dominance.

Definition 4 (Maximal Time-span) *Each pitch event has the maximal time-span within which the event becomes most salient, and outside the time-span its supremacy is lost.*

In Fig. 4, a reducible branch on pitch event e_2 has the time-span s_2. After e_2 is reduced, branch on e_1 becomes reducible and the connected span $s_1 + s_2$ becomes e_1's maximal time-span, though its original duration was s_1. Finally, after e_1 is reduced, e_3 dominates the length of $s_1 + s_2 + s_3$. When e_2 and e_1 are reduced in this order, the distance between σ_A and σ_C becomes $s_2 + (s_1 + s_2)$.

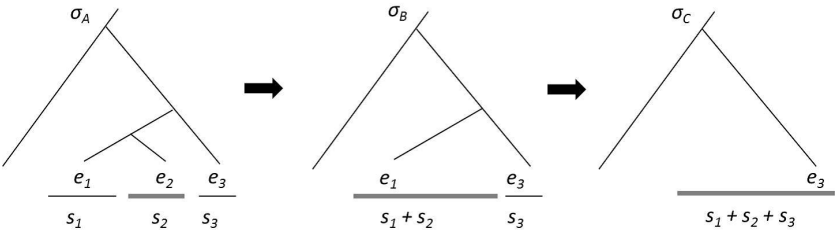


Fig. 4. Reduction by maximal time-spans; gray thick lines denote maximal time-spans while thin ones pitch durations.

Prior to the formal definition of distance, we impose *Head/Span Equality Condition* (*HSEC*, hereafter):

$$\sigma_A.head = \sigma_B.head \ \ \& \ \ \sigma_A.span = \sigma_B.span.$$

We have included this restriction in the following algorithm, so as to avoid any futile comparison; if the identity of two heads and their time-spans is disregarded, the distance between them is meaningless.

Let $\varsigma(\sigma)$ be a set of pitch events in σ, $\sharp\varsigma(\sigma)$ be its cardinality, and s_e be the maximal time-span of event e. Since reduction is made by one reducible branch at a time, a reduction path $\sigma_B = \sigma^n, \sigma^{n-1}, \ldots, \sigma^2, \sigma^1, \sigma^0 = \sigma_A$ suffices $\sharp\varsigma(\sigma^{i+1}) = \sharp\varsigma(\sigma^i) + 1$. For each reduction step, when a reducible branch on event e disappears, its maximal time-span s_e is accumulated as distance.

Definition 5 (Reduction Distance) *The distance $d_{\sqsubseteq}$ of two time-span trees such that $\sigma_A \sqsubseteq \sigma_B$ in a reduction path is defined by*

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

Although the distance is a simple summation of maximal time-spans at a glance, there is a latent order in adding the spans, for reducible branches change dynamically in the process of reduction. In order to give a constructive procedure on this summation, we introduce the notion of total sum of maximal time-spans.

Definition 6 (Total Maximal Time-span) *Given ˜tree f-structure σ,*

$$tms(\sigma) = \sum_{e \in \varsigma(\sigma)} s_e.$$

We present $tms(\sigma)$ as a recursive function in Algorithm 1.

Input: a ˜*tree* f-structure σ
Output: $tms(\sigma)$
1 **if** $\sigma = \bot$ **then**
2 **return** 0;
3 **else if** $\sigma.dtrs = \bot$ **then**
4 **return** $\sigma.span$;
5 **else**
6 **case** $\sigma.head = \sigma.dtrs.left.head$
7 **return** $tms(\sigma.dtrs.left) + tms(\sigma.dtrs.right) + \sigma.dtrs.right.span$;
8 **case** $\sigma.head = \sigma.dtrs.right.head$
9 **return** $tms(\sigma.dtrs.left) + tms(\sigma.dtrs.right) + \sigma.dtrs.left.span$;

Algorithm 1: Total Maximal Time-span

In Algorithm 1, Lines 1–2 are the terminal condition. Lines 3–4 treat the case that a tree consists of a single branch. In Lines 6–7, when the right subtree surrender to the left, the left extends the domination rightward by $\sigma.dtrs.right.span$. Ditto for the case the right-hand side overcomes the left, as Lines 8–9.

When $\sigma_A \sqsubseteq \sigma_B$, from Definition 5 and 6,

$$\begin{aligned} d_{\sqsubseteq}(\sigma_A, \sigma_B) &= \textstyle\sum_{e \in \varsigma(\sigma_B) \setminus \varsigma(\sigma_A)} s_e = \sum_{e \in \varsigma(\sigma_B)} s_e - \sum_{e \in \varsigma(\sigma_A)} s_e \\ &= tms(\sigma_B) - tms(\sigma_A). \end{aligned}$$

As a special case of the above, $d_{\sqsubseteq}(\bot, \sigma) = tms(\sigma)$.

Next, we consider the notion of distance that can be applicable to two trees reside in different paths.

Lemma 1 *For any reduction path from $\sigma_A \sqcup \sigma_B$ to $\sigma_A \sqcap \sigma_B$, $d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_A \sqcup \sigma_B)$ is unique.*

Proof As there is a reduction path between $\sigma_A \sqcap \sigma_B$ and $\sigma_A \sqcup \sigma_B$, and $\sigma_A \sqcap \sigma_B \sqsubseteq \sigma_A \sqcup \sigma_B$, $d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_A \sqcup \sigma_B)$ is computed by the difference of total maximal time-span in Algorithm 1. Because the algorithm returns a unique value, the distance is unique. ∎

Theorem 1 (Uniqueness of Reduction Distance) *If there exist reduction paths from σ_A to σ_B, $d_{\sqsubseteq}(\sigma_A, \sigma_B)$ is unique.*

Lemma 2 $d_{\sqsubseteq}(\sigma_A, \sigma_A \sqcup \sigma_B) = d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_B)$ *and* $d_{\sqsubseteq}(\sigma_B, \sigma_A \sqcup \sigma_B) = d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_A)$.

Proof From set-theoretical calculus, $\varsigma(\sigma_A \sqcup \sigma_B) \setminus \varsigma(\sigma_A) = \varsigma(\sigma_A) \cup \varsigma(\sigma_B) \setminus \varsigma(\sigma_A) = \varsigma(\sigma_B) \setminus \varsigma(\sigma_A) \cap \varsigma(\sigma_B) = \varsigma(\sigma_B) \setminus \varsigma(\sigma_A \sqcap \sigma_B)$. Then, by Definition 5, $d_{\sqsubseteq}(\sigma_A, \sigma_A \sqcup \sigma_B) = \sum_{e \in \varsigma(\sigma_A \sqcup \sigma_B) \setminus \varsigma(\sigma_A)} s_e = \sum_{e \in \varsigma(\sigma_B) \setminus \varsigma(\sigma_A \sqcap \sigma_B)} s_e = d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_B)$. ∎

Definition 7 (Meet and Join Distances)

– $d_{\sqcap}(\sigma_A, \sigma_B) = d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_A) + d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_B)$ *(meet distance)*

– $d_{\sqcup}(\sigma_A, \sigma_B) = d_{\sqsubseteq}(\sigma_A, \sigma_A \sqcup \sigma_B) + d_{\sqsubseteq}(\sigma_B, \sigma_A \sqcup \sigma_B)$ *(join distance)*

Lemma 3 $d_{\sqcup}(\sigma_A, \sigma_B) = d_{\sqcap}(\sigma_A, \sigma_B)$.

Proof Immediately from Lemma 2. ∎

Lemma 4 *For any* σ', σ'' *such that* $\sigma_A \sqsubseteq \sigma' \sqsubseteq \sigma_A \sqcup \sigma_B$, $\sigma_B \sqsubseteq \sigma'' \sqsubseteq \sigma_A \sqcup \sigma_B, d_{\sqcup}(\sigma_A, \sigma') + d_{\sqcap}(\sigma', \sigma'') + d_{\sqcup}(\sigma'', \sigma_B) = d_{\sqcup}(\sigma_A, \sigma_B)$. *Ditto for the meet distance.*

Now the notion of distance, which was initially defined in the reduction path as $d_{\sqsubseteq}$ is now generalized to $d_{\{\sqcap,\sqcup\}}$, and in addition we have shown they have the same values. From now on, we omit $\{\sqcap, \sqcup\}$ from $d_{\{\sqcap,\sqcup\}}$, simply denoting 'd'.

Theorem 2 (Uniqueness of Distance) $d(\sigma_A, \sigma_B)$ *is unique among shortest paths between* σ_A *and* σ_B.

Note that shortest paths can be found in ordinary graph-search methods, such as *branch and bound*, Dijkstra's algorithm, best-first search, and so on.

Corollary 1 $d(\sigma_A, \sigma_B) = d(\sigma_A \sqcup \sigma_B, \sigma_A \sqcap \sigma_B)$.

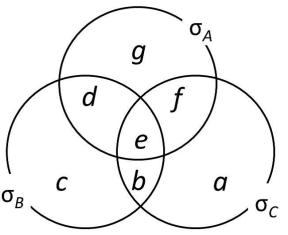
Proof From Lemma 2 and Lemma 3. ∎

Theorem 3 (Triangle Inequality) *For any* σ_A, σ_B *and* σ_C, $d(\sigma_A, \sigma_B) + d(\sigma_B, \sigma_C) \geq d(\sigma_A, \sigma_C)$.

Proof From Corollary 1 and by definition,

$$d(\sigma_i, \sigma_j) = d(\sigma_i \sqcup \sigma_j, \sigma_i \sqcap \sigma_j) = \textstyle\sum_{e \in \varsigma(\sigma_i \sqcup \sigma_j) \setminus \varsigma(\sigma_i \sqcap \sigma_j)} s_e.$$

Since we employ the set-notation of f-structure (cf. Section 2.4), the relationship between $\sigma_{\{A,B,C\}}$ can be depicted in Venn diagram. Then, $d(\sigma_A, \sigma_B) + d(\sigma_B, \sigma_C)$ becomes the sum of maximal time-spans in $\varsigma(\sigma_A \sqcup \sigma_B) \setminus \varsigma(\sigma_A \sqcap \sigma_B)$ plus those in $\varsigma(\sigma_B \sqcup \sigma_C) \setminus \varsigma(\sigma_B \sqcap \sigma_C)$, which corresponds to $(f + g + b + c) + (a + c + d + f) = a + b + 2c + 2f + d + g$ in the diagram. On the contrary, $d(\sigma_A, \sigma_C)$ becomes the sum of $a + b + d + g$. Since $(a+b+2c+2f+d+g) - (a+b+d+g) = 2c+2f \geq 0$, we obtain the result. ∎



In the above proof, c and f are counted twice because branches in these areas are once reduced and later added, or once added and later reduced. This implies that these reduction/addition can be skipped and there exists a short cut between σ_A and σ_C without visiting σ_B.

Finally in this section, we suggest that the distance can be a metric of similarity between two music pieces. As long as we stay in the lattice of reductions under *HSEC*, the distance exactly reflects the similarity. However, even though *head*s and *span*s are different in two pieces of music, we can calculate the similarity with our notion of distance. We show such examples in Section 4.

4 Examples

In this section, we illustrate our analyses. The first example is Mozart's K265, *Ah! vous dirais-je, maman*, equivalent to *Twinkle, Twinkle, Little Star*. The melody in the left-hand side of Fig. 5 is the theme, while those in the right-hand side are the third variation and its reduced melodies in downward order. The horizontal lines below each score are the maximal time-spans of pitch events though we omit explicit connection between events and lines in the figure. The lines drawn at the bottom level in each score correspond to reducible branches (i.e., reducible pitch events) at that step. For example, from Level c in the right-hand side of Fig. 5 to Level b, eight maximal time-spans of $1/3$-long disappear by reduction, thus, according to Algorithm 1 the distance is $1/3 \times 8 = 8/3$. The configuration of maximal time-spans at Level a in the right-hand

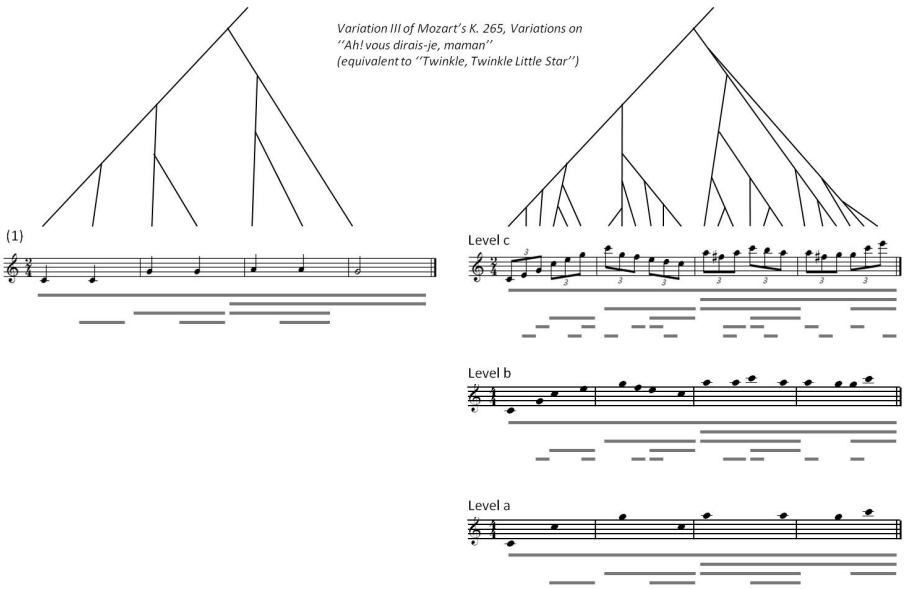


Fig. 5. Reduction of *Mozart: Ah! vous dirais-je, maman*

of Fig. 5 quite resembles that in the left-hand side, which is the theme of the variation. Actually, since the difference between (1) and Level a is the rightmost quarter note in the 4-th measure, the distance between these two is so close as just 1. This implies that we can retrieve the theme by reducing the variation.

In Fig. 6, we have arranged various reductions originated from a piece. As we can find three reducible branches in A we possess three different reductions: B, C, and D. In the figure, C (shown diluted) lies at the back of the lattice where three back-side edges meet.

The distances, represented by the length of edges, from A to B, D to F, C to E, and G to H are same, since the reduced branch is common. Namely, the reduction

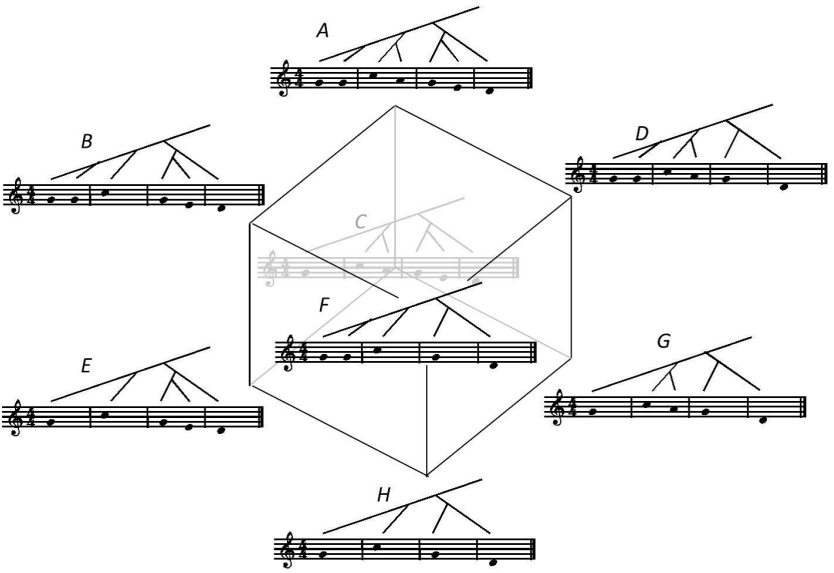


Fig. 6. Reduction lattice

lattice becomes parallelepiped,[4] and the distances from A to H becomes uniquely $2 + 2 + 2 = 6$, which we have shown as Theorem 1. We exemplify the triangle inequality (Theorem 3); from A through B to F, the distance becomes $2+2 = 4$, and that from F through D to G is $2+2 = 4$, thus the total path length becomes $4+4 = 8$. But, we can find a shorter path from A to G via D, in which case the distance becomes $2 + 2 = 4$. Notice that the lattice represents the operations of *join* and *meet*; e.g., $F = B \sqcap D$, $D = F \sqcup G$, $H = E \sqcap F$, and so on. In addition, the lattice is locally Boolean, being A and H regarded to be $\top$ and $\bot$, respectively. That is, there exists a complement,[5] and $E^c = D$, $C^c = F$, $B^c = G$, and so on.

In the next example, we compare two time-span trees in reduction. The left-hand side in Fig. 7 is *Massa's in De Cold Ground* (Stephen Collins Foster, 1852) and the right-hand side is *Londonderry Air* (transposed to C major). The vertical distance is strictly computable in each reduction, but in addition, we may notice that these two pieces are quite near in their skeletons in the abstract levels. Especially, we should compare the configurations of maximal time-spans in the bottom three levels and find them topologically equal to each other. This means the distance becomes 0, being *HSEC* disregarded. Then, in the next section, we discuss how to compute the distance where *HSEC* does not hold.

[4] In the case of Fig. 6, as all the edges have the length of 2, the lattice becomes a cube.
[5] For any member X of a set, there exists X^c and $X \sqcup X^c = \top$ and $X \sqcap X^c = \bot$.

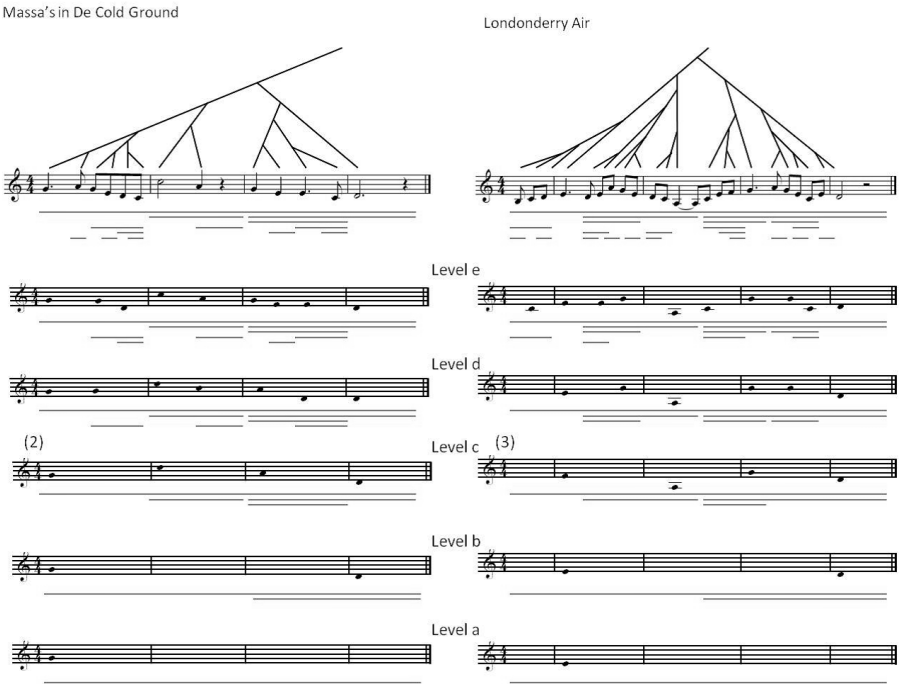


Fig. 7. Reduction processes of *Massa's in De Cold Ground* and *Londonderry Air*

5 Discussion

In this section, we discuss several open problems. In Section 2, we have introduced the representation of time-span tree in f-structure and *join* and *meet* operations, which however only work properly under *HSEC*. From a practical point of view, this condition is too restrictive for arbitrarily given two melodies. We found that *Massa's in De*

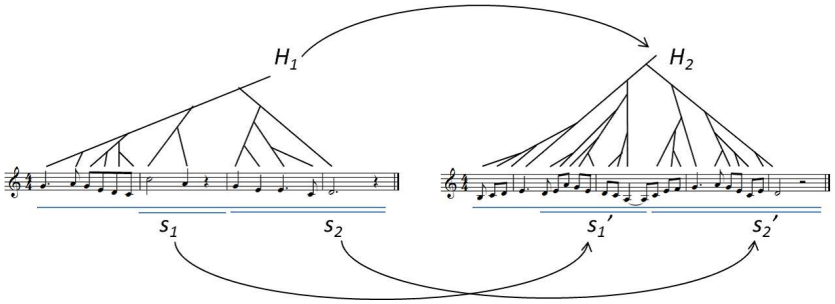


Fig. 8. flexible matching

Cold Ground and *Londonderry Air* do not share strictly common time-span trees, but are somewhat similar as a result of reduction as in Fig. 7. Since we actually recognize a flavor of similarity in them, we have a good reason to seek for a more flexible mechanism to map *head*s and *span*s as in Fig. 8 in *join* and *meet* computation. The situation is same for the comprison of pitch events residing at *head* feature. For the purpose, we have to provide the subsumption relations in time-spans and in pitch events, grounded to cognitive reality; if these partial orders truly coincide with our intuition or perception, we can tolerate the condition of unificaiton.

The similarity measures widely used in data mining and information retrieval include Jaccard, Simpson, Dice, and Point-wise mutual information (PMI) [20]. For instance, the Jaccard index (also known as Jaccard similarity coefficient) is regarded as an index of the similarity of two sets.

$$sim(\sigma_A, \sigma_B) = \frac{|\sigma_A \sqcap \sigma_B|}{|\sigma_A \sqcup \sigma_B|},$$

Here, we may naïvely interpret '$|\sigma|$' as the set of pitch events in the tree as '$\sharp_\varsigma(\sigma)$'. However, the number of notes does not fully reflect the internal structure. Then, it may be appropriate to weight an individual note by its time-span, and the content of a structure hence amounts to the total maximal time-span $tms(\sigma)$ in Definition 6, as

$$sim(\sigma_A, \sigma_B) = \frac{tms(\sigma_A \sqcap \sigma_B)}{tms(\sigma_A \sqcup \sigma_B)}.$$

Since the value of $tms(\sigma)$ represents the complexity of the whole structure, we can also consider the *density* of notes in the music piece. Similarly, we may make use of Simpson index with tms as follows:

$$sim(\sigma_A, \sigma_B) = \frac{tms(\sigma_A \sqcap \sigma_B)}{min(tms(\sigma_A), tms(\sigma_B))}.$$

We have treated the maximal time-spans evenly, independent of their lengths and levels at which they occur. However, suppose we listen to two melodies of the same length; one is with full of short notes while the other with a few long notes, then the psychological lengths of these two melodies may be different. This effect is actually well known as the Weber-Fechner law; the relationship between stimulus and perception is logarithmic in auditory and visual psychology. Since our initial purpose of this paper has been to present a stable and consistent similarity, we could not reflect such perceptional aspects.

6 Conclusions

In this paper, we relied on the strong reduction hypothesis of the tree structure in GTTM, and presented the notion of metric of similarity, based on the distance of reduction. In order to do that, we first designed an f-structure to represent a time-span tree, and we showed that its *head* feature and *span* feature properly reflected the original structure

proposed in GTTM. Thereafter, we regarded that a reduction was the loss of information, and the loss was quantified by the time-span of a reduced event. We defined the notion of distance by the lost time-span, and have generalized the notion as the metric of similarity. We have shown several mathematical properties concerning the metric, including uniqueness of distance in any shortest paths as well as the triangle inequality.

Our contribution in this paper is two-fold. One is that we have presented a stable and consistent metric of similarity, which does rely on neither subjective nor context-dependent factor. The other is that our metric is mathematically so sound that it can be employed in the framework of well-known traditional measures, such as Jaccard/Simpson indices.

At present, we have the following five open problems entangled each other. First, (i) if we are to apply our unification mechanism such as *join* and *meet* operations to practical problems, e.g., melodic morphing, we need to ease *HSEC*. Also, (ii) we need more statistical witness in comparison of such existing metrics as Jaccard/Simpson indices, referring to a large-scale music database. As was mentioned in Section 5, (iii) we have treated the maximal time-spans evenly, disregarding the psychological length of music. Since we have postponed such subjective and context-dependent metric, we are obliged to face this aspect from now. By the way, (iv) we still have various alternatives to render each reduced event on actual staff. Though we have mentioned this in the footnote 3 in Section 2.1, the problem is left undone. Finally, (v) the more fundamental problem is the reliability of time-span tree. We admit that some processes in the time-span reduction is still fragile and proper reduction is not promised yet. Thus far we have tackled the automatic reduction system, and even from now on we need to improve the system performance. All in all, to apply such an objective metric to practical cases we need further consideration, that would be our future works.

Acknowledgment

The authors would like to thank the all anonymous reviewers for their fruitful comments, which helped us to develop the contents and to improve the readability. This work was supported by KAKENHI 23500145, Grants-in-Aid for Scientific Research of JSPS.

References

1. Bod, R.: A Unified Model of Structural Organization in Language and Music. Journal of Artificial Intelligence Research 17, 289–308 (2002)
2. Carpenter, B.: The Logic of Typed Feature Structures. Cambridge University Press (1992)
3. Dibben, N.: Cognitive Reality of Hierarchic Structure in Tonal and Atonal Music. Music Perception 12(1), 1–25 (Fall 1994)
4. Downie, J.S., Byrd, D., Crawford, T.: Ten Years of ISMIR: Reflections of Challenges and Opportunities. In: Proceedings of ISMIR 2009, 13–18
5. ESCOM: 2007 Discussion Forum 4A. Similarity Perception in Listening to Music. Musicæ Scientiæ
6. ESCOM: 2009 Discussion Forum 4B. Musical Similarity. Musicæ Scientiæ

7. Grachten, M., Arcos, J.-L., de Mantaras, R.L.: Melody retrieval using the Implication/Realization model. 2005 MIREX. http://www.music-ir.org/evaluation/mirexresults/articles/similarity/grachten.pdf
8. Hamanaka, M., Hirata, K., Tojo, S.: Implementing "A Generative Theory of Tonal Music". Journal of New Music Research 35(4), 249–277 (2007)
9. Hewlett, W.B., Selfridge-Field, E.: Melodic Similarity. Computing in Musicology 11, The MIT Press (1998)
10. Hirata, K., Tojo, S.: Lattice for Musical Structures and Its Arithmetics. LNAI 4384 (Selected Papers from JSAI 2006, T. Washio et al. (Eds)) Springer-Verlag, 54–64 (2007)
11. Hirata, K., Tojo, S., Hamanaka, M.: Melodic Morphing Algorithm in Formalism, In: Proceedings of 3rd International Conference, MCM 2011 (LNAI 6726), 338–341
12. Lartillot, O.: Multi-Dimensional Motivic Pattern Extraction Founded on Adaptive Redundancy Filtering. Journal of New Music Research 34(4), 375–393 (2005)
13. Marsden, A.: Generative Structural Representation of Tonal Music. Journal of New Music Research 34(4), 409–428 (2005)
14. Ockelford, A.: Similarity relations between groups of notes: Music-theoretical and music-psychological perspectives. In: Musicae Scientiae, Discussion Forum 4B, Musical Similarity, 47–98 (2009)
15. Pampalk, E.: Computational Models of Music Similarity and their Application in Music Information Retrieval. PhD Thesis, Vienna University of Technology (March 2006)
16. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music. The MIT Press (1983)
17. Sag, I.A., Wasow, T.: Syntactic Theory: A Formal Introduction. CSLI Publications (1999)
18. Schedl, M., Knees, P., Böck, S.: Investigating the Similarity Space of Music Artists on the Micro-Blogosphere. In: Proceedings of ISMIR 2011, 323–328
19. Selfridge-Field, E.: Conceptual and Representational Issues in Melodic Comparison. Computing in Musicology 11, 3–64 (1998)
20. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley (2005)
21. Valero, D.R.: Symbolic Music Comparison with Tree Data Structure. Ph.D. Thesis, Universitat d' Alacant, Departamento de Lenguajes y Sistemas Informatícos (2010)
22. Volk, A., Wiering, F.: Music Similarity. In: ISMIR 2011 Tutorial on Musicology. http://ismir2011.ismir.net/tutorials/ISMIR2011-Tutorial-Musicology.pdf
23. Volk, A., van Kranenburg, P., Garbers, J., Wiering, F., Veltkamp, R.C., Grijp, L.P.: A manual annotation method for melodic similarity and the study of melody feature sets. In: Proceedings of ISMIR 2008, 101–106
24. Wiggins, G.A.: Semantic Gap?? Schematic Schmap!! Methodological Considerations in the Scientific Study of Music. In: 2009 11th IEEE International Symposium on Multimedia, 477–482
25. Wiggins, G.A., Müllensiefen, D., Pearce, M.T.: On the non-existence of music: Why music theory is a figment of the imagination. In: Musicae Scientiae, Discussion Forum 5, 231–255 (2010)
26. Wilson, R.A., Keil, F. (Eds): The MIT Encyclopedia of the Cognitive Sciences. The MIT Press (May 1999)