Journal of Information Processing Vol.21 No.2 256–263 (Apr. 2013)

[DOI: 10.2197/ipsjjip.21.256]

Regular Paper

Distance and Similarity of Time-span Trees

Satoshi Tojo[1,a)] Keiji Hirata[2,b)]

Received: July 3, 2012, Accepted: January 11, 2013

Abstract: Time-span tree in Lerdahl and Jackendoff's theory [12] has been regarded as one of the most dependable representations of musical structure. We first show how to formalize the time-span tree in *feature structure*, introducing *head* and *span* features. Then, we introduce *join* and *meet* operations among them. The *span* feature represents the temporal length during which the *head* pitch event is most salient. Here, we regard this temporal length as the amount of information which the pitch event carries; i.e., when the pitch event is reduced, the information comparable to the length is lost. This allows us to define the notion of distance as the sum of lost time-spans. Then, we employ the distance as a promising candidate of stable and consistent metric of similarity. We show the distance possesses proper mathematical properties, including the uniqueness of the distance among the shortest paths. After we show examples with concrete music pieces, we discuss how our notion of distance is positioned among other notions of distance/similarity. Finally, we summarize our contributions and discuss open problems.

Keywords: GTTM, time-span tree, music structure, distance, similarity

1. Introduction

Many research initiatives have explored stable and consistent musical similarity metrics as a central topic in music modelling and music information retrieval [4], [9]. Some of them are motivated by engineering demands such as music retrieval, classification, and recommendation [7], [15], [18], and others are by modelling the cognitive process as reported in the Discussion Forum on music similarity [5], [6]. In this paper, we also seek for a stable and consistent similarity, avoiding context-dependency and subjectivity [21]. However, as is remarked in Ref. [25], *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]. For instance, Volk stated in Ref. [22]: *Depending on the context, similarity can be described using very different features*.

To propose a stable and consistent similarity, we rely on such music theory that represents the cognitive reality or perceptual universality. As addressed in Ref. [23], *systems which aim to encode musical similarity must do so in a human-like way*. Now, we take the stance that *tree* structure underlies the 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) [12], 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 [24]. We basically follow their views and hypothesize the time-span tree of a melody represents its meaning.

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]. The subsumption relation between time-span trees can be defined as a partial order, and thus we may be able to treat time-span tree (i.e., the meaning of a melody) as a mathematical entity.

On the other hand, there are tree representation designed for assessing similarity and measuring distance. Marsden began with conventional tree representations and allowed joining of branches in the limited circumstances with preserving the directed acyclic graph (DAG) for expressing information dependency [13]. As a result, high expressiveness was achieved, while it was difficult to define consistent similarity between melodies. Rizo Valero proposed a representation method dedicated to a similarity comparison task, called metrical tree [16]. He used a binary tree representing the metrical hierarchy of music in which he avoided encoding onsets and duration, and only pitches were encoded. As a measure to compare metrical trees, he adopted the tree edit distance with many parameters, which were justified only by the best performance in experiments, but not by cognitive reality.

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

[1] Japan Advanced Institute of Science and Technology, Nomi Ishikawa 923–1292, Japan
[2] Future University Hakodate, Hakodate, Hokkaido 041–0803, Japan
[a)] tojo@jaist.ac.jp
[b)] hirata@fun.ac.jp

that the notion enjoys several desirable mathematical properties, including the triangle inequality that ordinary distance metrics should satisfy. In Section 4, we illustrate our analysis. In Section 5 we discuss how our distance can be placed in ordinary notions of distance and similarity. In Section 6 we summarize our contribution and open problems.

2. Time-Span Tree in Feature Structure

In this section, we develop the representation method for time-span tree in Refs. [10], [11], [21], in terms of *feature structure* [2]. 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 feature structures can be partially ordered, we can define such algebraic operations as *meet* and *join* and thus we 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.

2.1 Time-Span Tree and Reduction

Time-span reduction [12] 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 strong and weak beats are properly assigned on each pitch event. 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 each hierarchical level. Although a pitch event means a single note or a chord, we restrict our interest to monophonic analysis in this paper as the method of chord recognition is not included in the original theory.

In the sequence of reductions, each reduction should sound like a simplification of the previous one [*1]. In other words, the more reductions proceed, each sounds dissimilar to the original. Reduction can be regarded as abstraction, but if we could find a proper way of reduction, we can retrieve a basic melody line of the original music piece. 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) [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 $(f\ v) \in \sigma_1$ there exists $(f\ v) \in \sigma_2$ [*2]. Here '$\sqsubseteq$' corresponds to the so-called Hoare order of sets (e.g., $\{b, d\} \sqsubseteq \{a, b, c, d\}$). For example, σ_1 below is subsumed by the following σ_2 but not by σ_3 unless v_1 is another f-structure such that $v_1 \sqsubseteq [f_3\ \ v_3]$.

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

Since there is no direct subsumption relation in σ_2 and σ_3, 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 $\sigma.f = v$. In the above, $\sigma_1.f_1 = v_1$ 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 $\perp$ means that the value is empty. Even for $\perp$, any feature f_i is accessible though $\perp.f_i = \perp$.

2.3 Time-Span Trees in F-Structures

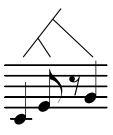
We define type *~tree* of f-structure, to represent a time-span tree, as follows.

Definition 1 (Tree Type F-structure) A full-fledged *~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 *~tree*. This *dtrs* feature is characterized by the following two conditions.
  - The value of *span* must be the addition of two *span*s of the daughters.
  - The value of *head* is chosen from either that of *left* or of *right* daughter.

If $dtrs = \perp$ then the tree consists of a single branch with a single pitch event at its leaf.

For example,



is represented by:

[*1] Once a music piece is reduced, each note with onset and duration properties becomes a virtual note that is just a pitch event being salient during the corresponding time-span, omitting onset and duration. Therefore, to listen to a reduced melody, we assume that it needs to be rendered by regarding a time-span as a real note with such onset timing and duration.

[*2] When a subsumption relation is also defined in atomic values, e.g., $v_1 \sqsubseteq v_2$, $\sigma_1 \sqsubseteq \sigma_2$ if and only if for any $(f\ v_1) \in \sigma_1$ there exists $(f\ v_2) \in \sigma_2$.

$$\begin{bmatrix} \tilde{}tree & \\ head & \boxed{\text{i}}.head \\ span & 3 \\ dtrs & \begin{bmatrix} left & \begin{bmatrix} \tilde{}tree & \\ head & \boxed{\text{i}}.head \\ span & 2 \\ dtrs & \begin{bmatrix} left & \boxed{\text{i}}\ \mathbf{C4} \\ right & \mathbf{E4} \end{bmatrix} \end{bmatrix} \\ right & \mathbf{G4} \end{bmatrix} \end{bmatrix}.$$

Such bold-face letters as **C4**, **E4** and **G4** are trees for pitch events, in which the value of *head* feature is occupied by *˜event* f-structure with *pitch*, *onset*, and *duration* features, where *duration* of *˜event* coincides with that of *span* in its upper *˜tree*. For example,

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

2.4 Unification, *Join* and *Meet*

We introduce the set notation of an f-structure using the set of (feature path, value) pairs: $\{(f_{11}.\cdots.f_{1n}\ v_1), (f_{21}.\cdots.f_{2m}\ v_2), \cdots\}$. Given two f-structures in which a common feature appears, we say they are *inconsistent* if the values of the feature does not match. Unification is the consistent union of f-structures in the set notation, resulting in another f-structure.

Now, when we compare two f-structures for unification, if there is a missing feature f_i on one f-structure let us complement it with $(f_i\ \bot)$. For example, we identify

$$\sigma_4 = \begin{bmatrix} \tilde{}type1 & \\ f_1 & v_1 \end{bmatrix} \text{ and } \sigma_5 = \begin{bmatrix} \tilde{}type1 & \\ f_2 & v_2 \end{bmatrix},$$

with

$$\begin{bmatrix} \tilde{}type1 & \\ f_1 & v_1 \\ f_2 & \bot \end{bmatrix} \text{ and } \begin{bmatrix} \tilde{}type1 & \\ f_1 & \bot \\ f_2 & v_2 \end{bmatrix},$$

respectively. Here, we extend the definition of unification in two different ways. If the unification of two values of v_i and $\bot$ is redefined as v_i, we call *join* operation; if the same two becomes $\bot$, we call *meet* operation. Then,

$$join(\sigma_4, \sigma_5) = \begin{bmatrix} \tilde{}type1 & \\ f_1 & v_1 \\ f_2 & v_2 \end{bmatrix}.$$

while $meet(\sigma_4, \sigma_5) = \bot$.

Although we have introduced the notions of *join*/*meet* in terms of unification, we should define these operations in terms of the subsumption relation in f-structures to emphasize their intrinsic property.

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 $\sigma_A \sqcup \sigma_B$.

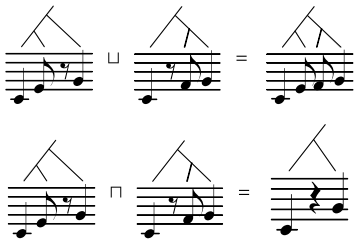


Fig. 1 *join* and *meet*.

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 illustrate *join* and *meet* in a simple example in **Fig. 1**. The '$\sqcup$' (*join*) operation takes quavers in the scores to fill *dtrs* value, so that missing note in one side is complemented. On the other hand, the '$\sqcap$' (*meet*) operation takes $\bot$ for mismatching features, and thus only the common notes appear as a result.

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.

3. Strict Distance in Time-Span Reduction

In GTTM, the following hypothesis is introduced: a listener mentally constructs pitch hierarchies (reductions) that express maximal importance among pitch relations [12], p.118. To define the domains over which reduction takes place, a hierarchy of time-spans is provided. We here observe a time-span becomes longer as the level of time-span hierarchy goes higher. Then, we can suppose that a longer time-span contains more information, and it is therefore regarded more important.

Based on the above consideration, we introduce the hypothesis of distance between two time-span trees as follows:

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

Thus, we regard the accumulation of such lost time-spans as the

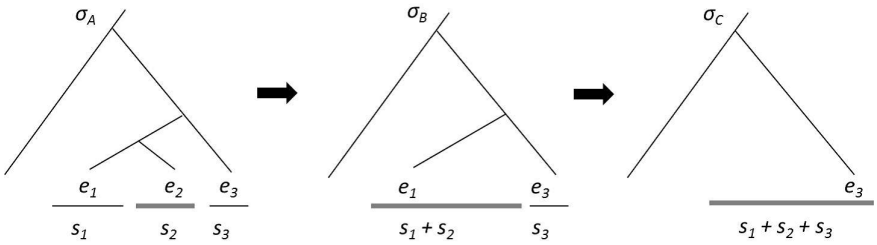


Fig. 2 Reduction by maximal time-spans; gray thick lines denote maximal time-spans while thin ones pitch durations.

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 [21]. Again as this section includes technical details, those who would like to see examples earlier may jump to Section 4 and can come back later.

3.1 Preparation

We presuppose that branches are reduced only one by one, for the convenience to sum up distances. A branch is *reducible* only in the bottom-up way, i.e., a reducible branch possesses no other sub-branches except a single pitch event at its leaf. In the similar way, we call the reverse operation *elaboration*; we can attach a new sub-branch when the original branch consists only of a single event.

The *head* pitch event of a tree structure is the most salient event of the whole tree, and the temporal duration of the tree appears at *span* feature. Though the event itself retains its original duration, we may regard its saliency is extended to the whole tree. The situation is the same as each subtree. Thus, we consider that each pitch event has the maximal length of saliency.

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 the salience is lost.

In **Fig. 2**, 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 becomes most salient during the length of $s_1 + s_2 + s_3$.

Prior to *join*/*meet* operations, if either two heads or their time-spans of time-span trees are different, the comparison itself is futile. Therefore, we impose *Head/Span Equality Condition* (*HSEC*, hereafter):

when σ_A *and* σ_B *are not* $\bot$, $\sigma_A.head = \sigma_B.head$ & $\sigma_A.span = \sigma_B.span$

on the operations. Note that if $\sigma.dtrs = \bot$, i.e., the tree consists of a single pitch event, we do not need to care this head/span equality, as $\sigma \sqcup \bot = \sigma$ and $\sigma \sqcap \bot = \bot$.

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

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

For example in Fig. 2, the distance between σ_A and σ_C becomes $s_2 + (s_1 + s_2)$ when e_2 and e_1 are reduced in this order, since the reduction of e_2 yields s_1 and e_1 yields $(s_1 + s_2)$. Although the distance is a simple summation of maximal time-spans at a glance, there is a latent order in the addition, for reducible branches are different in each reduction step. 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* type 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(σ)
1  if σ = ⊥ then
2      return 0;
3  else if σ.dtrs = ⊥ then
4      return σ.span;
5  else
6      case σ.head = σ.dtrs.left.head
7          return
           tms(σ.dtrs.left)+tms(σ.dtrs.right)+σ.dtrs.right.span;
8      case σ.head = σ.dtrs.right.head
9          return
           tms(σ.dtrs.left) + tms(σ.dtrs.right) + σ.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 saliency 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,

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

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

3.2 Properties of Distance

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, $\varsigma(\sigma_A \sqcup \sigma_B) \setminus \varsigma(\sigma_A) = \varsigma(\sigma_B) \setminus \varsigma(\sigma_A \sqcap \sigma_B)$. 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)

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

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,

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

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 $\varsigma(\sigma_B \sqcup \sigma_C) \setminus \varsigma(\sigma_B \sqcap \sigma_C)$, while $d(\sigma_A, \sigma_C)$ becomes $\varsigma(\sigma_A \sqcup \sigma_C) \setminus \varsigma(\sigma_A \sqcap \sigma_C)$. Thus, $d(\sigma_A, \sigma_B) + d(\sigma_B, \sigma_C) \geq d(\sigma_A, \sigma_C)$. ∎

In **Fig. 3**, we have laid out various reductions originated from a piece. As we can find three reducible branches in A there are three different reductions: B, C, and D. In the figure, C (shown diluted) lies behind 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 the same, since the reduced branch is common. Namely, the reduction lattice becomes parallelepiped [*3], and the distances from A to H becomes uniquely

[*3] In the case of Fig. 3, as all the edges have the length of 2, the lattice becomes equilateral.

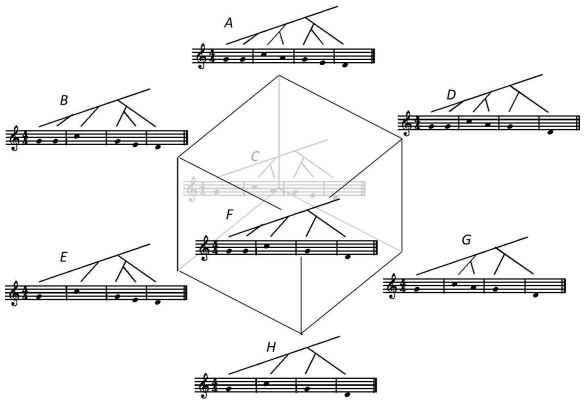


Fig. 3 Reduction lattice.

$2+2+2=6$, 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 either C or 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 [*4], and $E^c = D$, $C^c = F$, $B^c = G$, and so on.

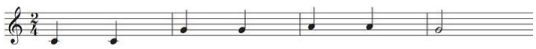
4. Examples

In this section, we concretely assess the distance of time-spans of music pieces.

The first is a rather simple comparison. The left-hand side in **Fig. 4** is *Massa's in De Cold Ground* (Stephen Collins Foster, 1852) and the right-hand side is *Londonderry Air* (transposed to C major). Their reduced melodies are shown in the 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. We may notice that these two pieces are quite near in their skeletons in the abstract levels. Especially, compare the configurations of maximal time-spans in the bottom three levels and find them topologically equal to each other.

Note, however, that we cannot calculate the distance between two arbitrary music pieces yet under the strict *HSEC* (cf. Section 3.1). Thus, the demonstrated comparison in this section is approximate and/or intuitive in some sense.

The next example is about Mozart's K265/300e *Ah! vous dirais-je, maman*, equivalent to *Twinkle, Twinkle, Little Star*.



The melody in the left-hand side of **Fig. 5** is the first variation while those in the right-hand side are the seventh variation. As both of which consist of sequences of semiquavers, we would like to compare these two variations. Here, we have marked asterisks '$*$' below different maximal-time spans, seeing each reduction of two variations laterally. Let us denote $A \approx B$ here if both of A and

[*4] For any member X of a set, there exists X^c and $X \sqcup X^c = \top$ and $X \sqcap X^c = \bot$.

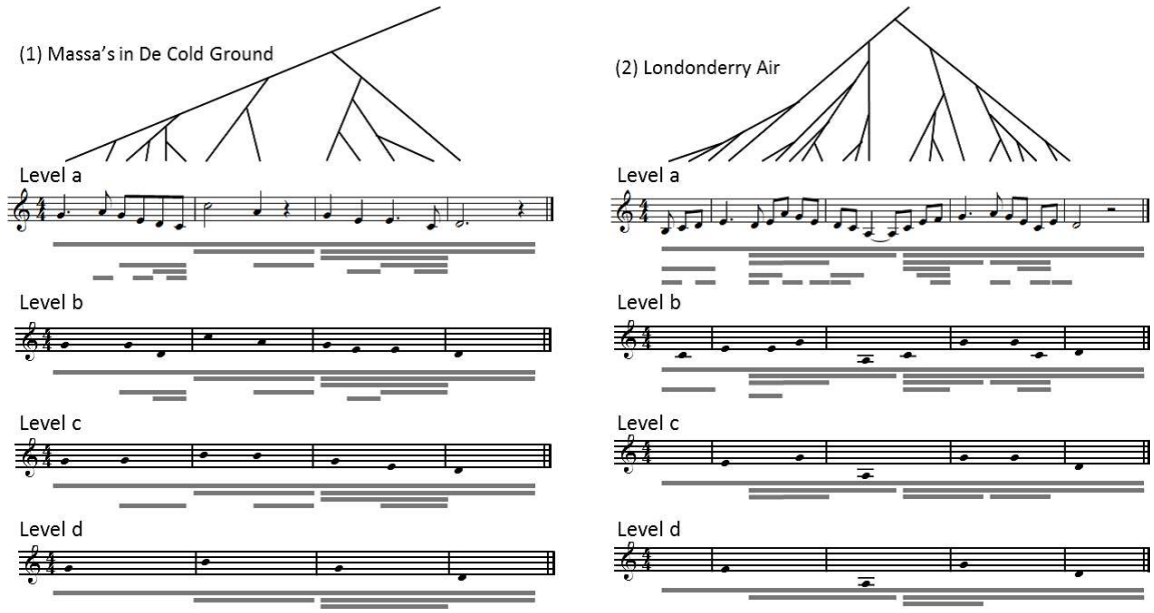


Fig. 4 Reduction processes of *Massa's in De Cold Ground* and *Londonderry Air*.

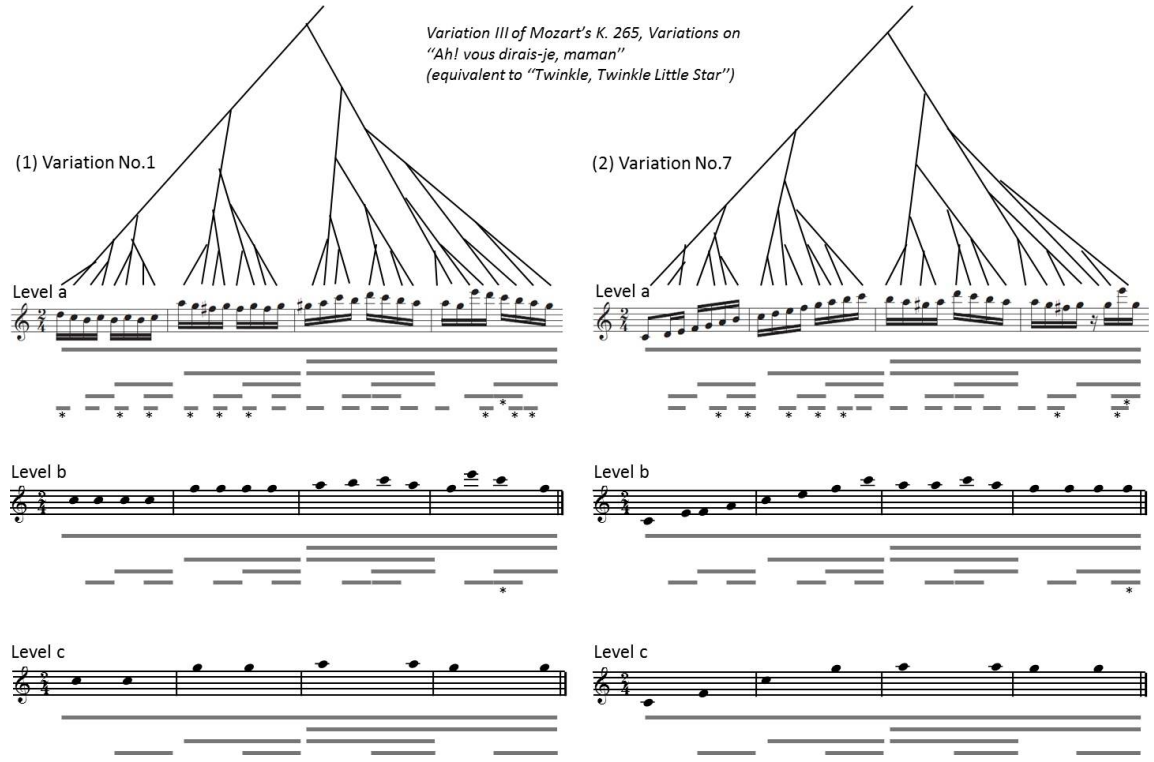


Fig. 5 Reduction of *Mozart: Ah! vous dirais-je, maman*.

B possess the same hierarchical configuration of maximal time-spans; that is, ignoring the difference in pitches of corresponding notes, A and C have the same tree structure. Then, we tentatively write $A\tilde{\sqcap}B$ for the reduction where A and B are reduced to be such A' and B', respectively, that $A' \approx B'$. This notation enables us to compare the distance of two variations. We write $(2a)$ for 'Level a of (2)' and so on, where (1) is the first variation and (2) is the seventh variation of K265/300e, as in Fig. 5. Then, $(1a)\tilde{\sqcap}(2a)$ is equal either to $(1c)$ or to $(2c)$. Remember here that we measure the length of a maximal time-span by the number of quarter notes. First, the difference between $(1b)\tilde{\sqcap}(2b)$ and $(1b)$ is one quaver, and thus $d((1b)\tilde{\sqcap}(2b),(1b)) = 1/2$. As $d((1b)\tilde{\sqcap}(2b),(2b)) = 1/2$, too, $d((1b),(2b)) = 1/2 + 1/2 = 1$. Next, the difference of $(1a)\tilde{\sqcap}(2a)$ and $(1a)$ is one quaver and nine semiquavers, that is, $d((1a)\tilde{\sqcap}(2a),(1a)) = 1/2 + 9/4 = 11/4$. As for the difference in $(1a)\tilde{\sqcap}(2a)$ and $(2a)$, there are one quaver and seven semiquavers, i.e., $d((1a)\tilde{\sqcap}(2a),(2a)) = 1/2 + 7/4 = 9/4$. Therefore, we obtain $d((1a),(2a)) = 11/4 + 9/4 = 5$, which is the structural distance between two variations.

5. Discussion: Distance and Similarity

In this section, we survey various definitions of distance and similarity, and try to position our measure among the existing criteria.

Since our procedure consists of a series of edits, such as deletion and insertion of branches, we may regard our distance belongs to a kind of *Levenshtein* distance. However, as every branch has its own maximal time-span, we can regard this length as the weight of *Earth Mover's Distance* (EMD) [*5], where the distance of each insertion/deletion operation is uniformly 1. Furthermore, since our target is a tree instead of a simple sequence of characters, we need to locate our method in the category of tree edit distance. As we have restricted our operations only on *reducible* branches, the distance by the sum of maximal time-spans is *1-degree tree edit distance*, i.e., editting operations are limited only to the leaves of the tree. As we have mentioned in Section 1, Rizo Valero [16] also employed the tree edit distance for similarity comparison. He compared the distance, however, in metrical trees, not in time-span trees. The metrical structure is essentially a binary or ternary tree and if two melodies have the same metrical pattern, the corresponding two metrical trees are the same, independently of pitches in melodies. In addition he customized the editting cost with the label-propagation and the pruning-level control. Thus, the direct numerical comparison with our method is difficult, but it is worth comparing the general behavior of two methods in larger database.

Next in this section, we consider the distance as a metric of similarity. As long as we stay in the lattice of reductions under *HSEC*, the distance strictly reflects the similarity. 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 '$\sharp_S(\sigma)$'. For example in the case of Fig. 1, $|\sigma_A \sqcap \sigma_B| = 2$ while $|\sigma_A \sqcup \sigma_B| = 4$, and thus, $sim(\sigma_A, \sigma_B) = 1/2$. 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. Again in the case of Fig. 1, $tms(\sigma_A \sqcap \sigma_B) = 1.5 + 3 = 4.5$ while $tms(\sigma_A \sqcup \sigma_B) = 0.5 + 0.5 + 1.5 + 3 = 5.5$, and thus $sim(\sigma_A, \sigma_B) = 9/11$. Similarly, we may make use of Simpson index with *tms* as follows:

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

In this case, $min(0.5 + 1.5 + 3, 0.5 + 1.5 + 3) = 5.0$ and $sim(\sigma_A, \sigma_B) = 4.5/5.0 = 9/10$. Such values of $9/11$ or $9/10$, in consideration of internal structure, seem rather convincing than that of $1/2$, which is by naïve set cardinality. However again, we need to scrutinize the general tendency of these similarities with the larger database including more complicated examples.

[*5] EMD becomes the the least amount of work to fill the holes with the multiple heaps of earth, measured by the sum of a mass times a distance.

6. Conclusions

In this paper, we relied on the strong reduction hypothesis of Generative Theory of Tonal Music [12] which could generate hierarchical tree structures, and presented the notion of distance between the trees. In addition, we applied the notion to the metric of similarity. In order to do this, we showed a feature structure to represent a time-span tree, employing *head* and *span* features. Thereafter, we regarded that a reduction was the loss of information, and the loss of an pitch event was quantified by its *maximal* time-span, within which the event is most salient. Then, the distance in a reduction path was defined as the sum of the length of such maximal time-spans, and could be the metric of similarity. We have shown several mathematical properties concerning the metric, including uniqueness of distance among any shortest paths as well as the triangle inequality.

Our main contribution in this paper is that we have presented a stable and consistent metric of similarity, which does rely on neither subjective nor context-dependent factor; it is mathematically sound since we can locate our index as 1-degree weighted tree edit distance.

Finally, we summarize open problems.

i) In Section 2, although we have introduced the representation of time-span tree in feature structure with *join* and *meet* operations, they can be applied properly only to those which were under *HSEC*. From a practical point of view, this condition is too restrictive. If we were to compute *join* and *meet* for two music pieces with different metrical structures, we need to seek for a more flexible mechanism to match *head*s and *span*s. The situation is the same as the comparison of two pitch events at *head* feature; we should tolerate the difference of on-time, duration, octave difference of pitch, and so on. For the purpose, we have to provide the flexible 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 loosen the condition of unification.

(ii) 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 perceptual 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 distance and similarity, we do not reflect such perceptual aspects.

(iii) The third problem concerns the footnote ∗1. After several reductions from an original music piece, we obtain a reduced

time-span tree together with remaining pitch events. However, since these remaining but salient pitch events possess only original durations, they cannot fill out the whole temporal length of saliency; unless we insert extra rests or extend their durations, we cannot obtain a proper music score, i.e., we cannot listen to the reduced time-span tree as music. Inversely, to listen to a time-span tree, we need to transform a time-span tree into a corresponding audible melody. We call the transformation *melodic rendering*. In general, such duration extension has multiple options, and thus, melodic rendering involves several possibilities. We have encountered the same issue in the footnote in Section 4; we have mentioned a common reduction from two music pieces which possesses the same hierarchical maximal time-spans, but each maximal time-span cannot inversely identify a pitch event with the proper onset/duration.

(iv) At last, we still need to recognize the fundamental problem of the original theory, that 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.

Acknowledgments 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*, Vol.17, pp.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*, Vol.12, No.1, pp.1–25 (Fall 1994).
[4] Downie, J.S., Byrd, D. and Crawford, T.: Ten Years of ISMIR: Reflections of Challenges and Opportunities, *Proc. ISMIR 2009*, pp.13–18 (2009).
[5] ESCOM: 2007 Discussion Forum 4A, Similarity Perception in Listening to Music, *Musicæ Scientiæ* (2007).
[6] ESCOM: 2009 Discussion Forum 4B, Musical Similarity, *Musicæ Scientiæ* (2009).
[7] Grachten, M., Arcos, J.-L. and de Mantaras, R.L.: Melody retrieval using the Implication/Realization model, 2005 MIREX, available from ⟨http://www.music-ir.org/evaluation/mirexresults/articles/similarity/grachten.pdf⟩.
[8] Hamanaka, M., Hirata, K. and Tojo, S.: Implementing "A Generative Theory of Tonal Music", *Journal of New Music Research*, Vol.35, No.4, pp.249–277 (2007).
[9] Hewlett, W.B. and Selfridge-Field, E.: Melodic Similarity, *Computing in Musicology*, Vol.11, The MIT Press (1998).
[10] Hirata, K. and Tojo, S.: Lattice for Musical Structures and Its Arithmetics, LNAI 4384, Selected Papers from JSAI 2006, Washio, T. et al. (Eds.), Springer-Verlag, pp.54–64 (2007).
[11] Hirata, K., Tojo, S. and Hamanaka, M.: Melodic Morphing Algorithm in Formalism, *Proc. 3rd International Conference, MCM 2011* (*LNAI 6726*), pp.338–341 (2011).
[12] Lerdahl, F. and Jackendoff, R.: *A Generative Theory of Tonal Music*, The MIT Press (1983).
[13] Marsden, A.: Generative Structural Representation of Tonal Music, *Journal of New Music Research*, Vol.34, No.4, pp.409–428 (2005).
[14] Ockelford, A.: Similarity relations between groups of notes: Music-theoretical and music-psychological perspectives, *Musicae Scientiae, Discussion Forum 4B, Musical Similarity*, pp.47–98 (2009).
[15] Pampalk, E.: Computational Models of Music Similarity and their Application in Music Information Retrieval, PhD Thesis, Vienna University of Technology (Mar. 2006).
[16] Rizo Valero, D.: Symbolic Music Comparison with Tree Data Structure, Ph.D. Thesis, Universitat d' Alacant, Departamento de Lenguajes y Sistemas Informatícos (2010).
[17] Sag, I.A. and Wasow, T.: Syntactic Theory: *A Formal Introduction*, CSLI Publications (1999).
[18] Schedl, M., Knees, P. and Böck, S.: Investigating the Similarity Space of Music Artists on the Micro-Blogosphere, *Proc. ISMIR 2011*, pp.323–328 (2011).
[19] Selfridge-Field, E.: Conceptual and Representational Issues in Melodic Comparison, *Computing in Musicology*, Vol.11, pp.3–64 (1998).
[20] Tan, P.N., Steinbach, M. and Kumar, V.: *Introduction to Data Mining*, Addison-Wesley (2005).
[21] Tojo, S. and Hirata, K.: Structural Similarity Based on Time-span Tree, *Proc. CMMR 2012*, pp.645–660 (2012).
[22] Volk, A. and Wiering, F.: Music Similarity, *ISMIR 2011 Tutorial on Musicology*, available from ⟨http://ismir2011.ismir.net/tutorials/ISMIR2011-Tutorial-Musicology.pdf⟩.
[23] Wiggins, G.A.: Semantic Gap?? Schematic Schmap!! Methodological Considerations in the Scientific Study of Music, *2009 11th IEEE International Symposium on Multimedia*, pp.477–482 (2009).
[24] Wiggins, G.A., Müllensiefen, D. and Pearce, M.T.: On the non-existence of music: Why music theory is a figment of the imagination, *Musicae Scientiae, Discussion Forum*, Vol.5, pp.231–255 (2010).
[25] Wilson, R.A. and Keil, F. (Eds.): *The MIT Encyclopedia of the Cognitive Sciences*, The MIT Press (May 1999).

Satoshi Tojo received degrees of Bachelor of Engineering, Master of Engineering, and Doctor of Engineering from University of Tokyo, Japan. He joined Mitsubishi Research Institute, Inc. (MRI) in 1983, and Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, as associate professor in 1995; professor from 2000. His research interest is in formal semantics of natural language, logic in artificial intelligence including knowledge and belief of artificial agents, and grammar acquisition, as well as linguistic model of music.

Keiji Hirata received degree of Doctor of Engineering from University of Tokyo in 1987. He joined NTT Basic Research Laboratories in 1987 (later changed to NTT Communication Science Laboratories) and Future University Hakodate as professor in 2011. His research interest includes music informatics (computational music theory), smart city (demand-responsive transportation), ICT support for depression, and video communication system.