Vol. 46 No. SIG 17(TOM 13) IPSJ Transactions on Mathematical Modeling and Its Applications Dec. 2005

Regular Paper

A Theoretical Analysis of Tree Edit Distance Measures

Tetsuji Kuboyama,[†] Kilho Shin[††] and Tetsuhiro Miyahara[†††]

The notion of the tree edit distance provides a unifying framework for measuring distance and finding approximate common patterns between two trees. A diversity of tree edit distance measures have been proposed to deal with tree related problems, such as minor containment, maximum common subtree isomorphism, maximum common embedded subtree, and alignment of trees. These classes of problems are characterized by the conditions of the tree mappings, which specify how to associate the nodes in one tree with the nodes in the other. In this paper, we study the declarative semantics of edit distance measures based on the tree mapping. In prior work, the edit distance measures have been not well-formalized. So the relationship among various algorithms based on the tree edit distance has hardly been studied. Our framework enables us to study the relationship. By using our framework, we reveal the declarative semantics of the alignment of trees, which has remained unknown in prior work.

1. Introduction

A tree is a mathematical abstraction which plays a significant role in the efficient organization of information. In particular, the problem of comparing tree structures emerges across a wide range of applications in computational biology [4),9),20)], image analysis [13),14)], pattern recognition [1)], natural language processing [17)], information extraction [5)] or wrapper induction [10)] from Web pages, and many others.

A *tree edit distance* method provides a general framework in comparing trees, measuring similarities, finding common tree patterns, and merging trees. The tree edit distance between two trees is basically defined as the minimum cost of edit operations to transform one tree into the other. The standard set of operations includes: (1) *relabeling* a node x; (2) *inserting* a new node x right under a node y (and moving a consecutive y's children and all their descendants under x); (3) *deleting* a node x (and contracting the edge between x and its parent).

Selkow [11)] and Tai [12)] first introduced edit distance measures for trees. Zhang and Shasha [22)] gave an efficient algorithm for the tree edit distance measure due to Tai [12)] as a natural generalization of string edit distance [3),18)]. These early works show that the study of the tree edit distance has a long history. These studies, however, did not have a firm theoretical foundation.

Many algorithms for calculating a tree edit distance are described and characterized by tree edit operations. A lot of those algorithms have been proposed independently in various fields, and the lack of a unifying framework has lead to confusion. That is, the relationship among various algorithms based on the tree edit distance has hardly been studied.

In this paper, we propose a new mathematical model as a unifying framework for describing tree edit distance measures. This model provides not only the operational semantics but also the declarative semantics of the tree edit distance. The declarative semantics enables us to study the relationship among existing tree edit distance measures. As a direct result of our model, we show the tree mapping condition of the alignment of trees, which has been unknown for the past decade.

Our model clarifies the meaning of similarity in the tree edit distance measures. Therefore, our model provides a useful framework for choosing appropriate algorithms of the tree edit distance in applying them to practical problems in various fields.

The rest of this paper is organized as follows: the next section describes the tree edit distance in an operational way, followed by reviewing existing distance measures in Section 3. Before taking up the main part, we give an overview of our contributions in Section 4. In Section 5, we propose a new formulation of the tree edit distance. In Section 6, we show the tree mapping condition of the alignment of trees by using

† Center for Collaborative Research, the University of Tokyo
†† Research Center for Advanced Science and Technology, the University of Tokyo
††† Faculty of Information Sciences, Hiroshima City University

our formulation. In Section 7, we conclude.

2. Tree Edit Distance

In this section, we review the tree edit distance.

Trees we consider in this paper are labeled rooted trees, in which each node is labeled from a finite alphabet Σ. We denote by $r(T)$ the root of a tree T, and by $T(x)$ the *maximum subtree* of T rooted at a node x. An *ancestor* of a node is recursively defined as follows: an ancestor of a node is either the node itself, or an ancestor of the parent of the node. We denote by $x \leq y$ that a node y is an ancestor of a node x, by $\mathrm{lca}(X)$ the *least* (or *nearest*) *common ancestor* of all nodes in a set of nodes X, and by $x \frown y$ the least common ancestor of x and y.

An *ordered tree* is a tree in which the left-to-right order among siblings is given. In an ordered tree, we say that a node x is *to the left* of a node y if $x \frown y$ is a proper ancestor of both x and y, and the child of $x \frown y$ on the path to x is to the left of the child of $x \frown y$ on the path to y. An *unordered tree* is a tree with no order among siblings. We refer to unordered trees simply as trees unless otherwise stated.

2.1 Operational Definition

The tree edit distance between two trees is defined as the minimum cost of elementary edit operations to transform one tree into the other. In general, the following edit operations are used [12),22)].

Let l be a labeling function which assigns a label from a set $\Sigma = \{a, b, c, \ldots\}$ to each node. Let λ denote the unique null symbol not in Σ. Let d be a cost function $d : (\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\}) \to \mathbb{N}$, where $\mathbb{N}$ is the set of non-negative integers.

Definition 1. An *edit operation* on a tree T is any of the following three operations:

relabeling of the label of a node x in T with the label of a new node y in T; the cost is denoted by $d(l(x) \to l(y))$,

insertion of a new node x into T as a child of a node y in T, moving a subset (a consecutive subsequence in the case of ordered trees) of y's children and their descendants right under the new node x; note that this is the complementary operation of deletion; the cost is denoted by $d(\lambda \to l(x))$, and

deletion of a non-root node x from T, moving all children of x right under the parent of x; the cost is denoted by $d(l(x) \to \lambda)$.

We assume, without loss of generality, that the root of a tree is not to be deleted or inserted. We refer to each cost factor as its edit operation when there is no confusion; i.e., $\alpha \to \beta$, $\lambda \to \beta$, and $\alpha \to \lambda$ for α, $\beta \in \Sigma$ are referred to as the edit operations of relabeling, insertion, and deletion, respectively. **Figure 1** illustrates the three edit operations.

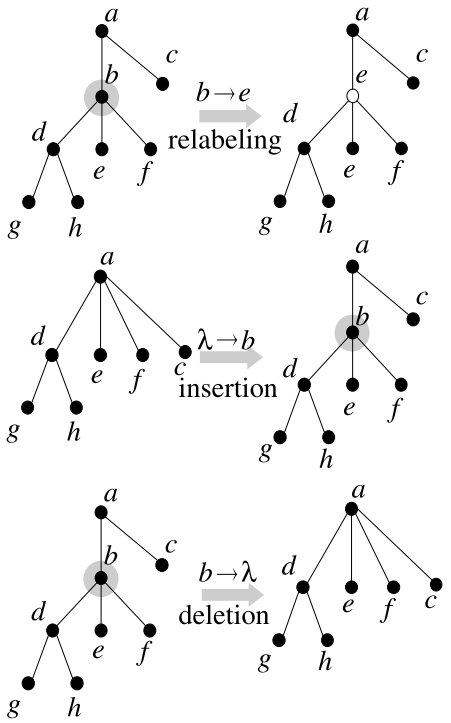


Fig. 1 Three elementary edit operations: (1) Relabeling of the node label a to b. (2) Inserting the node labeled with b. (3) Deleting the node labeled with b.

The cost function d is defined to be a metric; i.e., for any α, β, $\gamma \in \Sigma \cup \{\lambda\}$,

(1) $d(\alpha \to \beta) \geq 0$, $d(\alpha \to \alpha) = 0$,
(2) $d(\alpha \to \beta) = d(\beta \to \alpha)$, and
(3) $d(\alpha \to \gamma) \leq d(\alpha \to \beta) + d(\beta \to \gamma)$.

If a sequence of edit operations E transforms a tree T into a tree U, there exists a sequence of trees $\langle T_0, \ldots, T_n \rangle$ $(n \geq 1)$ such that $T_0 = T$, $T_n = U$, and the i-th edit operation $e_i = (\alpha_i \to \beta_i)$ transforms T_{i-1} into T_i for $i \in \{1, \ldots, n\}$. The cost function d for an edit operation is generalized to that for a sequences of edit operations $E = \langle e_1, \ldots, e_n \rangle$ by letting

$$d(E) = \sum_{i=1}^{n} d(e_i).$$

Let $\mathcal{E}$ be the set of all possible sequences of edit operations to transform T into U. The edit distance δ between two trees T and U is

defined [12] as

$$\delta(T,U) = \min_{E\in\mathcal{E}}\{d(E)\}.$$

2.2 Declarative Definition and Tree Mapping

The effect of a sequence of edit operations is reduced to a structure called *tree mapping* [12], which is a comparable notion to *trace* [18] in string edit distance. We also refer to the *tree mapping* as *mapping* if the context is clear. A *tree mapping* depicts node-to-node correspondences between two trees according to the structural similarity, or shows how nodes in one tree are preserved after transformed to the other.

Definition 2. A *tree mapping* from a tree T to a tree U is a set $M \subseteq V(T) \times V(U)$ such that, for all (x_1, x_2), $(y_1, y_2) \in M$,

(1) $x_1 \le y_1 \Leftrightarrow x_2 \le y_2$, and

(2) (only for ordered trees) x_1 is to the left of $y_1 \Leftrightarrow x_2$ is to the left of y_2.

For example, **Fig. 2** shows a tree mapping, in which all nodes connected with dashed lines preserve the tree mapping condition of Definition 2. Note that the original definition of the tree mapping by Tai [12] includes the condition $x_1 = y_1 \Leftrightarrow x_2 = y_2$ for all (x_1, x_2), $(y_1, y_2) \in M$. We omit this condition since it is implied by the condition (1). For a tree mapping M from T to U, we define:

$$M_D = V(T) \setminus \{x | (x,y) \in M\}, and$$
$$M_I = V(U) \setminus \{y | (x,y) \in M\}$$

The cost of M is defined as

$$\begin{aligned} d(M) = &\textstyle\sum_{(x,y)\in M} d(l(x) \to l(y)) \\ &+ \textstyle\sum_{x\in M_D} d(l(x) \to \lambda) \\ &+ \textstyle\sum_{y\in M_I} d(\lambda \to l(y)). \end{aligned}$$

The following theorem due to Tai [12] shows that the edit distance δ between T and U is given in two ways.

Theorem 1 (Tai [12])**.** Let $\mathcal{M}$ be the set of all possible tree mappings from T to U.

$$\delta(T,U) = \min_{E\in\mathcal{E}}\{d(E)\} = \min_{M\in\mathcal{M}}\{d(M)\}.$$

This theorem plays the role of a bridge between an operational definition and a declarative definition for the tree edit distance.

3. Tai Distance and Alignment of Trees

In this section, we give a cursory review of related work.

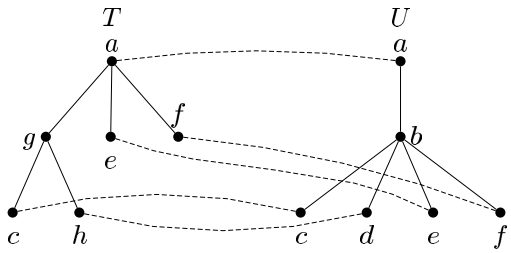


Fig. 2 An example of a tree mapping: relabeling the node label h with d, deleting the node labeled with g, and inserting the node labeled with b.

3.1 Tai Distance

The tree edit distance stated in the previous section is considered to be the most general form of distance measures. We refer to this measure and the tree mapping as the *Tai distance* and *Tai mapping* respectively.

For ordered trees, a polynomial-time algorithm for computing Tai distance and Tai mapping was given by Zhang and Shasha [22]. For similar ordered trees, an efficient algorithm was proposed [15]. As for unordered trees, this problem is known to be NP-complete [23] (in fact MAX-SNP hard [21]), even for binary trees with an alphabet of two size for node labels.

3.2 Alignment of Trees

The alignment of trees was introduced by Jiang, et al. [7] as a natural extension of alignment of strings. For ordered trees, a polynomial-time algorithm was introduced by Jiang et al. [7]. For unordered trees, this problem is known to be MAX-SNP hard [7]. An efficient algorithm for similar trees was proposed for ordered trees [6], and for unordered trees [2]. The definition of the alignment of trees has been given in an operational way [7),16),19] as follows.

Definition 3 (Jiang, et al. 1995 [7])**.** Let T and U be two trees. An alignment of T and U is obtained by first inserting nodes labeled with λ into T and U such that the two resulting trees T' and U' have the same structure, i.e., they are identical if the labels are ignored, and then *overlaying* T' on U'. The cost of the alignment is the sum of the costs of all overlaid pairs of labels, where the cost of a pair of labels is defined by a cost function $d : (\Sigma \cup \{\lambda\}) \times (\Sigma \cup \{\lambda\}) \to \mathbb{N}$. The *alignment distance* is defined as the minimum cost of the alignment.

An example of the alignment of trees is shown in **Fig. 3**.

It is well-known, in strings, that the alignment distance and the edit distance are two equivalent notions [3]. This equivalence, how-

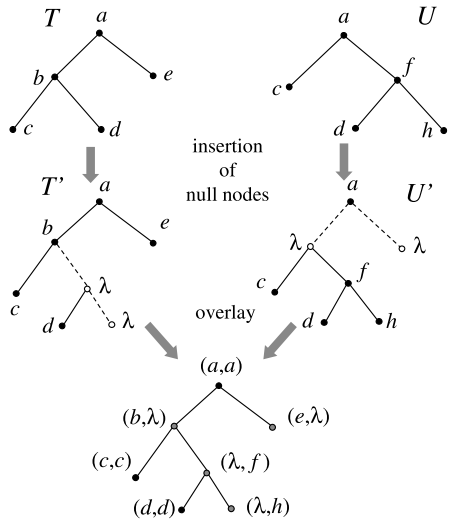


Fig. 3 An alignment of trees between T and U.

ever, does not hold for trees. Hence, the tree mapping condition for the alignment of trees is different from Tai mapping. In fact, the tree mapping condition for the alignment of trees has been unknown in spite of its significance.

4. Our Contributions

Before moving ahead with the formulation of the tree edit distance, we mention our contributions and the overview of our model.

Our contributions are as the followings:

- We give the tree mapping condition for alignment of tree, which has been unknown in prior work. This implies that we obtain the declarative definition for alignment of trees.
- This implies that finding a common subtree pattern between two trees under the tree mapping condition is equivalent to finding a common supertree pattern between two trees in terms of minor containment [8].
- Both the edit distance and the alignment distance have been introduced as natural generalizations of those for strings. Although these two measures are the same originally in strings, these are not the same in trees. We show the confluent point between the tree edit distance and the alignment distance.

The rest of paper is devoted to prove the main theorem.

In our formulation, we first introduce a general mapping between trees called *tree homomorphism*. Starting with the notion of the tree homomorphism, we tighten the mapping gradually to fit in existing edit operations. **Figure 4** shows the schematic view of our model.

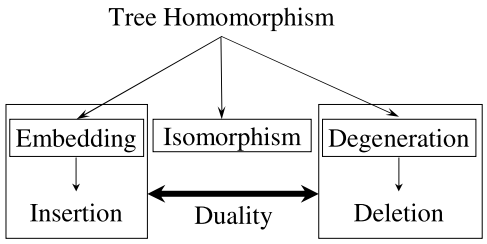


Fig. 4 The schematic view of our model.

5. Formulation of Tree Edit Distance

5.1 Rooted Trees

We formulate trees as ordered sets of nodes. In the course of the formulation, we redefine a few of the notions given in Section 2.

We adopt a standard notation $<$ to denote a *strict partial order*, that is, for a non-empty finite set V,

(1) $\forall x, y, z \in V\ [x < y\ \wedge\ y < z \Rightarrow x < z]$,
(2) $\forall x \in V\ [x \not< x]$.

We denote by $x \leq y$ that $x < y$ or $x = y$ for all $x, y \in V$. We say that two elements $x, y \in V$ are *comparable* if $x < y$, $x = y$ or $y < x$ holds.

Definition 4. A *rooted tree* $T = (V, <)$ is a nonempty, finite, and strict partially ordered set with the maximum element $r(T) \in V$ called the *root*, and such that $\{y \in V | x \leq y\}$ is a totally ordered set for every $x \in V$.

Remark. A rooted tree is called an *ordered tree* if and only if another partial order $\prec$ is defined over V and the followings hold.

(1) $x \prec y \vee y \prec x \Leftrightarrow x \not\leq y \wedge y \not\leq x$,
(2) $x \leq x' \wedge y \leq y' \wedge x' \prec y' \Rightarrow x \prec y$.

Although all the definitions, propositions, lemmas and theorems stated in this paper also hold for the ordered tree with no or slight modification, this paper does not mention all of them.

We call the elements of V the *nodes* of T, and denote the set of all nodes in T by $V(T)$. We define the set of *edges* in T by $E(T) = \{(x, y) \in V(T) \times V(T) | (x < y) \wedge \nexists z \in V(T)$ such that $x < z < y\}$. An *ancestor* of x is a node y such that $x \leq y$. In particular, if $x < y$, then y is called a *proper ancestor*. The *parent* of a node x is the minimum node of the proper ancestors of x, and denoted by $p(x)$. The *children* of a node x is the set of nodes such that $\{y | (y, x) \in E(T)\}$, and is denoted by $\mathrm{ch}(x)$. We call the elements of $\mathrm{ch}(x)$ a child of x. A *leaf* of a tree T is a minimal node in T.

We redefine the notion of the least common ancestor as follows.

Definition 5. For an arbitrary rooted tree $T = (V, <)$, a *common ancestor* of a set of nodes $V' \subseteq V$ is an element $x \in V$ such that $y \leq x$ for all $y \in V'$. A common ancestor x of V' is the *least common ancestor* of V' if, for any common ancestor x' of V', $x \leq x'$ holds. We denote the least common ancestor of V' by $\mathrm{lca}(V')$, and $\mathrm{lca}(\{x, y\})$ by $x \smile y$.

Lemma 2. The following properties hold in terms of the least common ancestor:

(1) $x \smile x = x$,
(2) $x \smile y = y \smile x$,
(3) $(x \smile y) \smile z = x \smile (y \smile z)$,
(4) $x \leq y \Leftrightarrow x \smile y = y$,
(5) $x \smile y < x \smile z \Rightarrow y \smile z = x \smile z$,
(6) $x \smile y = x \smile z \Rightarrow y \smile z \leq x \smile y$.

Proof. (1) to (4) are all easy to prove.
(5): Since $y < x \smile z$ by the premise, we have $y \smile z \leq x \smile z$. On the other hand, if $x \smile y < y \smile z$, then we have $x < y \smile z$, therefore, $y \smile z \geq x \smile z$. If $y \smile z \leq x \smile y$, then $z \leq x \smile y$, therefore, $x \smile z \leq x \smile y$, as is contradictory to the premise.
(6): The assertion immediately follows $x \leq x \smile z$ and $y \leq y \smile z$. □

Corollary 3. For any three nodes x, y, z, any of the following properties holds:

(1) $x \smile y < x \smile z$, and $x \smile z = y \smile z$,
(2) $x \smile y = x \smile z$, and $y \smile z \leq x \smile z$, or
(3) $x \smile y > x \smile z$, and $x \smile y = y \smile z$.

Proof. It follows straightforwardly from Lemma 2–(5), and (6). □

5.2 Formulation of Edit Operations

This section is preliminary towards formulating the edit operations in the tree edit distance and the alignment of trees.

5.2.1 Tree Homomorphism

We first introduce the notion of tree homomorphism to represent structural similarities between trees.

Definition 6 (Tree Homomorphism)**.** Let T and U be two trees. A *tree homomorphism* from T to U is a set-theoretic mapping $\varphi : V(T) \to V(U)$ such that $\varphi(x) \leq \varphi(y)$ if $x < y$ for all x, $y \in V(T)$.

When a mapping $\varphi : V(T) \to V(U)$ yields a tree homomorphism, we simply denoted it by $\varphi : T \to U$.

Remark. To extend the definition of a tree homomorphism to *ordered* trees, it suffices to add the condition:

$\varphi(x) \not\prec \varphi(y)$ if $x \prec y$ for all $x, y \in V(T)$.

Definition 7. For a tree homomorphism $\varphi : T \to U$, the *image* of φ is the tree $\Im(\varphi) = (V(\Im(\varphi), <_{\Im(\varphi)}))$ such that

(1) $V(\Im(\varphi)) = \{x \in V(U) | \exists (y \in V(T))[x \leq \varphi(r(T))]\}$, and
(2) $\forall x, y \in V(\Im(\varphi))[x <_{\Im(\varphi)} y \Leftrightarrow x < y]$.

Remark. Let U be an ordered tree with a sibling partial order $\prec$. A sibling partial order $\prec_{\Im(\varphi)}$ for $\Im(\varphi)$ is defined by $\forall x, y \in V(\Im(\varphi))[x \prec_{\Im(\varphi)} y \Leftrightarrow x \prec y]$ as well.

It is obvious from these definitions that a composition of tree homomorphisms is a tree homomorphism.

Definition 8 (Isomorphism)**.** Let T and U be two trees. An *isomorphism* from T to U is a bijection φ from $V(T)$ to $V(U)$ such that (x, y) is an edge of T if and only if $(\varphi(x), \varphi(y))$ is an edge of T.

Proposition 4. An isomorphism φ and its inverse φ^{-1} are both tree homomorphisms.

Proposition 5. Let T and U be two trees. Suppose that a tree homomorphism φ is a bijection from $V(T)$ to $V(U)$. Then the following properties are equivalent:

(1) φ is an isomorphism, and
(2) $x < y$ if $\varphi(x) < \varphi(y)$ for all x, y in $V(T)$.

Proof. (1)⇒(2): This is straightforward from Definition 8.
(2)⇒(1): Let η be a mapping $\eta : (x, y) \mapsto (\varphi(x), \varphi(y))$ for all $(x, y) \in E(T)$. We show that the mapping $\eta : E(T) \to E(U)$ is well-defined, and bijective, i.e. φ is an isomorphism, under the condition (2). For any edge $(x, y) \in E(T)$, let z be a node in $V(U)$ such that $\varphi(x) \leq z < \varphi(y)$. Then we have $x \leq \varphi^{-1}(z) < y$. It follows that $x = \varphi^{-1}(z)$ since (x, y) is an edge of T. Therefore we have $z = \varphi(x)$. So $(\varphi(x), \varphi(y))$ is an edge of U, and hence η is well-defined. Since the condition (2) implies that φ^{-1} is also a tree homomorphism, we also have η^{-1} is well-defined. Therefore, η is bijective. □

Remark. For a bijective tree homomorphism of ordered trees, the following properties are equivalent:

(1) φ is an isomorphism,
(2) $x < y$ if $\varphi(x) < \varphi(y)$ for all x, y in $V(T)$,
(3) $\varphi(x) \prec \varphi(y)$ if $x \prec y$ for all x, y in $V(T)$.

Proposition 6. Let T and U be two trees. For any tree homomorphism $\varphi : T \to U$, and x, $y \in V(T)$, it holds that $\varphi(x) \smile \varphi(y) \leq \varphi(x \smile y)$.

Proof. From the facts that $x \leq x \smile y$ and

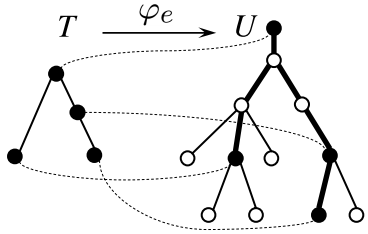


Fig. 5 An example of an embedding φ_e.

$y \leq x \smile y$, we obtain $\varphi(x) \leq \varphi(x \smile y)$ and $\varphi(y) \leq \varphi(x \smile y)$, respectively. Hence $\varphi(x) \smile \varphi(y) \leq \varphi(x \smile y)$. □

Even if $x \smile y < x \smile z$ for x, y, $z \in V(S)$, this proposition implies that, for a homomorphism $\varphi : T \to U$, any of the following three conditions may hold:

(1) $\varphi(x) \smile \varphi(y) < \varphi(x) \smile \varphi(z)$,
(2) $\varphi(x) \smile \varphi(y) = \varphi(x) \smile \varphi(z)$, and
(3) $\varphi(x) \smile \varphi(y) > \varphi(x) \smile \varphi(z)$.

5.2.2 Embedding

We introduce an important subclass of the tree homomorphism, called *embedding*, which is a mapping from a tree T to a tree U such that it preserves the Tai mapping condition, and $V(T) \subseteq V(U)$.

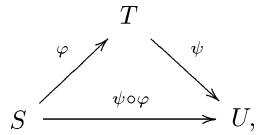
Definition 9 (Embedding)**.** Let T and U be two trees. A tree homomorphism $\varphi : T \to U$ is an *embedding* if the following conditions are satisfied:

(1) φ is injective, and
(2) $x < y$ if $\varphi(x) < \varphi(y)$ for all x, $y \in V(T)$.

We refer to $\mathrm{red}(\varphi) = |V(\Im(\varphi)) \setminus \varphi(V(T))|$ as the *redundancy* of $\varphi : T \to U$.

Figure 5 shows an example of embeddings.

Proposition 7. Let S, T and U be trees. Suppose that $\varphi : S \to T$ and $\psi : T \to U$ are tree homomorphisms,

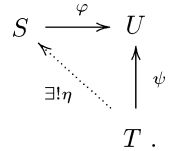


then the following properties hold:

(1) If φ and $\psi|_{\Im(\varphi)}$ are embeddings, $\psi \circ \varphi$ is also an embedding. Moreover, $\mathrm{red}(\psi \circ \varphi) = \mathrm{red}(\varphi) + \mathrm{red}(\psi|_{\Im(\varphi)})$ holds.
(2) If $\psi \circ \varphi$ is an embedding, φ is also an embedding.

Proof. (1) $\psi \circ \varphi$ is injective. By Definition 9, for any x, $y \in V(S)$, if $\psi(\varphi(x)) < \psi(\varphi(y))$, then $\varphi(x) < \varphi(y)$, and if $\varphi(x) < \varphi(y)$, then $x < y$. Therefore $\psi \circ \varphi$ is an embedding. Also, $\mathrm{red}(\psi \circ \varphi) = |V(\Im(\psi \circ \varphi))| - |V(S)| = (|V(\Im(\psi|_{\Im(\varphi)}))| - |V(\Im(\varphi))|) + (|V(\Im(\varphi))| - |V(S)|) = \mathrm{red}(\psi|_{\Im(\varphi)}) + \mathrm{red}(\varphi)$. (2) It is obvious that φ is injective. It suffices to show that $x < y$ holds assuming $\varphi(x) < \varphi(y)$. Since ψ is a tree homomorphism and $\psi \circ \varphi$ is injective, we have $\psi(\varphi(x)) < \psi(\varphi(y))$. Therefore, $x < y$ since $\psi \circ \varphi$ is an embedding. □

Proposition 8. Let S, T, and U be three trees. For an embedding $\varphi : S \to U$, and a tree homomorphism $\psi : T \to U$, if $\psi(V(T)) \subseteq \varphi(V(S))$, then there exists a unique tree homomorphism $\eta : T \to S$ such that $\psi = \varphi \circ \eta$;



Proof. Let us choose the unique mapping $\eta : V(T) \to V(S)$ so that $\psi = \varphi \circ \eta$. We show that η is a tree homomorphism. Let x and y be two nodes in $V(T)$. From the definition of η, it follows that $\psi(x) = \varphi(\eta(x))$ and $\psi(y) = \varphi(\eta(y))$. Assume that $x < y$. Then we have $\psi(x) = \psi(y)$ or $\psi(x) < \psi(y)$. Since φ is an embedding, if $\varphi(\eta(x)) = \varphi(\eta(y))$, then $\eta(x) = \eta(y)$, and if $\varphi(\eta(x)) < \varphi(\eta(y))$, then $\eta(x) < \eta(y)$. Hence $\eta(x) \leq \eta(y)$. □

An embedding is uniquely determined except for the isomorphism as shown in the following.

Corollary 9. Let S, T, and U be three trees. Let $\varphi : S \to U$ and $\psi : T \to U$ be two embeddings with $\varphi(V(S)) = \psi(V(T))$. There exists a unique isomorphism $\eta : T \to S$ such that $\psi = \varphi \circ \eta$.

Proposition 10. For an embedding $\varphi : S \to T$ and x, $y \in V(S)$, the minimum node $\varphi(z)$ in T such that $\varphi(x) \smile \varphi(y) < \varphi(z)$ is identical to $\varphi(x \smile y)$. Furthermore, the following conditions are equivalent.

(1) $\varphi(x) \smile \varphi(y) < \varphi(x \smile y)$.
(2) $\varphi(x) \smile \varphi(y) \notin \varphi(V(S))$.

Proof. Suppose that $\varphi(x) \smile \varphi(y) \leq \varphi(z)$. By Definition 9, we have $x \smile y \leq z$. Hence $\varphi(x \smile y) \leq \varphi(z)$. This implies that $\varphi(x \smile y)$ is the minimum $\varphi(z)$ such that $\varphi(x) \smile \varphi(y) \leq \varphi(z)$. The equivalence between (1) and (2) immediately follows this property. □

Corollary 11. For an embedding $\varphi : T \to U$, if $x \smile y < x \smile z$, then $\varphi(x) \smile \varphi(y) < \varphi(x) \smile \varphi(z)$.

Proof. $\varphi(x) \frown \varphi(y) \leq \varphi(x \frown y) < \varphi(x \frown z)$ holds. $\varphi(x \frown y)$ and $\varphi(x) \frown \varphi(z)$ are comparable since both are ancestors of $\varphi(x)$. If $\varphi(x) \frown \varphi(z) = \varphi(x \frown z)$, then there is nothing to prove. If $\varphi(x) \frown \varphi(z) \leq w < \varphi(x \frown z)$, then $w \notin \varphi(V(T))$ by Proposition 10. Therefore, we have $\varphi(x \frown y) < \varphi(x) \cup \varphi(z)$. □

5.2.3 Logical Expressions and Embeddings

We introduce a useful expression for filtering nodes of trees. For a tree T, let $\pi(\mathbf{x}) : V(T) \to \{\mathbf{t}, \mathbf{f}\}$ denote a unary predicate with a predicate variable $\mathbf{x}$.

Definition 10 (logical expressions)**.** $T[\pi(\mathbf{x})] = (V[\pi(\mathbf{x})], \leq_\pi)$ as follows:

(1) $V[\pi(\mathbf{x})] = \{x | x \in V(T) \text{ and } \pi(x) = \mathbf{t}\}$,

(2) $x \leq_\pi y$ if and only if $x \leq y$ for all x, $y \in V[\pi(\mathbf{x})]$.

For example, $T[\mathbf{x} \leq x]$ is equivalent to $T(x)$. Note that $T[\pi(\mathbf{x})]$ is not necessarily a tree since it may not have a root.

By $\mathbf{E}_{\pi(\mathbf{x})}$, we denote a natural inclusion $\mathbf{E}_{\pi(\mathbf{x})} : V(T[\pi(\mathbf{x})]) \to V(T)$.

Proposition 12. For $x, y \in V(T[\pi(\mathbf{x})])$, $x < y$ if and only if $\mathbf{E}_{\pi(\mathbf{x})}(x) < \mathbf{E}_{\pi(\mathbf{x})}(y)$.

Corollary 13. If $T[\pi(\mathbf{x})]$ is a tree, $\mathbf{E}_{\pi(\mathbf{x})}$ is an embedding with $\mathrm{red}(\mathbf{E}_{\pi(\mathbf{x})}) = |\{x \in V(T) | \pi(x) = \mathbf{f}\}|$.

5.2.4 Insertion

Now we are ready to give a declarative definition of the insertion operation.

Definition 11 (Insertion)**.** Let T and U be two trees. An embedding $\varphi : T \to U$ with $\mathrm{red}(\varphi) = 1$ is called an *insertion*. In particular, if $\varphi(V(T)) = V(U) \setminus \{x\}$ for $x \neq r(U)$, an insertion φ is called an x-insertion.

Proposition 14. For an arbitrary $x \in T$ such that $x \neq r(T)$, there exists an x-insertion φ into T. Furthermore, an x-insertion is unique up to an isomorphism.

Proof. Defining $\pi(\mathbf{x}) = (\mathbf{x} \neq x)$, $\mathbf{E}_{\pi(\mathbf{x})} : T[\pi(\mathbf{x})] \to T$ is an x-insertion into T by Proposition 12. By Corollary 9, an x-insertion is uniquely determined up to an isomorphism. □

We denote the unique x-insertion by I_x.

The following proposition shows that Definition 11 of the insertion is equivalent to the operational definition of the insertion.

Proposition 15. Let T and U be two trees. For $x \in V(U)$, $I_x : T \to U$ satisfies the following properties (See **Fig. 6**):

(1) for any $y \in \mathrm{ch}(x)$, $I_x : T(I_x^{-1}(y)) \to U(y)$ is an isomorphism, and

(2) $I_x : T[\bigwedge_{y \in \mathrm{ch}(x)} \mathbf{x} \not\leq I_x^{-1}(y)] \to U[\mathbf{x} \not\leq x]$ is an isomorphism.

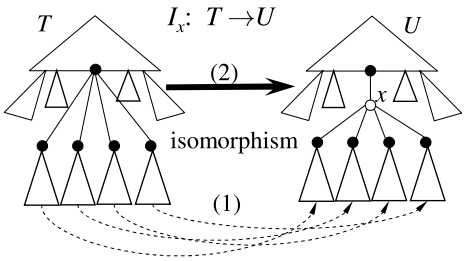


Fig. 6 Relationship between an insertion mapping and an operational definition.

Proof. Without loss of generality, we may assume that $T = U[\mathbf{x} \neq x]$. It follows from Proposition 12 that
$U[\mathbf{x} \neq x \wedge \mathbf{x} \leq y] = U[\mathbf{x} \leq y]$, and
$U[\mathbf{x} \neq x \wedge \bigwedge_{y \in \mathrm{ch}(x)} \mathbf{x} \not\leq I_x^{-1}(y)] = U[\mathbf{x} \not\leq x]$.
Hence, we obtain the assertions. □

Theorem 16 (Decomposition of embedding)**.** Let φ be an embedding from T to U with $V(\Im(\varphi)) \setminus \varphi(V(T)) = \{x_1, \ldots, x_n\}$. There exist a sequence of trees T_0, T_1, …, T_n, and a sequence of insertions $\varphi_i : T_i \to T_{i-1}$ ($i \in \{1, \ldots, n\}$) such that

(1) $T_0 = U$,

(2) $T_n = T$,

(3) $\varphi_1 \circ \cdots \circ \varphi_i(V(T_i)) = V(\Im(\varphi)) \setminus \{x_1, \ldots, x_i\}$, and

(4) $\varphi = \varphi_1 \circ \cdots \circ \varphi_n$;

$$\begin{array}{ccccccccc} T_n & \xrightarrow{\varphi_n} & T_{n-1} & \xrightarrow{\varphi_{n-1}} & \cdots & \xrightarrow{\varphi_2} & T_1 & \xrightarrow{\varphi_1} & T_0 \\ \| & {\scriptstyle I_{x_n}} & & {\scriptstyle I_{x_{n-1}}} & & {\scriptstyle I_{x_2}} & & {\scriptstyle I_{x_1}} & \| \\ T & & & & \xrightarrow{\qquad\qquad\varphi\qquad\qquad} & & & & U\,. \end{array}$$

Proof. We apply induction on $n = \mathrm{red}(\varphi)$. For $n = 1$, φ is an insertion by definition. Now assume that $n \geq 2$. Let φ_1 be the x_1-insertion into $\Im(\varphi)$. Note that φ_1 can be naturally regarded as an insertion from T_1 to U (Proposition 7). By Proposition 8, there exists $\varphi' : T \to T_1$ such that $\varphi = \varphi_1 \circ \varphi'$. φ' is an embedding by Proposition 7. Furthermore, since φ_1 is injective and $V(\Im(\varphi_1)) = V(\Im(\varphi))$, $V(\Im(\varphi')) = V(T_1)$, and therefore $\mathrm{red}(\varphi') = n - 1$ by Proposition 7. By the induction hypothesis, there exist a sequence of trees T_2, T_3, …, T_n, and a sequence of insertions $\varphi_i : T_i \to T_{i-1}$ ($i \in \{2, \ldots, n\}$) such that

(1) $T_n = T$,

(2) $\varphi_2 \circ \cdots \circ \varphi_i(V(T_i)) =$

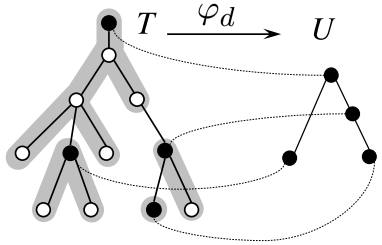


Fig. 7 An example of a degeneration.

$V(T_1) \setminus \{\varphi_1^{-1}(x_2), \ldots, \varphi_1^{-1}(x_i)\}$, and

(3) $\varphi' = \varphi_2 \circ \cdots \circ \varphi_n$.

Hence, the assertion of this theorem is proved as follows:

$$\begin{aligned}&\varphi_1 \circ \cdots \circ \varphi_i(V(T_i))\\ &\quad = \varphi_1(V(T_1) \setminus \{\varphi_1^{-1}(x_2), \ldots, \varphi_1^{-1}(x_i)\})\\ &\quad = \varphi_1(V(T_1)) \setminus \{x_2, \ldots, x_i\}\\ &\quad = V(\Im(\varphi)) \setminus \{x_1, x_2, \ldots, x_i\}\end{aligned}$$

□

5.2.5 Degeneration

We introduce the complementary notion of embedding, called degeneration as follows.

Definition 12 (Degeneration)**.** Let T and U be two trees. A homomorphism $\varphi : T \to U$ is a *degeneration* if the following conditions are satisfied:

(1) φ is surjective onto $V(\Im(\varphi))$,

(2) for all $x, y \in V(T)$, if $\varphi(x) = \varphi(y)$, then $\varphi(x \smile y) = \varphi(x)$, and

(3) for all $x, y \in V(T)$, if $\varphi(x) < \varphi(y)$, then there exists $z \in V(T)$ such that $\varphi(y) = \varphi(z)$ and $x < z$.

We refer to $\mathrm{Dup}(\varphi) = \{x \in V(T) | \varphi(x) = \varphi(p(x))\}$ as the *duplication* of the degeneration $\varphi : T \to U$.

Figure 7 shows an example of embeddings.

Proposition 17. Let T and U be two trees. For any degeneration $\varphi : T \to U$, the following properties hold:

(1) $(\varphi(x), \varphi(y)) \in E(U)$ or $\varphi(x) = \varphi(y)$ if $(x, y) \in E(T)$,

(2) for all $x \in \varphi(V(T))$, $\mathrm{lca}(\varphi^{-1}(x)) \in \varphi^{-1}(x)$, and

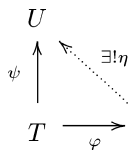
(3) $\mathrm{Dup}(\varphi) = \bigcup_{x \in \varphi(V(U))} \{\varphi^{-1}(x) \setminus \mathrm{lca}(\varphi^{-1}(x))\}$.

Proof. (1): Assume $\varphi(x) < \varphi(z) \leq \varphi(y)$. By Definition 12, we may assume $x < z$. Since $(x, y) \in E(T)$, we have $y \leq z$ and therefore $\varphi(y) = \varphi(z)$. This implies that $(\varphi(x), \varphi(y)) \in E(U)$

(2): Choose $y \in \varphi^{-1}(x)$ so that $y \not< z$ for all $z \in \varphi^{-1}(x)$. Let z be an arbitrary node of $\varphi^{-1}(x)$. By Definition 12, we have $\varphi(y \smile z) = x$, and hence $y \smile z = y$. We conclude $y = \mathrm{lca}(\varphi^{-1}(x))$ since $z \leq y$.

(3): Assume that $\varphi(x) = y$ and $x < \mathrm{lca}(\varphi^{-1}(y))$. Then, for all z such that $x < z \leq \mathrm{lca}(\varphi^{-1}(y))$, $\varphi(z) = y$. It follows that $\mathrm{Dup}(\varphi) \supseteq \bigcup_{x \in \varphi(V(T))} \varphi^{-1}(x) \setminus \mathrm{lca}(\varphi^{-1}(x))$. On the other hand, if $\varphi(x) = y$, then $x \leq \mathrm{lca}(\varphi^{-1}(y))$. Therefore, $\mathrm{lca}(\varphi^{-1}(y)) \notin \mathrm{Dup}(\varphi)$. □

Proposition 18. Let S, T, and U be three trees. For a degeneration $\varphi : T \to S$, let $\psi : T \to U$ be a homomorphism such that if $\varphi(x) = \varphi(y)$, then $\psi(x) = \psi(y)$. There exists a unique homomorphism $\eta : \Im(\varphi) \to U$ such that $\psi = \eta \circ \varphi$;



Proof. Without loss of generality, we may assume that $\Im(\varphi) = S$. By the premise of the theorem, we then have the unique set-theoretic mapping η such that $\eta \circ \varphi = \psi$. $x = \varphi(x') \in V(S)$. We show that η is a homomorphism. Let $x = \varphi(x')$ and $y = \varphi(y')$, and $x < y$. By Definition 12, we may assume $x' < y'$. Therefore, we have $\eta(x) = \psi(x') \leq \psi(y') = \eta(y)$. □

Corollary 19. Let S, T and U be three trees. For any degeneration $\varphi : T \to S$ and $\psi : T \to U$, there exists a unique isomorphism $\eta : \Im(\varphi) \to \Im(\psi)$ such that $\psi = \eta \circ \varphi$ if the following condition is satisfied: $\varphi(x) = \varphi(y)$ if and only if $\psi(x) = \psi(y)$.

Lemma 20. Let T and U be two trees. For an arbitrary degeneration $\varphi : T \to U$, there exists a unique embedding $\psi : \Im(\varphi) \to T$ such that $\varphi \circ \psi$ is the identity map on $V(\Im(\varphi))$ and $\psi \circ \varphi$ is the identity map on $V(T) \setminus \mathrm{Dup}(\varphi)$.

Proof. Without loss of generality, we may assume that $U = \Im(\varphi)$. Let $T' = T[\mathbf{x} \notin \mathrm{Dup}(\varphi)]$, and $\eta : T' \to T$ an embedding defined by Corollary 13. We show that $\varphi \circ \eta$ is an isomorphism. Since $\varphi \circ \eta : T' \to U$ is a bijective homomorphism by the definition of T', according to Proposition 5, it suffices to show that, if $\varphi(\eta(x)) < \varphi(\eta(y))$, then $x < y$. By the definition of degeneration, there exists $z \in V(T)$ such that $\eta(x) < z$ and $\varphi(\eta(y)) = \varphi(z)$. Since $\eta(y) = \mathrm{lca}(\varphi^{-1}(\varphi(\eta(y))))$, we have $\eta(x) < z \leq \eta(y)$. Hence $x < y$ since η is an embedding.

Therefore, by letting $\psi = \eta \circ (\varphi \circ \eta)^{-1} : U \to T$, we have the identity map $\varphi \circ \psi$.

The rest of the assertion is proved as follows. For an arbitrary $x \in V(T) \setminus \mathrm{Dup}(\varphi)$, $y \in V(T')$ such that $\eta(y) = x$ is uniquely determined by definition. Hence, we have $\psi(\varphi(x)) = \eta((\varphi \circ \eta)^{-1}(\varphi(\eta(y)))) = \eta(y) = x$. □

Proposition 21. For any degeneration $\varphi : T \to U$, $\varphi(x \frown y) = \varphi(x) \frown \varphi(y)$.

Proof. We first show that, if $\varphi(x) = \varphi(x')$, then $\varphi(x \frown y) = \varphi(x' \frown y)$ for any x, x', $y \in V(T)$. It follows from the definition of degeneration that $\varphi(x \frown x') = \varphi(x) = \varphi(x')$. So we may assume $x \leq x'$. Note that $x \frown y$ and x' are comparable. If $x' \leq x \frown y$, then $x \frown y = x' \frown y$. If $x \frown y < x'$, then $x' \frown y = x'$. Therefore $\varphi(x) \leq \varphi(x \frown y) \leq \varphi(x' \frown y) = \varphi(x')$. Hence $\varphi(x \frown y) = \varphi(x' \frown y)$.

We next show that $\varphi(x \frown y) = \varphi(x) \frown \varphi(y)$ for all x, $y \in V(T)$. According to Lemma 20, we choose $\psi : U \to T$ so that $\varphi \circ \psi$ is the identity map. If $x = \psi(x')$ and $y = \psi(y')$, then $x \frown y \leq \psi(x' \frown y')$. Hence we have $x' \frown y' = \varphi(x) \frown \varphi(y) \leq \varphi(x \frown y) \leq \varphi(\psi(x' \frown y')) = x' \frown y'$. Therefore $\varphi(x \frown y) = \varphi(x) \frown \varphi(y)$. If $v = \psi(\varphi(x))$ and $w = \psi(\varphi(y))$ for any x, $y \in V(T)$, then $\varphi(x \frown y) = \varphi(v \frown w) = \varphi(v) \frown \varphi(w) = \varphi(x) \frown \varphi(y)$. Hence $\varphi(x \frown y) = \varphi(x) \frown \varphi(y)$. □

Corollary 22. Let $\varphi : T \to U$ be a degeneration. For any x, y, $z \in V(T)$, if $x \frown y < x \frown z$, then the following conditions hold:

(1) $\varphi(x) \frown \varphi(y) \leq \varphi(x) \frown \varphi(z)$,
(2) $\varphi(x) \frown \varphi(y) = \varphi(x) \frown \varphi(z)$ if and only if $\varphi(x \frown y) = \varphi(x \frown z)$, and
(3) $\varphi(y) \frown \varphi(z) = \varphi(x) \frown \varphi(z)$.

Proof. Straightforward from Proposition 21. □

Proposition 23. Let S, T, and U be three trees. For two homomorphisms $\varphi : S \to T$ and $\psi : T \to U$, the following properties hold:

(1) if φ and $\psi|_{\Im(\varphi)}$ are both degenerations, then $\psi \circ \varphi$ is also a degeneration. In particular $\mathrm{Dup}(\psi \circ \varphi) = \mathrm{Dup}(\varphi) \cup \varphi^{-1}(\mathrm{Dup}(\psi|_{\Im(\varphi)}))$ holds.
(2) if φ is surjective onto $V(\Im(\varphi))$ and $\psi \circ \varphi$ is a degeneration, then $\psi|_{\Im(\varphi)}$ is also a degeneration.

Proof. (1): Without loss of generality, we may assume that $\Im(\varphi) = T$. It is obvious that $\psi \circ \varphi$ is surjective onto $V(\Im(\psi))$. First, we show $\psi(\varphi(x \frown y)) = \psi(\varphi(x))$ holds for arbitrary x, $y \in V(S)$ such that $\psi(\varphi(x)) = \psi(\varphi(y))$. This is because $\psi(\varphi(x \frown y)) = \psi(\varphi(x) \frown \varphi(y)) = \psi(\varphi(x)) \frown \psi(\varphi(y)) = \psi(\varphi(x))$ according to Proposition 21. Next, we show that there exists $y'' \in V(S)$ such that $\psi(\varphi(y'')) = \psi(\varphi(y))$ and $x < y''$ for arbitrary x, $y \in V(S)$ such that $\psi(\varphi(x)) < \psi(\varphi(y))$. There exists $y' \in V(S)$ such that $\psi(\varphi(y')) = \psi(\varphi(y))$ and $\varphi(x) < \varphi(y')$ since ψ is a degeneration. In the same way, there exists $y'' \in V(S)$ such that $\varphi(y'') = \varphi(y')$ and $x < y''$ since φ is a degeneration. Obviously, $\psi(\varphi(y'')) = \psi(\varphi(y')) = \psi(\varphi(y))$ holds.

According to Proposition 17, either $(\varphi(x), \varphi(y)) \in E(T)$ or $\varphi(x) = \varphi(y)$ holds for any $(x, y) \in E(S)$. Therefore that $\psi(\varphi(x)) = \psi(\varphi(y))$ holds is equivalent to that either $x \in \mathrm{Dup}(\varphi)$ or $\varphi(x) \in \mathrm{Dup}(\psi)$ holds. Hence $\mathrm{Dup}(\varphi \circ \psi) = \mathrm{Dup}(\varphi) \cup \varphi^{-1}(\mathrm{Dup}(\psi))$.

(2): Without loss of generality, we may assume that $\Im(\varphi) = T$. It is obvious that ψ is surjective onto $V(\Im(\psi)) = V(\Im(\psi \circ \varphi))$. First, we show $\psi(\varphi(x) \frown \varphi(y)) = \psi(\varphi(x))$ holds for arbitrary $x, y \in V(S)$ such that $\psi(\varphi(x)) = \psi(\varphi(y))$. Although $\psi(\varphi(x)) \frown \psi(\varphi(y)) \leq \psi(\varphi(x) \frown \varphi(y)) \leq \psi(\varphi(x \frown y))$ generally holds, according to Proposition 21, $\psi(\varphi(x)) \frown \psi(\varphi(y)) = \psi(\varphi(x \frown y)) = \psi(\varphi(x))$ since $\psi \circ \varphi$ is a degeneration. Next, we show that there exists $y' \in V(S)$ such that $\psi(\varphi(y')) = \psi(\varphi(y))$ and $\varphi(x) < \varphi(y')$ for arbitrary x, $y \in V(S)$ such that $\psi(\varphi(x)) < \psi(\varphi(y))$. There exists $y' \in V(S)$ such that $\psi(\varphi(y')) = \psi(\varphi(y))$ and $x < y'$ since $\psi \circ \varphi$ is a degeneration. Moreover, we have $\varphi(x) < \varphi(y')$ since φ is a homomorphism. □

5.2.6 Logical Expressions and Degenerations

For a tree T, let $\pi(\mathbf{x}) : V(T) \to \{\mathbf{t}, \mathbf{f}\}$ denote a unary predicate with a predicate variable $\mathbf{x}$.

Definition 13. If $\pi(r(T)) = \mathbf{t}$, $\mathbf{D}_{\pi(\mathbf{x})} : V(T) \to V(T[\pi(\mathbf{x})])$ is defined as $\mathbf{D}_{\pi(\mathbf{x})}(x) = x$ if $\pi(x) = \mathbf{t}$ and $\mathbf{D}_{\pi(\mathbf{x})}(x) = \mathbf{D}_{\pi(\mathbf{x})}(p(x))$ if $\pi(x) = \mathbf{f}$.

Proposition 24. $\mathbf{D}_{\pi(\mathbf{x})}$ is a degeneration with $\mathrm{Dup}(\mathbf{D}_{\pi(\mathbf{x})}) = \{x \in V(T) | \pi(x) = \mathbf{f}\}$.

Proof. For simplicity, we denote $\mathbf{D}_{\pi(\mathbf{x})}$ by φ. Obviously, φ is surjective.

Let x, y be arbitrary nodes of $V(T)$. By definition, there exist x', $y' \in V(T)$ such that $x \leq x' \wedge \varphi(x) = x'$ and $y \leq y' \wedge \varphi(y) = y'$. Note that x' (y') is the minimum ancestor of x (y) such that $\pi(x') = \mathbf{t}$ ($\pi(y') = \mathbf{t}$, resp.).

If $x < y$, we have $x' \leq y'$ and therefore φ is a

tree homomorphism.

If $\varphi(x) = \varphi(y)$, then $x' = y'$. Therefore, we have $\varphi(x \smile y) = x' = y'$ since $x \smile y \leq x' = y'$.

Next, assume $\varphi(x) <_{\pi(\mathbf{x})} \varphi(y)$. By definition of $T[\pi(\mathbf{x})]$, $x' < y'$ holds. Obviously, we have $x < y'$ and $\varphi(y) = \varphi(y')$. □

5.2.7 Deletion

Definition 14 (Deletion)**.** Let T and U be two trees. A degeneration $\varphi : T \to U$ is called a *deletion* from T if $|\mathrm{Dup}(\varphi)| = 1$. In particular, if a deletion φ is surjective and $\mathrm{Dup}(\varphi) = \{x\}$, φ is called an x-deletion and denoted by D_x.

The following proposition shows that Definition 14 of the deletion is equivalent to the operational definition of the deletion. We omit the proof because it is similar to that of Proposition 15.

Proposition 25. Let T and U be two trees. For $x \in V(T)$, $D_x : T \to U$ satisfies the following properties:

(1) for any $y \in \mathrm{ch}(x)$, $D_x : T[\mathbf{x} \leq y] \to U[\mathbf{x} \leq D_x(y)]$ is an isomorphism, and
(2) $D_x : T[\mathbf{x} \not\leq x] \to U[\bigwedge_{y \in \mathrm{ch}(x)} \mathbf{x} \not\leq D_x(y)]$ is an isomorphism.

Theorem 26 (Decomposition of degeneration)**.** Let φ be a degeneration from T to U with $\mathrm{Dup}(\varphi) = \{x_1, \ldots, x_n\}$. There exist a sequence of trees T_0, T_1, …, T_n, and a sequence of deletions $\varphi_i : T_i \to T_{i+1}$ $(i \in \{0, \ldots, n-1\})$ such that

(1) $T_0 = T$,
(2) $T_n = U$,
(3) $\mathrm{Dup}(\varphi_{i-1} \circ \cdots \circ \varphi_0) = \{x_1, \ldots, x_i\}$, and
(4) $\varphi = \varphi_{n-1} \circ \cdots \circ \varphi_0$;

$$\begin{array}{ccccccccc} T_0 & \xrightarrow{\varphi_0} & T_1 & \xrightarrow{\varphi_1} & \cdots & \xrightarrow{\varphi_{n-2}} & T_{n-1} & \xrightarrow{\varphi_{n-1}} & T_n \\ \| & & \downarrow{\scriptstyle D_{x_0}} & & & & \downarrow{\scriptstyle D_{x_{n-2}}} & & \| \\ T & & \xrightarrow{\qquad\qquad\varphi\qquad\qquad} & & & & & & U\,. \end{array}$$

Proof. We apply induction on n. For $n = 1$, φ is a deletion by definition. Now assume that $n \geq 2$. Let $\varphi_0 : T \to T_1$ be D_{x_1}. By Proposition 18, there exists $\psi : T_1 \to U$ such that $\varphi = \psi \circ \varphi_0$. Moreover, by Proposition 23, ψ is a degeneration such that $\mathrm{Dup}(\varphi) = \mathrm{Dup}(\varphi_0) \cup \varphi_0^{-1}(\mathrm{Dup}(\psi))$. In particular, $\mathrm{Dup}(\psi) = \{\varphi_0(x_2), \ldots, \varphi_0(x_n)\}$ follows $\mathrm{Dup}(\varphi) = \mathrm{Dup}(\varphi_0) \cup \varphi_0^{-1}(\mathrm{Dup}(\psi))$: if $\varphi_0(x_1) \in \mathrm{Dup}(\psi)$, then $p(x_1) \in \mathrm{Dup}(\varphi)$, and therefore $\varphi_0(x_1) \in \{\varphi_0(x_2), \ldots, \varphi_0(x_n)\}$.

Now we can apply the induction hypothesis to ψ. Hence, there exists a sequence of trees T_2, T_3, …, T_n such that

(1) $T_n = U$,
(2) $\mathrm{Dup}(\varphi_{i-1} \circ \cdots \circ \varphi_1) = \{\varphi_0(x_2), \ldots, \varphi_0(x_i)\}$ for $i \in 2, \ldots, n-1$, and
(3) $\psi = \varphi_{n-1} \circ \cdots \circ \varphi_1$.

Obviously, $\varphi = \psi \circ \varphi_0 = \varphi_{n-1} \circ \cdots \circ \varphi_0$ holds. In addition, we have $\mathrm{Dup}(\varphi_{i-1} \circ \cdots \circ \varphi_1 \circ \varphi_0) = \mathrm{Dup}(\varphi_0) \cup \varphi_0^{-1}(\mathrm{Dup}(\varphi_{n-1} \circ \cdots \circ \varphi_1)) = \{x_1, \ldots, x_i\}$. Therefore, the assertion of this theorem holds. □

5.2.8 Duality between Embedding and Degeneration

In Lemma 20, we see that, for a given degeneration φ, there exists an embedding ψ such that $\varphi \circ \psi$ is an identity map. In fact, its reverse also holds.

Theorem 27. Let T and U be two trees. The following two properties hold:

(1) For an arbitrary degeneration $\varphi : T \to U$, there exists a unique embedding $\psi : \Im(\varphi) \to T$ such that $\varphi \circ \psi$ is the identity map on $V(\Im(\varphi))$ and $\psi \circ \varphi$ is the identity map on $V(T) \setminus \mathrm{Dup}(\varphi)$.
(2) For an arbitrary embedding $\psi : U \to T$, there exists a unique degeneration $\varphi : \Im(\psi) \to U$ such that $\varphi \circ \psi$ is the identity map on $V(U)$ and $\psi \circ \varphi$ is the identity map on $V(\Im(\psi)) \setminus \mathrm{Dup}(\varphi)$.

Proof. As the proof of (1) is already given in Lemma 20, we prove (2) in the following.

Without loss of generality, we may assume that $T = \Im(\psi)$. Let T' be $T[\mathbf{x} \in \psi(V(U))]$, and η be the canonical degeneration $\mathbf{D}_{\mathbf{x} \in \psi(V(U))} : T \to T'$ (Proposition 24). First, we show that $\eta \circ \psi$ is an isomorphism. Since $\eta \circ \psi : U \to T'$ is a bijective homomorphism by the definition of T', according to Proposition 5, it suffices to show that, if $\eta(\psi(x)) < \eta(\psi(y))$, then $x < y$. By the definition of a degeneration, there exists $z \in V(T)$ such that $\psi(x) < z$ and $\eta(\psi(y)) = \eta(z)$. Since $\psi(y) = \mathrm{lca}(\eta^{-1}(\eta(\psi(y))))$, we have $\psi(x) < z \leq \psi(y)$. Hence, $x < y$ since ψ is an embedding, and we conclude that $\eta \circ \psi$ is an isomorphism.

By letting $\varphi = (\eta \circ \psi)^{-1} \circ \eta : T \to U$, we have $\varphi \circ \psi$ is the identity map on $V(U)$.

The rest of the assertion is proved as follows. For an arbitrary $x \in V(T) \setminus \mathrm{Dup}(\eta)$, $y \in V(T')$ such that $\psi(y) = x$ is uniquely determined by definition. Hence, we have
$\psi(\varphi(x)) = \psi((\eta \circ \psi)^{-1}(\eta(\psi(y)))) = \psi(y) = x$. □

This theorem is to the effect that there ex-

ists a unique degeneration $\bar{\psi}$ (an embedding $\bar{\varphi}$, resp.) if an embedding ψ (a degeneration φ, resp.) is given.

5.3 Characterization of Alignment of Trees

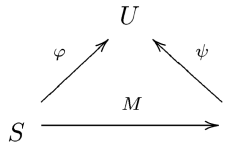
Now we are ready to give a definition of the alignment of trees in a formal manner.

Throughout in this section, S and T are rooted trees, and $M \subseteq V(S) \times V(T)$ is a tree mapping from S to T.

Definition 15. A tree mapping M from S to T is *alignable* if and only if there exists a triplet (U, φ, ψ) such as

(1) $\varphi : S \to U$ is an embedding,
(2) $\psi : T \to U$ is an embedding, and
(3) $\varphi(x) = \psi(y)$ for all $(x, y) \in M$;

We call (U, φ, ψ) a *union* on M.



Lemma 28. For an alignable mapping M with a union (U, φ, ψ), $(s, t) = (s, \bar{\psi}(\varphi(s)))$ holds for an arbitrary $(x, y) \in M$, where $\bar{\psi}$ is the degeneration such that $\bar{\psi} \circ \psi$ is the identity map of T.

Proof. The assertion is obvious since $\bar{\psi} \circ \psi$ is the identity map of T. □

Let $T = (V(T), <)$ be a rooted tree and a and b be two nodes in $V(T)$ such that $p(a) = p(b)$. We define $V(T)/\{(a, b)\}$ and $<_{/\{(a,b)\}}$ as follows.

$$V(T)/\{(a, b)\} = V(T) \setminus \{a, b\} \cup \{\nu\}.$$

For distinct $x, y \in V(T)/\{(a, b)\}$, $x <_{/\{(a,b)\}} y$ holds if and only if one of the following conditions holds.

(1) $x \neq \nu$, $y \neq \nu$ and $x < y$.
(2) $x = \nu$ and $y > a$ (therefore, $y > b$).
(3) $y = \nu$ and $x < a \vee x < b$.

Lemma 29. $(V(T)/\{(a, b)\}, <_{/\{(a,b)\}})$ is a rooted tree.

Proof. First, we show that $x <_{/\{(a,b)\}} z$ if $x <_{/\{(a,b)\}} y$ and $y <_{/\{(a,b)\}} z$. If $x, y, z \neq \nu$, $x < y$ and $y < z$ hold, and therefore $x < z$ holds. If $x = \nu$, $a < y$ and $y < z$ hold, and therefore $a < z$ holds. If $y = \nu$, $x < a \vee x < b$ and $z > a \wedge z > b$ hold, and therefore $x < z$ holds. If $z = \nu$, $x < y$ and $y < a \vee y < b$ hold, and therefore $x < a \vee x < b$ holds. Hence, we have $x <_{/\{(a,b)\}} z$ for each case.

Secondly, we show that $\mathcal{A}_x = \{y \in V(T)/\{(a, b)\} | y > x\}$ is totally ordered. Take arbitrary distinct $y, z \in \mathcal{A}_x$ If $y \neq \nu$ and $z \neq \nu$, $y > x \wedge z > x$ for $x \neq \nu$ or $y > a \wedge z > a$ for $x = \nu$ holds. In any case, $y < z$ or $y > z$ holds, and therefore we have $y <_{/\{(a,b)\}} z$ or $y >_{/\{(a,b)\}} z$. If $y = \nu$, $x < a \vee x < b$ and $x < z$ hold, and therefore one of $a < z$, $b < z$, $z < a$ and $z < b$ holds. Hence, we have $\nu <_{/\{(a,b)\}} z$ or $\nu >_{/\{(a,b)\}} z$. □

Definition 16. Let T be a tree. For distinct a, b in $V(T)$ such that $p(a) = p(b)$, $T/\{(a, b)\}$ denotes the tree $(V(T)/\{(a, b), <_{/\{(a,b)\}}\})$.

Proposition 30. Let S and T be two trees. Any singleton tree mapping $M = \{(s, t)\}$ from S to T is alignable.

Proof. Let $V(\bar{U})$ be $V(S) \cup V(T)$ and define a relation $<_{\bar{U}}$ such that $x <_{\bar{U}} y$ holds, for distinct $x,\ y \in V(\bar{U})$, if and only if one of the following conditions holds.

(1) $x, y \in V(S)$ and $x < y$,
(2) $x, y \in V(T)$ and $x < y$,
(3) $x \in V(S)$ and $y \in V(T[\mathbf{x} > t])$,
(4) $x \in V(T(t))$ and $y \in V(S[\mathbf{x} > s])$.

It is easy to show that $\bar{U} = (V(\bar{U}), <_{\bar{U}})$ is a rooted tree, and the proof is left to the reader.

We have $p(s) = p(t)$ in $\bar{U}$, since $\mathcal{A}_s = \mathcal{A}_t = V(S[\mathbf{x} > s]) \cup V(T[\mathbf{x} > t])$ by definition of $<_{\bar{U}}$.

Thus, we can apply Lemma 29 to $\bar{U}$, and $U = (V(U), <) = \bar{U}/\{(s, t)\}$ is a rooted tree. Moreover, it is easy to see that natural inclusion maps $\varphi : V(S) \to V(U)$ and $\psi : V(T) \to V(U)$ are embeddings. In particular, since $\varphi(s) = \psi(t)$ holds, $M = \{(s, t)\}$ is an alignable mapping. □

Lemma 31. Let $\eta : S \to \bar{S}$ is an embedding. For a tree mapping M, the following properties are equivalent.

(1) M is alignable.
(2) $\bar{M} = \{(\eta(s), t) | (s, t) \in M\}$ is alignable.

Proof. By definition of an alignable mapping, (2)⇒(1) is trivial.

In the following, we show (1)⇒(2). Let (U, φ, ψ) be a union on M: hence, the embeddings $\varphi : S \to U$ and $\psi : T \to U$ satisfy $\varphi(s) = \psi(t)$ for all $(s, t) \in M$. By Theorem 16, we only have to take care of the following two cases.

(1) η is not surjective and $\mathrm{red}(\eta) = 0$
(2) $\Im(\eta) = \bar{S}$ and $\mathrm{red}(\eta) = 1$.

Case (1): Letting $V(\bar{U}) = V(\bar{S}[\mathbf{x} \notin \eta(S)]) \cup V(U)$, we define the relation $<_{\bar{U}}$ over $V(\bar{U})$ such that, for distinct $x, y \in V(\bar{U})$, $x <_{\bar{U}} y$ if and only if one of the following holds.

(a) $x, y \in V(\bar{S}[\mathbf{x} \notin \eta(S)])$ and $x < y$;
(b) $x, y \in V(U)$ and $x < y$;
(c) $x \in V(U)$ and $y \in V(\bar{S}[\mathbf{x} > \eta(r(S))])$.
It is easy to see that $\bar{U} = (V(\bar{U}), <_{\bar{U}})$ is a tree, and the proof is left to the reader. Let $\alpha : V(\bar{S}[\mathbf{x} \notin \eta(S)]) \to V(\bar{U})$ and $\beta : V(U) \to V(\bar{U})$ denote the natural inclusions. We define $\bar{\varphi} : V(\bar{S}) \to V(\bar{U})$ by $\bar{\varphi}(x) = \alpha(x)$ if $x \in V(\bar{S}[\mathbf{x} \notin \eta(S)])$ and $\bar{\varphi}(x) = \beta(\varphi(\eta^{-1}(x)))$ if $x \in \eta(V(S))$. Also, we define $\bar{\psi} : V(T) \to V(\bar{U})$ by $\bar{\psi} = \beta \circ \psi$. It is easy to see that both $\bar{\varphi}$ and $\bar{\psi}$ are embeddings, and the proof is left to the reader. Since $\bar{\varphi}(\eta(s)) = \beta(\varphi(\eta^{-1}(\eta(s)))) = \beta(\varphi(s)) = \beta(\psi(t)) = \bar{\psi}(t)$, we have the conclusion in the case of (1).

Case (2): In the following, we use the following notation.

- $V(\bar{S}) \setminus \eta(V(S)) = \{\sigma\}$
- $p(\sigma) = \eta(p)$
- $\mathrm{ch}(\sigma) = \{\eta(c_1), \ldots, \eta(c_n)\}$

Now, letting $V(\bar{U}) = V(U) \cup \{\bar{\sigma}\}$, we define the relation $<_{\bar{U}}$ over $V(\bar{U})$ such that, for distinct $x, y \in V(\bar{U})$, $x <_{\bar{U}} y$ if and only if one of the following holds.
(1) $x, y \in V(U)$ and $x < y$.
(2) $x = \bar{\sigma}$ and $y \geq \varphi(p)$.
(3) $x \in V(U)$, $\exists (z \in V(U))[\varphi(c_i) \leq z < \varphi(p) \wedge x \leq z]$ and $y = \bar{\sigma}$.
It is easy to see that $\bar{U} = (V(\bar{U}), <_{\bar{U}})$ is a tree, and the proof is left to the reader. Letting $\alpha : V(U) \to V(\bar{U})$ be the natural inclusion, we define $\bar{\varphi} : V(\bar{S}) \to V(\bar{U})$ by $\bar{\varphi}(x) = \alpha(\varphi(\eta^{-1}(x)))$ if $x \neq \sigma$ and $\bar{\varphi}(\sigma) = \bar{\sigma}$. Further, we define $\bar{\psi} = \alpha \circ \psi : V(T) \to V(\bar{U})$. It is easy to see that both $\bar{\varphi}$ and $\bar{\psi}$ are embeddings, and the proof is left to the reader. Since $\bar{\varphi}(\eta(s)) = \alpha(\varphi(\eta^{-1}(\eta(s)))) = \alpha(\varphi(s)) = \alpha(\psi(t)) = \bar{\psi}(t)$, we have the conclusion in the case of (2). □

Lemma 32. Let M' be a subset of M. If M is alignable, then M' is also alignable.
Proof. A union on M is also a union on M'. □

By definition of a tree mapping, for $(s, t) \in M$, if $s = r(S)$, then $t = r(T)$.
Lemma 33. Let (U, φ, ψ) be a union on M. Then, there exist φ' and ψ' such that (U, φ', ψ') is also a union on M and $\varphi'(r(S)) = \psi'(r(T))$.

In particular, the following are equivalent.
(1) M is alignable.
(2) $M \cup \{(r(S), r(T))\}$ is alignable.
Proof. Let $(s, t) \in M$. $\varphi(r(S))$ and $\psi(r(T))$ are comparable, since they are ancestors of $\varphi(s) = \psi(t)$. If $\varphi(r(S)) = \psi(r(T))$, there is nothing to prove. Without loss of generality, we may assume that $\varphi(r(S)) < \psi(r(T))$. Define $\varphi' : V(S) \to V(U)$ by $\varphi'(x) = \varphi(x)$ if $x \neq r(S)$ and $\varphi'(r(S)) = \psi(r(T))$. In the following, we see that φ' is an embedding. First, let $x, y \in V(S)$ satisfy $x < y$. If $y \neq r(S)$, $\varphi'(x) < \varphi'(y)$ holds since φ is a homomorphism. If $y = r(S)$, $\varphi'(x) = \varphi(x) < \varphi(r(S)) < \varphi'(r(S))$ holds. Thus, φ' is a homomorphism. The property $x < y$ if $\varphi'(x) < \varphi'(y)$ is also easily proved. Consequently, we see that φ' is an embedding.

Since (2)⇒(1) follows Lemma 32, we only have to show (1)⇒(2). As shown in the first part, if (U, φ, ψ), we have another union (U, φ', ψ') such that $\varphi'(r(S)) = \psi'(r(T))$. Therefore, $M \cup \{(r(S), r(T))\}$ is alignable. □

In Lemma 34, we use the following notations.

- S_i denotes the tree $S(\sigma_i)$ for $\mathrm{ch}(r(S)) = \{\sigma_1, \ldots, \sigma_m\}$.
- T_i denotes the tree $T(\tau_i)$ for $\mathrm{ch}(r(T)) = \{\tau_1, \ldots, \tau_n\}$.
- By symmetry, we assume that $m \leq n$.
- $M_i \subset V(S_i) \times V(T_i)$ for $i = 1, \ldots, m$ denotes the tree mapping $\{(s, t) \in M | s \in V(S_i) \wedge t \in V(T_i)\}$

Lemma 34. If $M = \bigcup_{i=1}^{m} M_i$ and each M_i is alignable, then M is alignable.
Proof. Let (U_i, φ_i, ψ_i) be a union on M_i: hence, the embeddings $\varphi_i : S_i \to U_i$ and $\psi_i : T_i \to U_i$ satisfy $\varphi_i(s) = \psi_i(t)$ for all $(s, t) \in M_i$.

Letting $V(U)$ be $\{\rho\} \cup \bigcup_{i=1}^{m} V(U_i) \cup \bigcup_{i=m+1}^{n} V(T_i)$, we define the relation $<_U$ so that, for distinct $x, y \in V(U)$, $x <_U y$ if and only one of the following holds.
(1) $1 \leq i \leq m$, $x, y \in V(U_i)$ and $x < y$;
(2) $m < i \leq n$, $x, y \in V(T_i)$ and $x < y$;
(3) $y = \rho$.
It is easy to see $U = (V(U), <_U)$ is a tree, and the proof is left to the reader.

Let $\alpha_i : V(U_i) \to V(U)$ for $i \in \{1, \ldots, m\}$ and $\beta_i : V(T_i) \to V(U)$ for $i \in \{m+1, \ldots, n\}$ be the natural inclusions. Thus, we define $\varphi : V(S) \to V(U)$ and $\psi : V(T) \to V(U)$ as follows: $\varphi(x) = \alpha_i(\varphi_i(x))$ if $x \in V(S_i)$; $\varphi(r(S)) = \rho$; $\psi(x) = \alpha_i(\psi_i(x))$ if $x \in V(T_i)$ for $i \in \{1, \ldots, m\}$; $\psi(x) = \beta_i(x)$ if $x \in V(T_i)$ for $i \in \{m+1, \ldots, n\}$; and $\psi(r(T)) = \rho$. It is easy to see that φ and ψ are embeddings. Since $\varphi(s) = \alpha_i(\varphi_i(s)) = \alpha_i(\psi_i(t)) = \psi(t)$ for $(s, t) \in M_i$, we have the conclusion. □

6. Tree Mapping Condition for Alignment of Trees

Now we are ready to prove our main theorem, where the tree mapping condition for the alignment of trees is shown.

Theorem 35. For a tree mapping M from a tree S to a tree T, the following two properties are equivalent.

(1) M is alignable.

(2) $\forall(s_1,t_1), (s_2,t_2), (s_3,t_3) \in M\ [s_1 \smile s_2 < s_1 \smile s_3 \Rightarrow t_2 \smile t_3 = t_1 \smile t_3]$.

Proof. **(1)⇒(2):** Let (U,φ,ψ) be a union on M: hence, $\varphi : S \to U$ and $\psi : T \to U$ are embeddings such that $\varphi(s) = \psi(t)$ for an arbitrary $(s,t) \in M$. Further, $\bar{\psi}$ denote the degeneration such that $\bar{\psi} \circ \psi$ is the identity map of T (Theorem 27). Suppose that (s_1,t_1), (s_2,t_2), and (s_3,z_3) are any three elements of M such that $s_1 \smile s_2 < s_1 \smile s_3$. We have $\varphi(s_1) \smile \varphi(s_2) < \varphi(s_1) \smile \varphi(s_3)$ by Corollary 11, and therefore $\varphi(s_2) \smile \varphi(s_3) = \varphi(s_1) \smile \varphi(s_3)$. Also, we have $\bar{\psi}(\varphi(s_2)) \smile \bar{\psi}(\varphi(s_3)) = \bar{\psi}(\varphi(s_2) \smile \varphi(s_3)) = \bar{\psi}(\varphi(s_1) \smile \varphi(s_3)) = \bar{\psi}(\varphi(s_1)) \smile \bar{\psi}(\varphi(s_3))$ by Proposition 21. Since $\bar{\psi}(\varphi(s_1)) = t_1$, $\bar{\psi}(\varphi(s_2)) = t_2$ and $\bar{\psi}(\varphi(s_3)) = t_3$ hold by Lemma 28, we conclude that $t_2 \smile t_3 = t_1 \smile t_3$.

(2)⇒(1): The assertion in the case of $|M| = 1$ directly follows Proposition 30.

Let $|M| \geq 2$ for the induction step. Let M be the set of node pairs $\{(s_1,t_1),\ldots,(s_n,t_n)\}$, $X \subseteq V(S)$ denote the set of nodes $\{s_1,\ldots,s_n\}$, and $Y \subseteq V(T)$ denote the set of nodes $\{t_1,\ldots,t_n\}$.

It suffices to prove the assertion of the theorem under the hypothesis that $\mathrm{lca}(X) = r(S)$ and $\mathrm{lca}(Y) = r(T)$. In fact, for the embeddings $\alpha = \mathcal{E}_{\mathbf{x} \leq \mathrm{lca}(X)} : S(\mathrm{lca}(X)) \to S$ and $\beta = \mathcal{E}_{\mathbf{x} \leq \mathrm{lca}(Y)} : T(\mathrm{lca}(Y)) \to T$, Lemma 31 asserts that, if $M' = \{(\alpha^{-1}(s), \beta^{-1}(t)) | (s,t) \in M\}$ is alignable, then M is alignable.

Also, we may assume that M does not contain $(r(S), r(T))$, since, if M contains it, we only have to eliminate it by Lemma 33.

We now choose $X_k = \{s_1,\ldots,s_k\}$, by reordering s_i's if necessary, such that

- $k \geq 1$,
- $\mathrm{lca}(X_k)$ is not the root of S, and
- for any $x \in X \setminus X_k$, $\mathrm{lca}(X_k \cup \{x\}) = r(S)$.

Note that $k < n$. Let us denote by Y_k the set of nodes $\{t_1,\ldots,t_k\}$ corresponding to X_k.

Claim 1. For any $i \leq k$ and $j > k$, $s_i \smile s_j$ is the root of S.

Proof. The two nodes $s_i \smile s_j$ and $\mathrm{lca}(S_k)$ are comparable since $s_i \in X_k$. Now assume that $s_i \smile s_j \leq \mathrm{lca}(X_k)$. It follows that $\mathrm{lca}(S_k \cup \{s_j\}) = \mathrm{lca}(X_k)$. This contradicts the definition of X_k. Hence $\mathrm{lca}(X_k) < s_i \smile s_j$, and in particular $s_i \smile s_j = \mathrm{lca}(X_k \cup \{s_j\})$. This implies that $s_i \smile s_j$ is the root of S. □

Let $A = \{x \in \mathrm{ch}(r(S)) | \exists(i)[1 \leq i \leq k \wedge s_i \leq x]\}$ and $B = \{x \in \mathrm{ch}(r(S)) | \exists(j)[k < j \leq n \wedge s_j \leq x]\}$. We have $A \cap B = \emptyset$, since, if $x \in A \cap B$, we have $s_i \smile s_j \leq x$ for $1 \leq i \leq k$ and $k < j \leq n$, as is contradictory with Claim 1.

Thus, by inserting nodes as children of $r(S)$ if necessary, we may assume the following properties (Lemma 31 asserts that, if M is alignable after insertion of nodes, it is alignable without the insertion):

- the children of $r(S)$ are only two nodes a and b,
- $\mathrm{lca}(S_k) \leq a$, and
- $\mathrm{lca}(X \setminus S_k) \leq b$.

Now, to apply similar discussion to Y_k, we claim the following.

Claim 2. For any $i \leq k$ and $j > k$, $t_i \smile t_j$ is the root of T.

Proof. We start by showing that, for any $i' \leq k$ and $j' > k$, $t_i \smile t_j = t_{i'} \smile t_{j'}$. By Claim 1, we now have $s_i \smile s_{i'} < s_i \smile s_j$. Hence, since M satisfies (2) in the statement, $t_{i'} \smile t_j = t_i \smile t_j$ holds. In the same way, we have $s_j \smile s_{j'} \leq b < s_{i'} \smile s_j$ and therefore $t_{i'} \smile t_j = t_{i'} \smile t_{j'}$. Hence, we conclude $t_i \smile t_j = t_{i'} \smile t_{j'}$. Therefore, we have $t_i \smile t_j = t_{i'} \smile t_{j'}$. Next, we show the assertion of the claim. Since $t_i \smile t_j = t_{i'} \smile t_{j'}$ for all $i' \leq k$ and $j' > k$, we have $\mathrm{lca}(Y) \leq t_i \smile t_j$. Since $\mathrm{lca}(Y)$ is the root of T, $t_i \smile t_j$ is also the root of T. □

Therefore, in the same way as the case of S, by inserting nodes as children of $r(T)$ if necessary, we may assume the following properties:

- the children of $r(T)$ are only two nodes α and β,
- $\mathrm{lca}(Y_k) \leq \alpha$, and
- $\mathrm{lca}(Y \setminus Y_k) \leq \beta$.

By the induction hypothesis, $M_k = \{(s_1,t_1),\ldots,(s_k,t_k)\}$ is an alignable mapping from $S(a)$ to $T(\alpha)$, and $M \setminus M_k$ is an alignable mapping from $S(b)$ to $T(\beta)$. Then, by Lemma 34, M is alignable. □

7. Conclusion

In this paper, we have introduced a new theoretical formulation of the tree edit distance, which allows us to describe distinct semantics of tree edit distance measures. We have focused on a significant distance measure, the alignment of trees, and shown the tree mapping condition of the alignment of trees, which has remained unknown in prior work. By using our formulation, we have redefined the alignment of trees. We then established the declarative definition of the alignment of trees.

The theoretical framework that we have formulated in the paper is generally applicable to all the edit-based approaches in trees. Then, this framework can be utilized for the analysis of the other edit-based tree matching problems as well.

Acknowledgments This work is partially supported by Grand-in-Aid for Scientific Research 16016275 and 17700138 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1) Ferraro, P. and Godin, C.: A Distance Measure between Plant Architectures, *Annals of Forest Science*, Vol.57, pp.445–461 (2000).
2) Fukagawa, D. and Akutsu, T.: Fast algorithms for comparison of similar unordered trees, *Proc. 15th Int. Symp. Algorithms and Computation* (*ISAAC 2004*), LNCS 3341, pp.452–463 (2004).
3) Gusfield, D.: *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press (1997).
4) Höchsmann, M., Töller, T., Giegerich, R. and Kurtz, S.: Local Similarity in RNA Secondary Structures, *Proc. Computational Systems Bioinformatics*, IEEE, pp.159–168 (2003).
5) Hogue, A. and Karger, D.: Thresher: Automating the Unwrapping of Semantic Content from the World Wide Web, *14th International World Wide Web Conference* (*WWW2005*), pp.86–95 (2005).
6) Jansson, J. and Lingas, A.: A fast algorithm for optimal alignment between similar ordered trees, *Fundamenta Informaticae*, Vol.56, pp.105–120 (2003).
7) Jiang, T., Wang, L. and Zhang, K.: Alignment of trees — An alternative to tree edit, *Theoretical Computer Science*, Vol.143, pp.137–148 (1995).
8) Nishimura, N., Ragde, P. and Thilikos, D.M.: Finding Smallest Supertrees under Minor Containment, WG '99, LNCS 1665, pp.303–312 (1999).
9) Sakakibara, Y.: Pair hidden Markov models on tree structures, *Bioinformatics*, Vol.19, pp.232–240 (2003).
10) Sakamoto, H., Murakami, Y., Arimura, H. and Arikawa, S.: Extracting Partial Structures from HTML Documents, *Proc. 14th International FLAIRS Conference*, AAAI Press, pp.264–268 (2001).
11) Selkow, S.M.: The Tree-to-Tree Editing Problem, *Information Processing Letters*, Vol.6, No.6, pp.184–186 (1977).
12) Tai, K.-C.: The Tree-to-Tree Correction Problem, *J. ACM*, Vol.26, No.3, pp.422–433 (1979).
13) Torsello, A. and Hancock, E.R.: Matching and Embedding through Edit-Union of Trees, ECCV 2002, LNCS 2352, pp.822–836 (2002).
14) Torsello, A. and Hancock, E.R.: Graph Clustering with Tree-Unions, CAIP 2003, LNCS 2756, pp.451–459 (2003).
15) Touzet, H.: A linear tree edit distance algorithm for similar ordered trees, *16th Annual Symposium on Combinatorial Pattern Matching* (*CPM 2005*), LNCS 3537, pp.334–345 (2005).
16) Valiente, G.: An Efficient Bottom-Up Distance between Trees, *Proc. 8th Int. Symposium on String Processing and Information Retrieval*, IEEE Computer Science Press, pp.212–219 (2001).
17) Vilares, M., Ribadas, F.J. and Darriba, V.M.: Approximate VLDC Pattern Matching in Shared-Forest, *Proc. 2nd International Conference on Computational Linguistics and Intelligent Text Processing*, LNCS 2004, pp.483–494 (2001).
18) Wagner, R. and Fischer, M.: The string-to-string correction problem, *J. ACM*, Vol.21, No.1, pp.168–173 (1974).
19) Wang, J.-L. and Zhang, K.: Finding similar consensus between trees: An algorithm and a distance hierarchy, *Pattern Recognition*, Vol.34, pp.127–137 (2001).
20) Wang, L. and Zhao, J.: Parametric Alignment of Ordered Trees, *Bioinformatics*, Vol.19, No.17, pp.2237–2245 (2003).
21) Zhang, K. and Jiang, T.: Some MAX SNP-hard results concerning unordered labeled trees, *Information Processing Letters*, Vol.49, No.5, pp.249–254 (1994).
22) Zhang, K. and Shasha, D.: Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems, *SIAM Journal on Computing*, Vol.18, No.6, pp.1245–1262 (1989).

23) Zhang, K., Statman, R. and Shasha, D.: On the editing distance between unordered labeled trees, *Information Processing Letters*, Vol.42, No.3, pp.133–139 (1992).

(Received February 9, 2005)
(Accepted March 10, 2005)

Tetsuji Kuboyama is a Research Associate of Center for Collaborative Research, the University of Tokyo. He received the B.Eng. in Computer Science and Communication Engineering, the M.Eng. degree in Information Systems all from Kyushu University, in 1992 and 1994, respectively. His research interests include Approximate Pattern Matching, Data Mining, and Automated Reasoning.

Kilho Shin is a Visiting Researcher of Research Center for Advanced Science and Technology, the University of Tokyo. He received the M.S. degree in Mathematics from the University of Tokyo in 1995. His research area includes Theory for Security Protocols and Algebraic Aspects of Graph Theory.

Tetsuhiro Miyahara is an Associate Professor of the Department of Intelligent Systems, Hiroshima City University, Hiroshima, Japan. He received the B.S. degree in Mathematics, the M.S. and Dr.Sci. degrees in Information Systems all from Kyushu University, Fukuoka, Japan in 1984, 1986 and 1996, respectively. His research interests include Algorithmic Learning Theory, Knowledge Discovery, Inductive Logic Programming and Machine Learning.