アルゴリズム 51−7
(1996. 5. 29)

順序がない木の間の類似度問題

劉 紹明† 田中 栄一††

† 神戸大学大学院自然科学研究科
†† 神戸大学工学部電気電子工学科

順序がない木について，木の間の距離を計算する問題，木の中に他の木とよく類似している部分を抽出する問題，木の間の最大共通類似部分を抽出する問題がある．最後の問題は他の二つの問題を含んでいる．本論文では，共通類似部分間の距離を用いて順序がない木の間の最大共通類似部分 (LCSS) を抽出する問題を論じ，共通類似部分間の距離の計算法を提案する．計算法の時間計算量は，根があり順序がない木の場合は $O(m_a N_a N_b)$ であり，根がなく順序がない木の場合は $O(m_a m_b N_a N_b)$ である．ここで，$m_a(m_b)$ と $N_a(N_b)$ はそれぞれ木 $T_a(T_b)$ の最大頂点次数と頂点数を表す．距離の計算法を用いて一つの LCSS，或は全ての LCSSs を抽出するアルゴリズムを作ることは容易である．これらのアルゴリズムは構造・活性問題や構造比較問題等に応用できる．

Several Similarity Problems between Unordered Trees

Shaoming LIU† and Eiichi TANAKA††

†The Graduate School of Science and Technology, Kobe University
††Department of Electrical and Electronics Engineering, Faculty of Engineering, Kobe University

There are three similarity problems for unordered trees. That is, to compute the distance between trees, to find one of the largest similar substructures in a tree to a specific tree, and to find one of the largest common similar substructures(in short, LCSS) for trees. The third problem includes the first and the second ones. In this paper we define a distance which determines the LCSS and discuss the LCSS problem using the distance. We give a computing method of this distance between rooted and unordered trees(in short *R-trees*) and that between unrooted and unordered trees(in short *trees*). The time complexity of the computing method for *R-trees* is $O(m_a N_a N_b)$, and that for *trees* is $O(m_a m_b N_a N_b)$, where, $m_a(m_b)$ and $N_a(N_b)$ are the largest degree of a vertex of tree $T_a(T_b)$ and the number of vertices of $T_a(T_b)$, respectively. It is easy to make an algorithm for finding one of the LCSSs largest common similar substructures or enumerating all of the largest common similar substructures between two trees. Those computing methods can be applied to structure-activity studies and structure comparison problems.

1 Introduction

The study of relationship between the structures of chemical compounds and their properties is one of the most important problems in chemistry [1],[2], and several substructure search systems have already been constructed [3]. For these purposes methods for extracting similar substructures and common similar substructures for plural graphs must be studied. If we replace a ring or a set of rings with a super-atom, a comparatively simple graph can be expressed as a tree [4]. In relation to those problems, the largest common substructure(in short, LCS) problem has been studied [5]−[7]. As the first step to study similarity measures between graphs, several tree distances for rooted and ordered trees(in short *RO-tree*) [8]−[11], those for trees embedded in a plane [12],[13], those for unordered trees [14]−[16] have already been proposed. The largest similar substructure(in short, LSS) problem [17],[18] were discussed. Recently, the largest common similar substructure(in short, LCSS) problem between trees embedded in a plane has been reported [19]. Obviously, the LCSS problem is a generalization of both the LCS problem and the LSS problem, and has wider applications. In this paper we will define a distance between unordered trees and define the LCSS problem using this distance. This distance is a general similarity measure between unordered trees, and can be applied to structure-activity studies and structure comparison problems.

2 Definitions

Let $T = (V, E)$ be an unrooted and unordered tree(in short, *tree*), where V is the set of vertices and E is the set of edges. In this paper all of the vertices in a *tree* are numbered and labeled. If a tree has the root, we can make a rooted and unordered tree(in short, *R-tree*). Let $T(u) = (V(u), E(u))$ denote an *R-tree* with the root u. Consider an *R-tree*. Let $An(x)$ be the set of proper ancestors of x. Note that $x \notin An(x)$. Let $\widehat{An}(x, x') = An(x') - An(x)$ for $x \in An(x')$, where "$A - B$" denotes that removing all of the elements in set B from set A. Let $Ch(x)$ denote the set of children of x. Let $T(u, x)$ be the subtree of $T(u)$ with the root x. If there is no confusion, we use notation $T(x) = (V(x), E(x))$ instead of $T(u, x)$. Let $x_i (i = 1, \ldots, |Ch(x)|)$ denote a child of x, and $F(\bar{x}) = (V(\bar{x}), E(\bar{x}))$ denote the forest that consists of trees $T(x_1), \ldots, T(x_{|Ch(x)|})$, where |A| indicates the number of elements of set A.

Consider a mapping from vertices of T_a to those of T_b. If vertex x of T_a maps to vertex y of T_b, we write (x, y). We call a set of (x, y)s a mapping from T_a to T_b, and denote it by M. If vertex x of T_a maps to vertex y of T_b and the label of x is different from that of y, we say label of x is substituted to that of y. If a vertex x of T_a does not map to any vertex y of T_b, we say x is deleted. If a vertex y of T_b does not be mapped from any vertex x of T_a, we say y is inserted. By this interpretation, we can see that a mapping defines a transformation from T_a to T_b. A substitution, an insertion and a deletion are called transformational operations. The weight of a transformation is the total of all of the weights of edit operations. Furthermore, the weight of a transformation determined by a mapping M is called the weight of M.

Tanaka [10] and Tanaka [15] defined a strongly structure preserving mapping(in short, SSP-mapping) between *RO-trees* and that between unodered trees, respectively. Muguruma, Tanaka and Masuda [14] defined a subclass of the SSP-mapping between unordered trees, which is called a closest common ancestor mapping(in short, C-mapping). To express the largest similar substructure, Liu, Tanaka and Masuda [17] defined a maximal closest common ancestor mapping(in short, maximal C-mapping) between *RO-trees* and that between trees embedded in a plane, and Liu and Tanaka [16] defined a maximal C-mapping between *R-trees* and that between *trees*. In Reference [15] Tanaka showed that the SSP-mapping can make more suitable correspondences between the similar substructures of T_a and those of T_b than Tai mapping [8]. To express a similar substructure and a common similar substructure, the maximal C-mapping is better than the C-mapping and the SSP-mapping, since the number of corresponding vertices between trees by a maximal C-mapping is larger than or equal to that by a C-mapping or an SSP-mapping.

3 Several Similarity Problems

In this section we will define a distance between two trees. Using this distance we can define the LCSS problem, the LSS problem and the distance problem for trees. That is, this distance which will be defined in the following is a general measure to express the similarity between two trees.

3.1 A Distance and the LCSS Problem

Consider a maximal C-mapping from T_a to T_b. The smallest connected part of T_a that includes all of the mapping vertices and that of T_b includes all of the image vertices are considered to be a common similar substructure between T_a and T_b. That is, one maximal C-mapping between T_a and T_b determines one common similar substructure between T_a and T_b. In general, there are many common similar substructures between T_a and T_b, say $[T_a^1, T_b^1], \ldots, [T_a^n, T_b^n]$. If the weight to transform T_a^k to T_b^k is the smallest among those paris of substructures, we can say $[T_a^k, T_b^k]$ is the most similar common substructure between T_a and T_b.

3.1.1 The LCSS Problem for *R-Trees*

To define the LCSS problem for *R-trees* mathematically, we first consider the inserted vertices and the deleted vertices that determined by a maximal C-mapping. Consider a maximal C-mapping M_{xy} from $T_a(x)$ to $T_b(y)$. Let $\mathcal{I}(M_{xy})$ denote the set of all of the mapping vertices of $T_a(x)$, and $\mathcal{J}(M_{xy})$ denote the set of all of the image vertices of $T_b(y)$. Vertex $r_a(r_b)$ is the root of subtree of $T_a(x)(T_b(y))$ that includes $\mathcal{I}(M_{xy})(\mathcal{J}(M_{xy}))$ and the number of its vertices is smallest. If all of the vertices of a subtree of $T_a(x)$(or $T_b(y)$) are deleted(or inserted), it is called that the subtree is deleted(or inserted). Let $Del(M_{xy})$ and $Ins(M_{xy})$ denote "the set of vertices of all of the deleted subtrees of $T_a(x)$" and "the set of vertices of all of the inserted subtrees of $T_b(y)$" determined by M_{xy}, respectively. We can express them as follows.

$$Del(M_{xy}) = \{i \mid V_a(i) \subset (V_a(x) - \mathcal{I}(M_{xy}))\}, \tag{1}$$

$$Ins(M_{xy}) = \{j \mid V_b(j) \subset (V_b(y) - \mathcal{J}(M_{xy}))\}. \tag{2}$$

The part of $T_a(x)$ obtained by removing the vertices of $\left(Del(M_{xy}) \cup \widehat{A}n(x, r_a)\right)$ from $T_a(x)$ and the part of $T_b(y)$ obtained by removing the vertices of $\left(Ins(M_{xy}) \cup \widehat{A}n(y, r_b)\right)$ from $T_b(y)$ is defined as the common similar substructure determined by M_{xy}. We denote it by $\left[\mathbf{S}^a_{M_{xy}}, \mathbf{S}^b_{M_{xy}}\right]$. Let $S^a_{M_{xy}}$ and $S^b_{M_{xy}}$ be the set of vertices of $\mathbf{S}^a_{M_{xy}}$ and that of $\mathbf{S}^b_{M_{xy}}$, respectively. Obviously, $S^a_{M_{xy}}$ and $S^b_{M_{xy}}$ can be expressed as follows.

$$S^a_{M_{xy}} = V_a(x) - Del(M_{xy}) - \widehat{A}n(x, r_a), \tag{3}$$

$$S^b_{M_{xy}} = V_b(y) - Ins(M_{xy}) - \widehat{A}n(y, r_b). \tag{4}$$

One mapping M_{xy} determines one common similar substructure $\left[\mathbf{S}^a_{M_{xy}}, \mathbf{S}^b_{M_{xy}}\right]$. Let p, q_i and r_i be the weights of a substitution, an insertion in $\mathbf{S}^b_{M_{xy}}$ and a deletion in $\mathbf{S}^a_{M_{xy}}$, respectively. Let q_o and r_o be the weights of an insertion in $T_b(y)$ but not in $\mathbf{S}^b_{M_{xy}}$, and a deletion in $T_a(x)$ but not in $\mathbf{S}^a_{M_{xy}}$, respectively. Let $q_i = r_i > q_o = r_o \geq 1$ and $q_o + r_o > p \geq 1$. The reason of this setting will be explained in Subsection 3.4. Assume that a mapping M_{xy} determines n_d deleted vertices in $\mathbf{S}^a_{M_{xy}}$, and n_s substituted vertices and n_i inserted vertices in $\mathbf{S}^b_{M_{xy}}$. The weight to transform $\mathbf{S}^a_{M_{xy}}$ to $\mathbf{S}^b_{M_{xy}}$, denoted by $W(M_{xy})$, is defined as $W(M_{xy}) = r_i n_d + p n_s + q_i n_i$.

Example 1: Consider a maximal C-mapping $M_{xy} = \{(3, y), (5, 5'), (6, 6'), (7, 8'), (8, 1'), (9, 2')\}$ shown in Fig.1(a). We have $Del(M_{xy}) = \{1, 4, 10\}$, $Ins(M_{xy}) = \{3', 4', 7', 9'\}$, $r_a = 3$, $r_b = y$, $\widehat{A}n(x, r_a) = \widehat{A}n(x, 3) = \{x, 2\}$ and $\widehat{A}n(y, r_b) = \widehat{A}n(y, y) = \emptyset$(the empty set). The common similar substructure $\left[\mathbf{S}^a_{M_{xy}}, \mathbf{S}^b_{M_{xy}}\right]$ determined by M_{xy} is shown in Fig.1(b). The weight $W(M_{xy})$ is $\delta(3, y) + \delta(5, 5') + \delta(6, 6') + \delta(7, 8') + \delta(8, 1') + \delta(9, 2')$, where

$$\delta(x, y) \triangleq \begin{cases} 0 & : \textit{the label of } x = \textit{the label of } y, \\ p & : \textit{otherwise}. \end{cases}$$

□

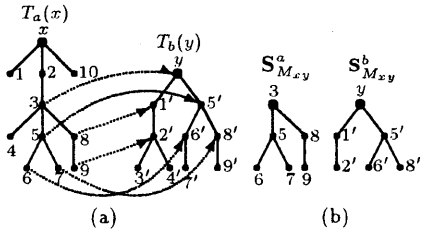

Fig.1 (a) A maximal C-mapping M_{xy}, and (b) the common similar substructure $\left[\mathbf{S}^a_{M_{xy}}, \mathbf{S}^b_{M_{xy}}\right]$ determined by the mapping M_{xy}.

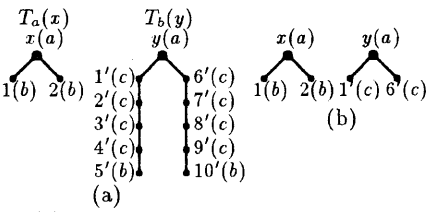

Fig.2 (a) Two *R-trees* and a largest common similar substructure determined by D, and (b) a largest common similar substructure determined by D_C, where $x(a)$ indicades that vertex x has label a.

The distance between two parts of a common similar substructure of $T_a(x)$ and $T_b(y)$, denoted by $D_C(T_a(x), T_b(y))$, is defined as follows.

$$D_C(T_a(x), T_b(y)) = \min_{M_{xy}} \left\{ W(M_{xy}) + \left(|V_a(x)| - |S^a_{M_{xy}}|\right) \cdot r_o + \left(|V_b(y)| - |S^b_{M_{xy}}|\right) \cdot q_o \right\}. \tag{5}$$

Note that there is at least one maximal C-mapping that determines $D_C(T_a(x), T_b(y))$. Let the collection of such mappings be $\{M^1, \ldots, M^d\}$. Those mappings determin d' common similar substructures between two subtrees, where $d' \leq d$. If $|S^a_{M_k}| + |S^b_{M_k}|$ is the largest among those pairs of substructures, $[\mathbf{S}^a_{M_k}, \mathbf{S}^b_{M_k}]$ is called an LCSS of $T_a(x)$ and $T_b(y)$, and denoted by $[\mathbf{S}^a_{xy}, \mathbf{S}^b_{xy}]$.

3.1.2 The LCSS Problem for *Trees*

We use $D_C(T_a(u), T_b(v))$ to indicate the distance between two parts of a common similar substructure between $T_a(u)$ and $T_b(v)$. The distance between two parts of a common similar substructure of T_a and T_b, denoted by $D_C(T_a, T_b)$, is defined as follows.

$$D_C(T_a, T_b) = \min_{u \in V_a, v \in V_b} \{D_C(T_a(u), T_b(v))\}. \tag{6}$$

Note that there is at least one maximal C-mapping that determines $D_C(T_a, T_b)$. Let the collection of such mappings be $\{M^1, \ldots, M^d\}$. Those mappings determin d' common similar substructures between two *trees*, where $d' \leq d$. If $|S^a_{M_k}| + |S^b_{M_k}|$ is the largest among those pairs of substructures, $[\mathbf{S}^a_{M_k}, \mathbf{S}^b_{M_k}]$ is called an LCSS of T_a and T_b, and denoted by $[\mathbf{S}^a, \mathbf{S}^b]$.

3.2 The LSS Problem

In Reference [18] we defined a distance from T_a to "a similar substructure in T_b to T_a", denoted by $D_S(T_a, T_b)$, and defined the LSS problem using $D_S(T_a, T_b)$. Using the notations defined in this paper, we can express $D_S(T_a(x), T_b(y))$ in the following.

$$D_S(T_a(x), T_b(y)) = \min_{M_{xy}} \left\{ W(M_{xy}) + \left(|V_a(x)| - \left|S^a_{M_{xy}}\right| \right) \cdot r_i \right\}. \tag{7}$$

Compare formulae (5) and (7). We can see that the distance $D_S(T_a(x), T_b(y))$ does not contain all of the weights of inserted subtrees, and the weight r_i is equal to r_o. Therefore, setting $r_i = r_o$, $q_o = 0$ and $r_i > p \geq 1$ for formula (5), we can get formula (7) from formula (5). We can say that the distance $D_S(T_a(x), T_b(y))$ is a special case of the distance $D_C(T_a(x), T_b(y))$ which is described in Subsection 3.1. As described in Reference [18], an LSS for *R-trees* is $\mathbf{S}^b_{xy}$ and that for *trees* is $\mathbf{S}^b$ in this paper.

3.3 The Distance Problem

Many papers discussed the distances between trees and their computing methods. In this subsection we will show that the distance which is described in Subsection 3.1 can express the distance between unordered trees based on a maximal C-mapping. Using the notations defined in this paper, we can express the distance between $T_a(x)$ and $T_b(y)$ based on a maximal C-mapping which is defined in Reference [16] in the following.

$$D(T_a(x), T_b(y)) = \min_{M_{xy}} \left\{ W(M_{xy}) + \left(|V_a(x)| - |S^a_{M_{xy}}| \right) \cdot r_i + \left(|V_b(y)| - |S^b_{M_{xy}}| \right) \cdot q_i \right\}. \tag{8}$$

Compare formulae (5) and (8). We can see that the weight r_i is equal to r_o and the weight q_i is equal to q_o in formula (8). Therefore, setting $r_i = r_o = q_i = q_o$ for formula (5), we can get formula (8) from formula (5). We can say that the distance $D(T_a(x), T_b(y))$ is a special case of the distance $D_C(T_a(x), T_b(y))$ which is described in Subsection 3.1.

3.4 The Weights of Transformational Operations

In this subsection, we will explaine the reason of setting "$q_i = r_i > q_o = r_o$ and $q_o + r_o > p$" for expressing the similarity measure between two parts of a common similar substructure.

(i) Consider the distance D_C. Since $q_i = r_i$ and $q_o = r_o$, we have $D_C(T_a(x), T_b(y)) = D_C(T_b(y), T_a(x))$. That is, the distance D_C is symmetric.

(ii) Consider two *R-trees* $T_a(x)$ and $T_b(y)$ such that the labels of all of the vertices of $T_a(x)$ are different from those of $T_b(y)$. If we set $q_o + r_o \leq p$, from the definition of the distance D_C, we have $D_C(T_a(x), T_b(y)) = |V_a(x)| \cdot r_o + |V_b(y)| \cdot q_o$. This means that all of the vertices of $T_a(x)$ are deleted and all of the vertices of $T_b(y)$ are inserted. Therefore, we can only obtain a pair of empty trees as their LCSS. This is unsuitable. This is why we set $q_o + r_o > p$.

(iii) Set $q_i = r_i = q_o = r_o$. As described in Subsection 3.3, the distance D_C denotes the distance between two trees. Consider the two *R-trees* shown in Fig.2(a). We have $D(T_a(x), T_b(y)) = 8q_i$, where one of the maximal C-mappings determines $D(T_a(x), T_b(y))$ is $\{(x, y), (1, 5'), (2, 10')\}$. An LCSS determined by $D(T_a(x), T_b(y))$ is $[T_a(x), T_b(y)]$. If we set $q_i = r_i > q_o = r_o$ as described in Subsection 3.1, we can get an LCSS which is shown in Fig.2(b). We think the LCSS shown in Fig.2(b) is more suitable than $[T_a(x), T_b(y)]$ shown in Fig.2(a). This is why we set $q_i = r_i > q_o = r_o$.

4 Computing Methods of the Distances

In this section we will show three types of the maximal C-mapping between *R-trees*, and show a computing method of the distance between *R-trees* and that between *trees*.

4.1 Types of the Maximal C-mappings

Consider two *R-trees* $T_a(x)$ and $T_b(y)$. From the definition of the maximal C-mapping, the maximal C-mapping satisfies the mapping conditions of Tai mapping. Tai mapping from $T_a(x)$ to $T_b(y)$ can be classified into the following four types [11].

(a1) Vertex x maps to vertex y, and forest $F_a(\bar{x})$ maps to forest $F_b(\bar{y})$.

(a2) Vertex x' of $F_a(\bar{x})$ maps to vertex y, and forest $F_a(\bar{x}')$ maps to forest $F_b(\bar{y})$. The vertices of $T_a(x)$ except those of $T_a(x')$ are deleted.

(a3) Vertex x maps to vertex y' of $F_b(\bar{y})$, and forest $F_a(\bar{x})$ maps to forest $F_b(\bar{y}')$. The vertices of $T_b(y)$ except those of $T_b(y')$ are inserted.

(a4) Vertex x is deleted, vertex y is inserted, and forest $F_a(\bar{x})$ maps to forest $F_b(\bar{y})$.

Assume that a mapping from $F_a(\bar{x})$ to $F_b(\bar{y})$, that from $F_a(\bar{x}')$ to $F_b(\bar{y})$ and that from $F_a(\bar{x})$ to $F_b(\bar{y}')$ are the maximal C-mappings. Then, the mappings of types (a1)–(a3) satisfy the conditionds of the maximal C-mapping. Let M be a maximal C-mapping from $F_a(\bar{x})$ to $F_b(\bar{y})$. Since $M \cup \{(x, y)\}$ is a C-mapping from $T_a(x)$ to $T_b(y)$, a mapping of type (a4) is not a maximal C-mapping. Therefore, the maximal C-mappings from $T_a(x)$ to $T_b(y)$ can be classified into (a1)–(a3) types.

4.2 A Computing Method of the Distance between *R-Trees*

Let $\Delta 1_{xy}$, $\Delta 2_{xy}$ and $\Delta 3_{xy}$ be the minimum value of "$W(M_{xy}) + (|V_a(x)| - |S^a_{M_{xy}}|) \cdot r_o + (|V_b(y)| - |S^b_{M_{xy}}|) \cdot q_o$" for all of the maximal C-mappings M_{xy} of type (a1), that of type (a2) and that of type (a3), respectively. Note that as described in the previous section, the value $W(M_{xy})$ denotes the weight to transform $\mathbf{S}^a_{M_{xy}}$ to $\mathbf{S}^b_{M_{xy}}$. From the definition of the distance D_C, we have the following formula.

$$D_C(T_a(x), T_b(y)) = \min\{\Delta 1_{xy},\ \Delta 2_{xy},\ \Delta 3_{xy}\}. \tag{9}$$

A common similar substructure $\left[\mathbf{S}^a_{M_{xy}}, \mathbf{S}^b_{M_{xy}}\right]$ determined by a maximal C-mapping M_{xy} of type (a2) is shown in Fig.3(a). In this case we have

$$\Delta 2_{xy} = \min_{x' \in V_a(\bar{x})} \{\Delta 1_{x'y} + (|V_a(x)| - |V_a(x')|) \cdot r_o\}. \tag{10}$$

Consider a maximal C-mapping M_{xy} of type (a3). A common similar substructure $\left[\mathbf{S}^a_{M_{xy}}, \mathbf{S}^b_{M_{xy}}\right]$ determined by this mapping is shown in Fig.3(b). Therefore, we have

$$\Delta 3_{xy} = \min_{y' \in V_b(\bar{y})} \{\Delta 1_{xy'} + (|V_b(y)| - |V_b(y')|) \cdot q_o\}. \tag{11}$$

Consider two *R-trees* $T_a(x)$ and $T_b(y)$. If x maps to y, the SSP-mapping from $F_a(\bar{x})$ to $F_b(\bar{y})$ has the following characteristics.

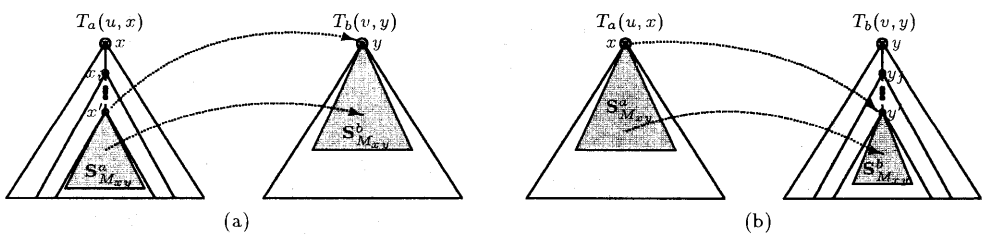

Fig.3 (a) A common similar substructure $\left[\mathbf{S}^a_{M_{xy}}, \mathbf{S}^b_{M_{xy}}\right]$ determined by a maximal C-mapping M_{xy} of type (a2), and (b) that determined by a maximal C-mapping M_{xy} of type (a3).

"For any vertices x_{i_1}, x_{i_2}, y_{j_1} and $y_{j_2}(i_1 \neq i_2, x_{i_1}, x_{i_2} \in Ch(x),\ j_1 \neq j_2, y_{j_1}, y_{j_2} \in Ch(y))$, both $T_a(x_{i_1})$ and $T_a(x_{i_2})$ cannot map to $T_b(y_{j_1})$ at the same time, and $T_a(x_{i_1})$ cannot map to both $T_b(y_{j_1})$ and $T_b(y_{j_2})$ at the same time.[10]"(*1)

Since the maximal C-mapping is a subclass of the SSP-mapping, the maximal C-mapping holds the characteristics (*1). Let $M_{\bar{x}\bar{y}}$ denote a maximal C-mapping from $F_a(\bar{x})$ to $F_b(\bar{y})$ in the case that x maps to y. If x maps to y, we have $M_{\bar{x}\bar{y}} = M_{xy} - \{(x,y)\}$. Let $\langle x_i, y_j \rangle$ denote that $T_a(x_i)$ maps to $T_b(y_j)$. From the characteristics (*1), we can express $M_{\bar{x}\bar{y}}$ using a set C_{xy} of $\langle x_i, y_j \rangle$, that is, $C_{xy} = \{\langle x_{i_1}, y_{j_1} \rangle, \ \cdots\ , \langle x_{i_k}, y_{j_k} \rangle\}(x_{i_h} \in Ch(x),\ y_{j_h} \in Ch(y),\ 1 \leq h \leq k)$. Then, a maximal C-mapping M_{xy} can be expressed as follows.

$$M_{xy} = \{(x,y)\} \cup M_{\bar{x}\bar{y}} = \{(x,y)\} \cup \left(\bigcup_{\langle x_i, y_j \rangle \in C_{xy}} M_{x_i y_j} \right). \tag{12}$$

Consider the set C_{xy} and the maximal C-mapping M_{xy} in the case that x maps to y. Assume that "subtree $T_a(x_s)(x_s \in Ch(x))$ is deleted and subtree $T_b(y_t)(y_t \in Ch(y))$ is inserted"(*2). Since $M_{xy} \cup M_{x_s y_t}$ is a C-mapping from $T_a(x)$ to $T_b(y)$, M_{xy} is not a maximal C-mapping. Therefore, the assumption (*2) is incorrect. Then, we have

$$|C_{xy}| = \min\{|Ch(x)|, |Ch(y)|\}. \tag{13}$$

Lemma 1: Assume that $\Delta 1_{xy}$, all of the $\Delta 1_{x,y}$ and $\Delta 1_{xy_j}(x_i \in Ch(x),\ y_j \in Ch(y))$ have been computed. The distance $D_C(T_a(x), T_b(y))$ can be computed by the following formula.

$$D_C(T_a(x), T_b(y)) = \min \begin{cases} \Delta 1_{xy}, \\ \min_{x_i \in Ch(x)} \{D_C(T_a(x_i), T_b(y)) + (|V_a(x)| - |V_a(x_i)|) \cdot r_o\}, \\ \min_{y_i \in Ch(y)} \{D_C(T_a(x), T_b(y_j)) + (|V_b(v,y)| - |V_b(v,y')|) \cdot q_o\}. \end{cases} \tag{14}$$

□

Hereafter we will describe a computing method of $\Delta 1_{xy}$. Consider the weight $W(M_{xy})$ in the case that x maps to y. Assume that $T_a(x_i)(x_i \in Ch(x))$ maps to $T_b(y_j)(y_j \in Ch(y))$ shown in Fig.4. Since x maps to y, the vertives of $\widehat{A}n(x_i, r_a)$ belong to $\mathbf{S}^a_{M_{xy}}$(Fig.4(a)) and that of $\widehat{A}n(y_j, r_b)$ belong to $\mathbf{S}^b_{M_{xy}}$(Fig.4(b)). From the characteristics (*1) and formula (12), we can express $W(M_{xy})$ in the case that x maps to y, and $\Delta 1_{xy}$ as follows.

$$W(M_{xy}) = \delta(x,y) + \sum_{\langle x_i, y_j \rangle \in C_{xy}} \left(W(M_{x_i y_j}) + \left|\widehat{A}n(x_i, r_a)\right| \cdot r_i + \left|\widehat{A}n(y_j, r_b)\right| \cdot q_i \right), \tag{15}$$

$$\begin{aligned} \Delta 1_{xy} \ = \ & \min_{C_{xy}} \left\{ \sum_{\langle x_i, y_j \rangle \in C_{xy}} \widehat{D}_C(T_a(x_i), T_b(y_j)) + \sum_{\langle x_i, y_j \rangle \notin C_{xy}} |V_a(x_i)| \cdot r_o + \sum_{\langle x_i, y_j \rangle \notin C_{xy}} |V_b(y_j)| \cdot q_o \right\} \\ & + \delta(x,y), \end{aligned} \tag{16}$$

where

$$\begin{aligned} \widehat{D}_C(T_a(x_i), T_b(y_j)) \ \triangleq \ & \min_{M_{x_i y_j}} \left\{ W(M_{x_i y_j}) + \left|\widehat{A}n(x_i, r_a)\right| \cdot (r_i - r_o) + \left|\widehat{A}n(y_j, r_b)\right| \cdot (q_i - q_o) \right. \\ & \left. + \left(|V_a(x_i)| - |S^a_{M_{x_i y_j}}|\right) \cdot r_o + \left(|V_b(y_j)| - |S^b_{M_{x_i y_j}}|\right) \cdot q_o \right\}. \end{aligned} \tag{17}$$

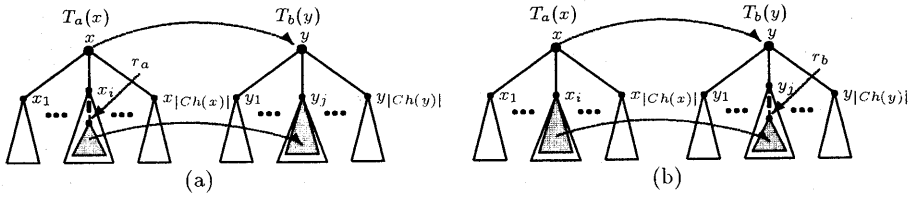

Fig.4 Illustration of the weight M_{xy} in the case that x maps to y.

Note that either $\widehat{An}(x_i, r_a)$ or $\widehat{An}(y_j, r_b)$ is the empty set. If both of them are not the empty sets, from the definition of the maximal C-mapping the vertices of $\widehat{An}(x_i, r_a)$ must map to those of $\widehat{An}(y_j, r_b)$. Before discussing a computing method of $\Delta 1_{xy}$, we show a computing method of $\widehat{D}_C(T_a(x), T_b(y))$. Similar to Lemma 1 we have Lemma 2. We omit the proof for the limitation of pages.

Lemma 2: Assume that $\Delta 1_{xy}$, all of the $\Delta 1_{x_i y}$ and $\Delta 1_{x y_j} (x_i \in Ch(x),\ y_j \in Ch(y))$ have been computed. $\widehat{D}_C(T_a(x), T_b(y))$ can be computed by the following formula.

$$\widehat{D}_C(T_a(x), T_b(y)) = \min \begin{cases} \Delta 1_{xy}, \\ \min_{x_i \in Ch(x)} \left\{ \widehat{D}_C(T_a(x_i), T_b(y)) + (|V_a(x)| - |V_a(x_i)| - 1) \cdot r_o + r_i \right\}, \\ \min_{y_i \in Ch(y)} \left\{ \widehat{D}_C(T_a(x), T_b(y_j)) + (|V_b(y)| - |V_b(y')| - 1) \cdot q_o + q_i \right\}. \end{cases} \tag{18}$$

□

Consider a computing method of $\Delta 1_{xy}$ based on formula (16). Create a bipartite graph $G(A, B, E)$ from $T_a(x)$ and $T_b(y)$. A and B are the sets of vertices of G, E is the set of edges of G. $A = \{\alpha_1, \ \ldots\ , \alpha_{|Ch(x)|}\}$ and $B = \{\beta_1, \ \ldots\ , \beta_{|Ch(y)|}\}$. $\alpha_i(\beta_j)$ denotes subtree $T_a(x_i)(T_b(y_j))(x_i \in Ch(x),\ y_j \in Ch(y))$. Let e_{ij} denote an edge between α_i and β_j, and $E = \{e_{ij} | i = 1, \ \ldots\ , |A|,\ j = 1, \ \ldots\ , |B|\}$. The weight of edge e_{ij}, denoted by $w(e_{ij})$, is defined as follows.

$$w(e_{ij}) = |V_a(x_i)| \cdot r_o + |V_b(y_j)| \cdot q_o - \widehat{D}_C(T_a(x_i), T_b(y_j)). \tag{19}$$

From the definition of $\widehat{D}_C$ and the assumption $q_o + r_o > p$, we have $\widehat{D}_C(T_a(x_i), T_b(y_j)) \leq |V_a(x_i)| \cdot r_o + |V_b(y_j)| \cdot q_o$. That is, $w(e_{ij})$ is nonnegative. Let $\mathcal{M}$ denote one of the weight matchings of G. If vertex α_i of G maps to vertex β_j, it denotes $T_a(x_i)$ maps to $T_b(y_j)$. From formula (12), we can see that a matching of G expresses a C_{xy}. Let $\mathcal{M}_{max}$ denote one of the maximal weight matchings of G, and $W(\mathcal{M}_{max})$ denote its weight. We have Lemma 3 for computing $\Delta 1_{xy}$.

Lemma 3: Assume that all of $\widehat{D}_C(T_a(x_i), T_b(y_j))(x_i \in Ch(x),\ y_j \in Ch(y))$ have been computed. We can compute $\Delta 1_{xy}$ by the following formula.

$$\Delta 1_{xy} = \sum_{x_i \in Ch(x)} |V_a(x_i)| \cdot r_o + \sum_{y_j \in Ch(y)} |V_b(y_j)| \cdot q_o + \delta(x, y) - W(\mathcal{M}_{max}). \tag{20}$$

□

Let $m_a(m_b)$ and $N_a(N_b)$ be the largest degree of a vertex of $T_a(T_b)$ and the number of vertices of $T_a(T_b)$, respectively. Using Lemma 1—Lemma 3, we can get a procedure Dist-*subtree* and an algorithm Dist-*R-tree* for computing the distance between two subtrees and that between two *R-trees*, respectively. Since the time complexity for computing $\Delta 1_{xy}$ is $O(m_a^2 m_b)$, the time complexity of the procedure Dist-*subtree* and that of the algorithm Dist-*R-tree* are $O(m_a^2 m_b)$ and $O(m_a N_a N_b)$, respectively.

4.3 A Computing Method of the Distance for *Trees*

If all of the distances $D_C(T_a(u), T_b(v))$ are given, we can compute $D_C(T_a, T_b)$ by formula (6). In Reference [16], an efficient algorithm for computing the distance $D(T_a, T_b)$ was proposed. Modifing this algorithm in the following way, we can obtain an algorithm Dist-*tree* for computing $D_C(T_a, T_b)$. The time complexity of the algorithm is $O(m_a m_b N_a N_b)$.

(b1) Replace the notations concerning the distances based on the maximal C-mapping with those concerning the distances described in this paper.

(b2) Replace the procedure for computing the distance between subtrees with the procedure Dist-*subtree*.

5 Conclusions

In this paper we defined the LCSS problem for unordered trees and gave a computing method of the distance which determines the LCSS between *R-trees* and that between *trees*. The time complexity of the computing method for *R-trees* is $O(m_a N_a N_b)$, and that for *trees* is $O(m_a m_b N_a N_b)$. Using the maximal C-mappings that determine the distance, it is easy to make an algorithm for finding one of the LCSSs(or LSSs) or enumerating all of the LCSSs(or LSSs) between two unordered trees. The computing methods can be applied to structure-activity studies and structure comparison problems. One of the future problems is to make a file for similar structure search using the proposed metrics.

References

[1] P.Willett, Similarity and clustering in chemical information systems, Research Studies Press, 1987.

[2] K.J.Lipscomb, M.F.Lynch and P.Willett, "Chemical structure processing," Ann. Rev. Inf. Sci. Technol., 24, pp.189-238, 1989.

[3] M.G.Hicks and C.Jochum, "Substructure search system. 1. Performance comparison of the MACCS, DARC, HTSS, CAS registry MVSSS, and S4 substructure search systems," J. Chem. Inf. Comput. Sci., vol.30, pp.191-199, 1990.

[4] W.A.Warr, Chemical structures 2, Springer, 1993.

[5] Y.Takahashi, Y.Satoh, H.Suzuki and S.Sasaki, "Recognition of largest common structural fragment among a variety of chemical structures," Analytical Sciences, no.3, pp.23-28, 1987.

[6] T.Akutsu, "An RNC algorithm for finding a largest common subtree of two trees," IEICE Trans. INF. & SYST., vol.E75-D, no,1, pp.95-101, 1992.

[7] S.Masuda, I.Mori and E.Tanaka, "Algorithms for finding the largest common subgraphs of two trees," IEICE Trans., vol.J77-A, no.3, pp.460-470, 1994.

[8] K.C.Tai, "The tree-to-tree correction problem," JACM, vol.26, no.3, pp.422-433, 1979.

[9] E.Tanaka and K.Tanaka, (a)"A tree metric and its computing method," IEICE Trans., vol.J65-D, no.5, pp.511-518, 1982. (b)"Correction to "A tree metric and its computing method"," IEICE Trans., vol.J76-D-I, no.11, p.635, 1993.

[10] E.Tanaka, "The metric between rooted and ordered trees based on strongly structure preserving mapping and its computing method," IEICE Trans., vol.J67-D, no.6, pp.722-723, 1984.

[11] K.Ohmori and E.Tanaka, "A unified view on tree metrics," Proc. Workshop on Syntactic and Structural Pattern Recognition, Barcelona, pp.85-100, 1986. (Eds. G.Ferraté et al. Syntactic and Structural Pattern Recognition, Springer 1988.)

[12] S.M.Liu, E.Tanaka and S.Masuda, "The distances between unrooted and cyclically ordered trees and their computing methods," IEICE Trans. INF. & SYST., vol.E77-D, no.10, pp.1094-1105, Oct. 1994.

[13] E.Tanaka, "Metrics between trees embedded in a plane and their computing methods," IEICE Trans. vol.E79-A, no.4, pp.441-447, April, 1996.

[14] T.Muguruma, E.Tanaka and S.Masuda, "A metric between unrooted and unordered trees and its top-down computing method," IEICE Trans. INF. & SYST., vol.E77-D, no.5, pp.555-566, 1994.

[15] E.Tanaka, "A metric between unrooted and unordered trees and its bottom-up computing method," IEEE Trans. Pattern Anal. & Mach. Intell., vol.16, no.12, pp.1233-1238, 1994.

[16] S.M.Liu and E.Tanaka, "Algorithms for computing the distances between unordered trees," IEICE Trans., vol.J78-A, no.10, pp.1358-1371, Oct. 1995.

[17] S.M.Liu, E.Tanaka and S.Masuda, "Largest similar substructure problems for trees and their algorithms," IEICE Trans., vol.J78-A, no.10, pp.1348-1357, Oct. 1995.

[18] S.M.Liu and E.Tanaka, "Efficient algorithms for finding largest similar substructures in unordered trees," IEICE Trans. vol.E79-A, no.4, pp.428-440, April, 1996.

[19] S.M.Liu and E.Tanaka, "A largest common similar substructure problem for trees embedded in a plane," Technical Report of IEICE, COMP 95-74, pp.11-20, JAN. 1996.