アルゴリズム 2－4
（1988. 7. 25）

木の距離に関する統一的見解

大森　勝美　　　田中　栄一

宇都宮大学　情報工学科

本報告は既に提案された木の距離の１つの統一的な取扱いについて述べている。木の距離には、Tai 距離、構造保存写像に基づく距離、強構造保存写像に基づく距離、Selkow 距離等がある。「写像によって定まる直近の先祖」の概念を導入すると、これらの距離が見事に分類され、また弱構造保存写像に基づく類似度が定義できる。これらの距離や類似度の計算法についても述べている。

A UNIFIED VIEW ON TREE METRICS

Katsumi OHMORI　and　Eiichi TANAKA

Department of Information Science, Utsunomiya University

2753 Ishiimachi, Utsunomiya, 321 Japan

This paper describes a unified treatise on tree metrics already proposed. These are the Tai metric, the metric defined by the structure preserving mapping, the metric by the strongly structure preserving mapping and the Selkow metric. By introducing "the nearest ancestor determined by a mapping", these metrics are classified in a simple way and the similarity by the weakly structure preserving mapping can be introduced. The computing methods for these metrics and similarity are also stated.

1. Introduction

A considerable amount of work has been done on sequence comparison for problems such as string correction, molecular biology, human speech, codes and error control, and so on [1]. Tree metrics have been also studied [3-14]. Potential applications of tree metrics include the areas of behavioral science [3], data base [5], clustering [7], waveform correlation [9], and so on. Among various tree metrics, Tai's metric [6] seems to be the most fundamental one. Selkow's metric [4] is a strictly restricted Tai's metric. Between these two metrics, several metrics have been defined [10-12]. However, the interrelation between these metrics is not known clearly. In this paper, by introducing a concept "the nearest ancestor determined by a mapping", we give a unified point of view for tree metrics. Furthermore, we propose a new similarity between two trees.

2. Definitions

In this paper all trees we discuss are rooted, ordered, and labeled.

[Definition 1] Numbering in preorder. A tree T is numbered from one in preorder for nodes of T. A positive integer represents a node.

[Definieion 2] Notations

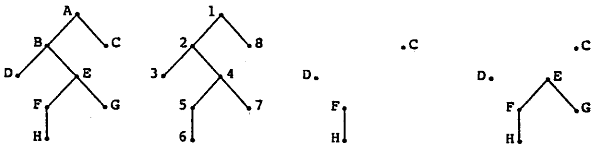
T(k) denotes a subtree of a tree T whose root is k. Ch(k) and An(k) denote the set of children of k and that of ancestors of k, respectively. N(k) denotes the number of nodes of T(k). Let N mean N(1). t(k) denotes the label of node k. The rightmost leaf of T(k) is called the end leaf of T(k) and denoted by el(k).

[Definition 3] Separation of nodes and subtrees.

For any nodes k_1 and k_2 ($k_1 \neq k_2$), k_1 and k_2 are said to be separated if k_1 is neither an ancestor of k_2 nor a descendant of k_2. Furthermore, if k_1 and k_2 are separated, $T(k_1)$ and $T(_2)$ are said to be separated.

[Definition 4] Forest

A sequence of separate subtrees $T(k_1)$, $T(k_2)$, ..., $T(k_n)$ ($k_1<k_2<...<k_n$) is called a forest of T. A subtree T(k) is also a forest. If the forest is composed of all the nodes from k to m($k \leq m$), it is denoted by F(k,m).



(a) Labeled tree T. (b) Preorder numbering for T. (c) Forest T(3),T(5),T(8). (d) Forest F(3,8).

Fig. 1 Tree and Forest.

In Fig.1, Ch(4)={5,7}, An(4)={1,2}, N(4)={1,2}, N(4)=4, t(4)="E", el(4)=7.

Before discussing tree metrics, let us review briefly the weighted Levenshtein distance [2].

The following three operations to transform one string into the other and their weights (or costs) are considered: (1) substitute another symbol for a symbol (cost p); (2) insert an extra symbol (cost q); and (3) delete a symbol (cost r). In general, p<q+r. Because if p>q+r, a substitution is always regarded as a pair of an insertion and a deletion.

[Definition 5] The weighted Levenshtein distance.

Let $A=a_1a_2...a_m$ and $B=b_1b_2...b_n$ be two finite strings of symbols. A mapping M from A to B is a set of ordered pairs (i,j) ($1 \leq i \leq m$, $1 \leq j \leq n$). Let $I=\{i|(i,j) \in M\}$ and $J=\{j|(i,j) \in M\}$. Then, M represents a transformation from A to B under the following interpretation:

(1) For $(i,j) \in M$, if $a_i \neq b_j$, b_j is substituted for a_i;

(2) If $j \notin J$, b_j is inserted;

(3) If $i \notin I$, a_i is deleted.

Let M be the set of all these possible mappings from A to B. THen, the minimum cost from A to B, denoted by D(A,B), can be defined as follows:

$$D(A,B) = \min \{ p(i,j) + (n-|J|)*q + (m-|I|)*r \}. \quad (2\text{-}1)$$

where |I| denotes the number of elements in I, and

$$p(i,j) = \begin{cases} 0: \text{ if } a_i=b_j; \\ p: \text{ if } a_i \neq b_j. \end{cases}$$

D(A,B) is called the weighted Levenshtein distance (WLD) from A to B if the mapping M satisfies the following conditions.

For any pairs (i_1,j_1), $(i_2,j_2) \in M$,

(1) $i_1=i_2$ iff $j_1=j_2$;

(2) $i_1<i_2$ iff $j_1<j_2$.

D(A,B) can be computed by applying the following formula, iteratively:

$$d[i,j] = \min \begin{cases} d[i-1,j-1]+p(i,j), \\ d[\ i\ ,j-1]+q, \\ d[i-1,\ j\ ]+r, \end{cases} \qquad (2\text{-}2)$$

where

$d[0,j]=j*q \ (0 \le j \le n)$,

$d[i,0]=i*r \ (0 \le i \le m)$.

Then, $D(A,B) = d[m,n]$.

The time and space complexities to compute D(A,B) are proportional to mn.

In mathematical literatures, the word "distance" is ordinarily used to indicate a function 'd' which satisfies the metric axtioms:

For all A, B and C,

(1) Nonnegative property : $d(A,B) \ge 0$;

(2) Zero property : $d(A,B)=0$ iff $A=B$;

(3) Symmetry : $d(A,B)=d(B,A)$;

(4) Triangle inequality : $d(A,C) \le d(A,B)+d(B,C)$.

WLD satisfies the metric axioms if the insertion cost equals to the deletion cost, that is, q=r.

Let us turn to tree metrics. We will use a similar approach to define transformation between trees and tree-to-tree distances. The three edit operations on a labeled node, that is, substitution (cost p), insertion (cost q) and deletion (cost r) are considered. A mapping between trees is regarded as a transformation between trees. $(i,j) \in M$ means that a labeled node i mapped to a labeled node j $(1 \le i \le N_A,\ 1 \le j \le N_B)$. Since the mapping conditions of WLD have no information on tree structures, a mapping M from T_A to T_B must satisfy conditions about tree structures. Tai proposed the following mapping.

[Definition 6] The mapping conditions of Tai's distance [6].

For any pairs $(i_1,j_1),(i_2,j_2) \in M$,

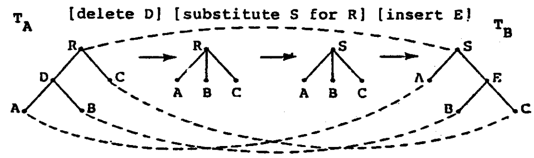


Fig.2 Transformation and mapping from T_A to T_B.

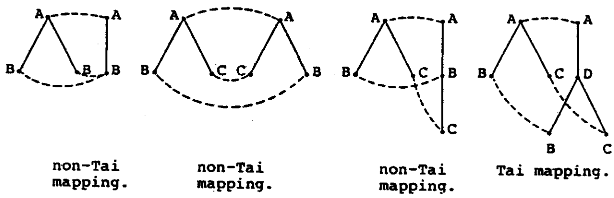


Fig.3 Examples of non-Tai mappings and a Tai mapping.

(1) $i_1=i_2$ iff $j_1=j_2$;

(2) $i_1<i_2$ iff $j_1<j_2$;

(3) $i_1 \in An_A(i_2)$ iff $j_1 \in An_B(j_2)$.

Here we call it the Tai mapping. (1) means one-to-one correspondence, (2) means that a cross mapping is prohibited and (3) means that the ancestor-descendant relation does not change. If $i_1 \in An_A(i_2)$ and $j_1 \in An_B(j_2)$, then it is obvious from preorder numbering that $i_1<i_2$ and $j_1<j_2$. That is, (2) states that the order relation between separate nodes is preserved.

The minimum cost under the Tai mapping is called the Tai distance. The Tai distance $D(T_A,T_B)$ satisfies the metric axioms in case of q=r.

3. Computation of the Tai distance

In this section, we propose a simple algorithm for computing the Tai distance.

The word "mapping" and M mean a Tai mapping in this section. The set I and J, defined in definition 5, are again used hereafter.

[Lemma 1]

Let R_A and R_B be the roots of T_A and T_B, respectively. Suppose M is the minimum cost mapping from T_A to T_B.

Then $R_A \in I$ and/or $R_B \in J$.

(Proof) Assume, for the sake of contradiction, that $R_A \notin I$, $R_B \notin J$ and M is the minimum cost mapping. Let An(I) $=\{An_A(i) \mid i \in I\}$. Then $An(I) \neq \{\}$ because $R_A \in An(I)$. Similarly, $An(J) \neq \{\}$. Let $M'=M \{(i',j')\}$ for $i' \in An(I)$ and $j' \in An(J)$. Since M is a Tai mapping, M' is also a Tai mapping. Assume that the cost of the transformation represented by M be c+q+r, that is, i' is deleted and j' is inserted. Then the cost of the transformation by M' is c+p, since j' is substituted for i'. Apparently, c+p<c+q+r. This contradicts our assumption that M is the minimum cost mapping.

[Definition 7] Substitution between subtrees.

If $(i,j) \in M$, we say that subtree $T_B(j)$ is substituted for subtree $T_A(i)$. The mapping condition (3) is apparently equivalent to the following.

For any (i,j), $(i',j') \in M$ $(i \neq i', j \neq j')$,

i' is in $T_A(i)$ iff j' is in $T_B(j)$.

Therefore, a substitution of $T_B(j)$ for $T_A(i)$ means that a mapping from $T_A(i)$ to $T_B(j)$ meets the mapping condition (3).

From lemma 1 and the above definition, one of the three cases (a), (b) and (c) gives the minimum cost mapping from $T_A(x)$ to $T_B(y)$:

(a) $T_B(y)$ is substituted for $T_A(x)$;

(b) One subtree of $T_B(y)$ is substituted for $T_A(x)$;

(c) $T_B(y)$ is substituted for one subtree of $T_A(x)$.

Let $\Delta a(x,y)$, $\Delta b(x,y)$ and $\Delta c(x,y)$ be the minimum costs in case (a), (b) and (c), respectively. Then the Tai distance $D(T_A(x),T_B(y))$, which is stored in D[x,y], is the minimum value of $\Delta a(x,y)$, $\Delta b(x,y)$ and $\Delta c(x,y)$.

The main algorithm for computing the Tai distance is as follows:

[Main algorithm]

for $x:=N_A$ downto 1 do
for $y:=N_B$ downto 1 do
begin

if {x is a leaf} then (3-1-1)

$$D[x,y] := \begin{cases} (N_B(y)-1)*q: & \text{if } t_A(x) \in Lab_B(y); \\ (N_B(y)-1)*q+p: & \text{if } t_A(x) \notin Lab_B(y); \end{cases}$$

if {y is a leaf} then (3-1-2)

$$D[x,y] := \begin{cases} (N_A(x)-1)*r; & \text{if } t_B(y) \in Lab_A(x); \\ (N_A(x)-1)*r+p; & \text{if } t_B(y) \notin Lab_A(x); \end{cases}$$

if {Neither x nor y is a leaf} then (3-1-3)

$D[x,y] := \min \{ \Delta a(x,y), \Delta b(x,y), \Delta c(x,y)\};$

end;

$D(T_A,T_B):=D[1,1];$

where $Lab(k)=\{t(k') \mid k' \text{ in } T(k)\}$.

If x is a leaf, an arbitrary node of $T_B(y)$ can be substituted for x. The remaining nodes of $T_B(y)$ are considered to be inserted. Hence, we get the formula (3-1-1). If y is a leaf, we have the formula (3-1-2).

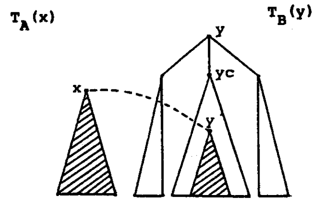


Fig.4 Substitution of $T_B(y')$ for $T_A(x)$.

Let us consider the case (b) (See Fig.4). Let $(x,y') \in M$, y' in $T_B(yc)$ and $yc \in Ch_B(y)$. Then the remaining subgraph by removing $T_B(yc)$ from $T_B(y)$ are inserted. Furthermore, the minimum cost mapping from $T_A(x)$ to $T_B(y')$ is identical with that from $T_A(x)$ to $T_B(yc)$. Therefore, $\Delta b(x,y)$ can be computed by the following formula:

$$\Delta b(x,y) = \min_{y_j \in Ch_B(x)} \{D[x,y_j]+(N_B(y)-N_B(y_j))*q\} \quad (3-2)$$

Similarly,

$$\Delta c(x,y) = \min_{x_i \in Ch_A(x)} \{D[x_i,y]+(N_A(x)-N_A(x_i))*r\} \quad (3-3)$$

Let us investigate the case (a). As in Fig.5, using the mapping conditions and definition 7, substitution of $T_B(y)$ for $T_A(x)$ can be decomposed into that of $T_B(j_1)$ for $T_A(i_1)$, that of $T_B(j_2)$ for $T_A(i_2)$, ..., that of $T_B(j_n)$ for $T_A(i_n)$ such that both $T_A(i_1)$, $T_A(i_2)$, .., $T_A(i_n)$ and $T_B(j_1)$, $T_B(j_2)$, .., $T_B(j_n)$ are forests, and any

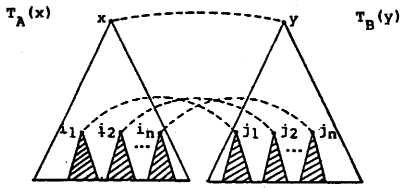


Fig.5 Substitution of $T_B(y)$ for $T_A(x)$.

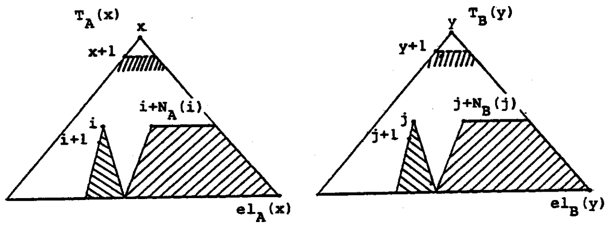


Fig.6 Computation of $\Delta a(x,y)$ for the Tai distance.

node outside these forests is inserted or deleted. Hence, to compute $\Delta a(x,y)$, we must check up the all possible pairs of forests. However, we can use the technique of dynamic programming.

[Computation of $\Delta a(x,y)$: the Tai distance]

Each $d[i,j]$ $(x<i\leq el_A(x), y<j\leq el_B(y))$ stores the distance from forest $F_A(i, el_A(x))$ to forest $F_B(j, el_B(y))$. As boundary conditions, $d[el_A(x)+1,j]$ is the cost of insertions of nodes j, j+1,

cost of deletions of nodes i, i+1, .., $el_A(x)$. $\Delta a(x,y)$ can be computed by applying the following formula iteratively:

$\delta 1 = d[i+N_A(i), j+N_B(j)] + D[i,j];$ (3-4-1)

$\delta 2 = d[\ i\ , j+1] + q;$ (3-4-2)

$\delta 3 = d[i+1,\ j\ ] + r;$ (3-4-3)

$d[i,j] = \min\{\ \delta 1,\ \delta 2,\ \delta 3\},$ (3-4)

where, the boundary conditions are

$d[el_A(x)+1, j] = \{el_B(y)+1-j\} * q$ $(y<j\leq el_B(y)+1),$

$d[i, el_B(y)+1] = \{el_A(x)+1-i\} * r$ $(x<i\leq el_A(x)+1).$

Let $p(x,) = \begin{cases} 0: & \text{if } t_A(x)=t_B(y); \\ p: & \text{if } t_A(x)\neq t_B(y). \end{cases}$

Then,

$\Delta a(x,y) = d[x+1,y+1] + p(x,y).$

The formula of $\Delta a(x,y)$ is a straightforward extension of that of WLD.

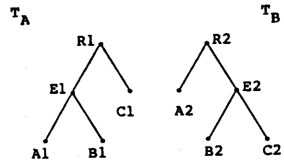


Fig.7 Mapping from T_A to T_B.

4. Subclasses of the Tai mapping

There are some problems that the Tai mapping can not be applied to. Consider a Tai mapping between classification trees T_A and T_B in Fig.7. In T_A, B1 is closer to A1 than to C1. On the other hand, B2 is closer to C2 than to A2 in T_B. Therefore, the Tai mapping is not appropriate to classification trees.

To introduce other mappings, we define a special ancestor, called "the nearest ancestor determined by a mapping".

[Definition 8] The nearest ancestor determined by a mapping.

Let m be a Tai mapping. For some $i \nmid I$, let $i' \nmid I$ be any separate node of i. The nearest common ancestor of i and i' is called the nearest ancestor of i determined by maping M, and denoted by $Nam_A(i)$. If i' can not be determined, $Nam_A(i)$ can not be also determined. $Nam_B(j)$ is defined in the same way.

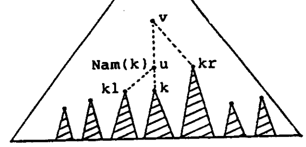


Fig.8 Nam(k).

Let us illustrate Nam using Fig.8. Let kl and kr be left and right neighbor separate nodes of k such that k, kl, kr $\in$ I (or $\in$J), respectively. Let u and v be the nearest common ancestors of k & kl and k & kr, respectively. Since u and v have a common descendant k, u and v are not separated. Then Nam(k) is the nearest node between u and v.

By putting restrictions on insertion and deletion of Nam, we propose the following mappings.

[Definition 9] WSPM, ISPM, DSPM, SSPM.

Let M be a Tai mapping. For any $(i,j)\in M$, if $Nam_A(i)$ and $Nam_B(j)$ can be determined,

(a) $Nam_A(i)\in I$ or $Nam_B(j)\in J$;

(b) $Nam_A(i)\in I$;

(c) $Nam_B(j)\in J$;

(d) $Nam_A(i)\in I$ and $Nam_B(j)\in J$.

Then, mappings satisfying (a), (b), (c) and (d) are called the weakly structure preserving mapping (WSPM), the structure preserving mapping for insertion (ISPM), the structure preserving mapping for deletion (DSPM), and the strongly structure preserving mapping (SSPM), respectively.

The minimum cost under WSPM is symmetric. However, since the triangle inequality is not satisfied, we call it the WSPM "similarity". The minimum costs under other mappings are called the "distance". Neither the ISPM distance nor the DSPM distance is symmetric. If q=r, the SSPM distance satisfies the metric axioms.

Since these four mappings belong to the Tai mapping, $\Delta b(x,y)$ and $\Delta c(x,y)$ for the Tai distance are available to compute d[x,y]. We will explain a(x,y).

As illustrated in the previous section, substitution of $T_B(y)$ for $T_A(x)$ is decomposed into that of forest $T_B(j_1)$, $T_B(j_2)$, .., $T_B(j_n)$ for forest $T_A(i_1)$, $T_A(i_2)$, .., $T_A(i_n)$. In the case of WSPM, by definition 9(a), the following holds:

$Nam_A(i_1)=x$ or $Nam_B(j_1)=y$, ...,

$Nam_A(i_k)=x$ or $Nam_B(j_k)=y$, ...,

$Nam_A(i_n)=x$ or $Nam_B(j_n)=y$.

Without loss of generality, assume $Nam_A(i_k)=x$. Let xc be a child of x such that i_k in $T_A(xc)$. Then, $i_h(h\neq k)$ is not in $T_A(xc)$. Because if i_h is in $T_A(xc)$, xc is a common ancestor of i_k and i_h, and x is not $Nam_A(i_k)$. In order to ensure that only i_k is in $T_A(xc)$, we must use $D[xc,y_k]$, not $D[i_k,j_k]$ in computing $\Delta a(x,y)$. Therefore, we can get the formula of $\Delta a(x,y)$ by only replacing (3-4-1) with the following formula.

[$\Delta a(x,y)$ for the WSPM similarity]

$$\delta 1=\begin{cases} d[i+N_A(i),j+N_B(j)] + D[i,j]: \\ \qquad \text{if } i\in Ch_A(x) \text{ or } j\in Ch_B(y); \\ \text{infinite} \qquad : \\ \qquad \text{otherwise.} \end{cases} \qquad (4\text{-}1\text{-}1)$$

Similarly, we can compute other a(x,y) by replacing (3-4-1) with the following formulae.

[$\Delta a(x,y)$ for the ISPM distance]

$$\delta 1=\begin{cases} d[i+N_A(i),j+N_B(j)] + D[i,j]: \\ \qquad \text{if } i\in Ch_A(x); \\ \text{infinite} \qquad : \\ \qquad \text{otherwise.} \end{cases} \qquad (4\text{-}2\text{-}1)$$

[$\Delta a(x,y)$ for the DSPM distance]

$$\delta 1=\begin{cases} d[i+N_A(i),j+N_B(j)] + D[i,j]: \\ \qquad \text{if } j\in Ch_B(y); \\ \text{infinite} \qquad : \\ \qquad \text{otherwise.} \end{cases} \qquad (4\text{-}3\text{-}1)$$

[$\Delta a(x,y)$ for the SSPM distance]

$$\delta 1=\begin{cases} d[i+N_A(i),j+N_B(j)] + D[i,j]: \\ \qquad \text{if } i\in Ch_A(x) \text{ and } j\in Ch_B(y); \\ \text{infinite} \qquad : \\ \qquad \text{otherwise.} \end{cases} \qquad (4\text{-}4\text{-}1)$$

However, except the WSPM similarity, we can improve the algorithms. We will explain how to improve the algorithm for computing the ISPM distance.

[Computation of $\Delta a(x,y)$: the ISPM distance]

The children of x are named x_1, x_2, ..., x_m from left to right.

$$d[i,j]=\min\begin{cases} d[i+1,j+N_B(j)] + D[x_i,j], \\ d[\ i\ ,j+1] + q \\ d[i+1,\ j\ ] + N_A(x_i)*r. \end{cases} \qquad (4\text{-}5)$$

The boundary conditions are

$d[m+1,j] = \{el_B(y)+1-j\}*q$

$(y<j\leq el_B(y)+1)$,

$d[i,el_B(y)+1] = \{N_A(x_i)+..+N_A(x_m)\}*r$

$(1\leq i\leq m)$.

Then,

$\Delta a(x,y)=d[1,y+1]+p(x,y)$.

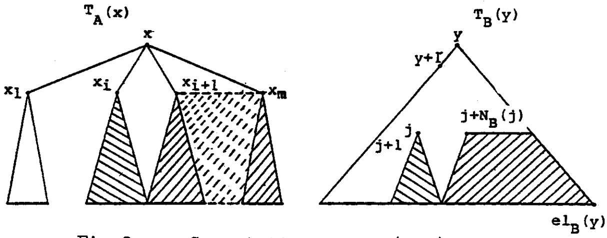


Fig.9 Computation of $a(x,y)$ for the ISPM distance.

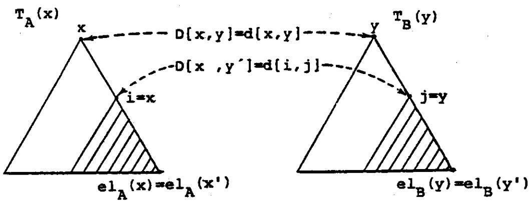


Fig.10 Computation of $D[x,y]$ and $D[x',y']$.

5. Improved algorithms

We can further improve the algorithms for computing the Tai, ISPM and DSPM distances. We take the Tai distance as an example.

Note that $\Delta b(x,y)$ and $\Delta c(x,y)$ can be computed by the formula (3-4). That is, although this formula is applied to $d[i,j]$ such that $x<i\leq el_A(x)$ and $y<j\leq el_B(y)$, we can compute $d[x,j]$ and $d[i,y]$ using it. Then, $D[x,y]$ is as follows:

$$D[x,y]=d[x,y]=\min\begin{cases} d[x+1,y+1]+p(x,y) \\ d[\ x\ ,y+1]+q \\ d[x+1,\ y\ ]+r \end{cases} \quad (5\text{-}1)$$

Consider the case shown in Fig.10. The end leaf of $T_A(x)$ ($T_B(y)$) is also that of $T_A(x')$ ($T_B(y')$). In the algorithm shown in section 3, we compute $d[i,j]$ for getting $D[x,y]$ separately from $d[i,j]$ for $D[x',y']$. However, since the computation of $d[i,j]$ proceeds leftward from the end leaves of two trees, we can get $D[x',y']$ in the midst of computing $d[i,j]$ for $D[x,y]$.

[Computation of the Tai distance]

(1) Main algorithm:

Let L be the number of leaf in T, leaf(k) be k-th leaf from the leftmost leaf, and R(u) be the youngest node v such that u is the end leaf of T(v).

for h:=1 to L_A do
for k:=1 to L_B do
begin
  x:=$R_A(leaf_A(h))$;
  y:=$R_B(leaf_B(k))$;
  "compute $d[x+N_A(x),\ y+N_B(y)]\sim d[x,y]$"; (5-2)
 (*where, $z+N(z)=el(z)+1$ (z=x or z=y),*)
end:
$D(T_A,T_B):=D[1,1]$.

(2) Computation of $d[i,j]$:
if $el_A(x)=el_A(i)$ and $el_B(y)=el_B(j)$, we get $D[i,j]$ by formula (5-1). Otherwise, we compute $d[i,j]$ by formula (3-4).

$$\delta 1=\begin{cases} d[i+1,j+1]+p(i,j) \\ \quad : \text{if } el_A(x)=el_A(i) \text{ and } \\ \qquad el_B(y)=el_B(j); \\ d[i+N_A(i),j+N_B(j)] \\ \quad : \text{otherwise;} \end{cases}$$

$\delta 2= d[\ i\ ,j+1] + q$:
$\delta 3= d[i+1,\ j\ ] + r$;
$d[i,j] = \min\{\delta 1, \delta 2, \delta 3\}$;
if $el_A(x)=el_A(i)$ and $el_B(y)=el_B(j)$
  then $D[i,j] = d[i,j]$;
where, the boundary conditions are
$d[el_A(x)+1,j]=\{el_B(y)+1-j\}*q$
  $(y\leq j\leq el_B(y)+1)$,
$d[i,el_B(y)+1]=\{el_A(x)+1-i\}*r$
  $(x\leq i\leq el_A(x)+1)$.

The time and space complexities of the above algorithm are $O(L_AL_BN_AN_B)$ and $O(N_AN_B)$, respectively.

6. Relation of metrics already proposed.

We clarify the relation between the metrics proposed in this paper and metrics in the previous works. The WSPM similarity, as a special case, becomes the metric between binary trees by Nakabayashi and Kamata [12]. Although the definition of ISPM is different from that of SPM by Tanaka [10], they are equivalent. In this paper SPM is called ISPM to express the meaning of SPM clearly. If the inverse mapping of a given mapping is SPM, this is called DSPM. Selkow's metric is the restricted SSPM metric. We can compute Selkow's metric by letting $D[x,y]:=\Delta a(x,y)$ in for formula (3-1-3), where $\Delta a(x,y)$ is that of the SSPM metric. Therefore, it can be defined by the following mapping.