

木の断片距離：木の編集距離の新たな階層

菅 智 宏^{†1} 樋 口 鐘 一^{†1}
山 本 恭 之^{†1} 平 田 耕 一^{†2}

本論文では、木の編集距離の変種として、隣接ノードの出現に基づく木の断片マッピングを導入し、その最小コストとして木の断片距離を定式化する。また、それらにトップダウン性とボトムアップ性という制限を導入することで、断片距離のトップダウン断片距離とボトムアップ断片距離を定式化する。そして、それらが距離性を満たすことを示すと共に、トップダウン断片距離はトップダウン距離と一致するがボトムアップ断片距離と断片距離がこれまでに知られている木の編集距離の階層とは別の階層を成すことを示す。

Segmental Distance: An Alternative Hierarchy of Tree Edit Distance

TOMOHIRO KAN,^{†1} SHOICHI HIGUCHI,^{†1}
YOSHIYUKI YAMAMOTO^{†1} and KOUICHI HIRATA^{†2}

In this paper, as a variation of the tree edit distance, we introduce a segmental mapping based on the occurrences of the contiguous nodes and formulate a segmental distance as the minimum cost of segmental mappings. Also we introduce a top-down segmental distance and a bottom-up segmental distance by adding a top-down condition and a bottom-up condition to segmental mappings. Then, we show that all of them are metric, and, while the top-down segmental distance coincides with a top-down distance, the bottom-up segmental distance and the segmental distance provide an alternative hierarchy of the tree edit distance to a known one.

^{†1} 九州工業大学 情報工学科

Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology

^{†2} 九州工業大学情報工学研究院

Department of Artificial Intelligence, Kyusyu Institute of Technology

1. はじめに

XML や HTML などの半構造データやバイオインフォマティクスの DNA データや糖鎖データからのデータマイニングや機械学習では、データをラベル付き根付き順序木（以後、単に木という）として扱い、2つの木の非類似度（距離）を計算する必要がある。現在、木の非類似度として一般的に用いられているものが、木の編集距離^{1),8)}である。

木の編集距離は、ノードの挿入、削除、置換という3つの編集操作を用いて、一方の木を他方の木に変換するために必要な最小コストとして定義され、最小コストの Tai マッピング⁸⁾（以後、単にマッピングという）と対応することが知られている。ここで、マッピングとは、2つの木に対する対応付けられたノードの組の集合である。ただし、このノードの組は、(1) 1対1の対応、(2) 先祖子孫関係の保存、(3) 兄弟関係の保存、という条件を満たしていなければならない。

木の編集距離の計算時間は、比較する木の頂点数 n に対して $O(n^3)$ 時間である³⁾。そこで、効率的に木の編集距離を近似し、また、適用する対象に応じた、制限付きマッピング (constrained mapping)、調和的マッピング (accordant mapping)、次数2マッピング (degree-2 mapping)、トップダウンマッピング (top-down mapping)、ボトムアップマッピング (bottom-up mapping) などのマッピングの変種が提案されており、それぞれのマッピングの最小コストとして、編集距離の変種である制限付き距離 (constrained distance)^{10),13)}、調和的距離 (accordant distance)^{4),5)}、次数2距離 (degree-2 distance)^{6),10),14)}、トップダウン距離 (top-down distance)^{2),7),12)}、ボトムアップ距離 (bottom-up distance)^{5),9),11)} が提案されている。これらの距離の計算時間はすべて $O(n^2)$ 時間である^{*1}。

ここで、制限付きマッピングとは、マッピングに含まれている2つのノードの最近共通先祖を根とする部分木の子孫のみにマッピングを許すようなマッピングであり、調和的マッピングとは、3つのノードのうち、もし2組のノードの最近共通先祖が同じであれば、その最近共通先祖の対応先も同様に相手の木の2組のノードの最近共通先祖でなければいけないというマッピングである。また、次数2マッピングとは、2つノードの最近共通先祖同士が対応していなければならないというマッピングであり、トップダウンマッピングとは、あるノードの親同士が対応していなければならないというマッピングである。最後に、ボトムアップ

*1 ボトムアップ距離の計算時間は、インデルコストの場合には $O(n)$ 時間である⁹⁾ が、任意コストの場合には $O(n^2)$ 時間となる¹¹⁾。

マッピングとは、あるノードが対応すると、その子孫すべてが対応していなければならないというマッピングである。

これらのマッピングの変種は、図 1(左) のような階層構造を成すことが知られている。ここで、図 1 の TAI, CST, ACC, DG2, TOP, BOT, ISO はそれぞれ、マッピング、制限付きマッピング、調和的マッピング、次数 2 マッピング、トップダウンマッピング、ボトムアップマッピング、木の同型写像のクラスである。図 1 の階層は、マッピングのクラスの包含関係、すなわち、上位のマッピングのクラス A と下位のマッピングのクラス B に対して、 $M \in A$ となる任意のマッピング M は必ず $M \in B$ となることを表している^{*1}。

本論文では、これまでのマッピングの変種とは異なり、新たに隣接ノードの出現に基づく断片マッピング (segmental mapping) を導入する。断片マッピングは、マッピングに含まれているノードの組がその先祖の組を含むときは、必ず親の組も含む、というものである。つまり、マッピングに含まれるノードの組は先祖子孫関係があれば必ず「繋がっている」ことを要請している。さらに、必ず根を含む断片マッピングであるトップダウン断片マッピング (top-down segmental mapping), および、必ず葉を含む断片マッピングであるボトムアップ断片マッピング (bottom-up segmental mapping) を導入する。そしてこれらのマッピングの最小コストとして、断片距離 (segmental distance), トップダウン断片距離 (top-down segmental distance), ボトムアップ断片距離 (bottom-up segmental distance) を定義する。

本論文では、まず、これらの距離が距離性を満たすことを示す。そして、トップダウン断片距離はトップダウン距離と一致するが、ボトムアップ断片距離はボトムアップ距離よりも真に一般的となり、断片距離はトップダウン断片距離とボトムアップ断片距離よりも真に一般的となることを示す。その結果、断片距離、トップダウン断片距離、ボトムアップ断片距離は、図 1(左) の編集距離の階層構造に加えて図 1(右) のような新たな階層構造を成す。ここで、SG, TOPSG, BOTSG はそれぞれ、断片マッピング、トップダウン断片マッピング、ボトムアップ断片マッピングのクラスである。

2. 木の編集距離とその変種

閉路を持たない連結無向グラフを木という。本論文では、木 $T = (V, E)$ に対して、 $v \in V(T)$ を $v \in T$, $|V|$ を $|T|$ で表す。1 つのノード r を根として選んだ木を根付き木といい、木 T の根を $r(T)$ と表す。根から v への経路上のノードを、 v の先祖といい、 v の先祖の集合を

^{*1} 後述するコスト γ が距離となるときは、 $\text{Acc} = \text{DG2}$ となる^{4),5)}。

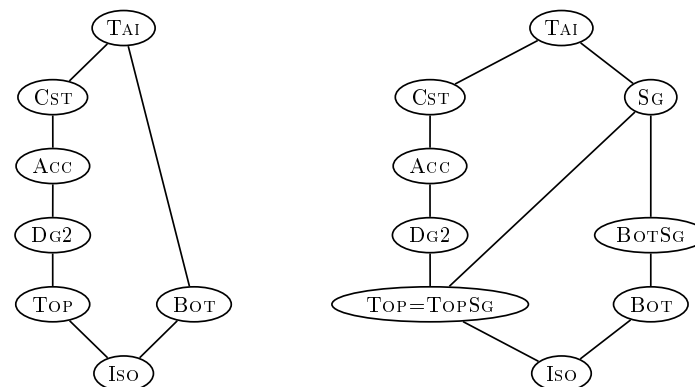


図 1 マッピングの階層構造^{4),10)}(左) と断片マッピングを含む新たなマッピングの階層構造 (右)。

$\text{ans}(v)$ と表す。 v が u の先祖であるとき、 $v < w$ と表す。特に、 v に隣接する先祖を v の親といい $\text{par}(v)$ と表す。さらに、ノード v と w のそれぞれの共通の先祖のうち、最も v, w に近い先祖を最近共通先祖といい、 $v \sqcap w$ と表す。また $v, u \in T$ において、 v が u の先祖であったとき u は v の子孫といい、 v に隣接する子孫を v の子という。子を持たないノードを葉といい、 T に含まれるすべての葉を $\text{lv}(T)$ で表す。

木 T における頂点 $v \in T$ に対して、 v の先行順 (preorder) における順番を $\text{pre}(v)$, 後行順 (postorder) における順番を $\text{post}(v)$ と表す。このとき、 $v, w \in T$ に対して、 $\text{pre}(v) < \text{pre}(w)$ かつ $\text{post}(v) < \text{post}(w)$ のとき、 v は w の左にあるといい、そのとき w は v の右にあるという。また、このことを $v \prec w$ と表す。

各頂点がアルファベットの有限集合 Σ に含まれる文字によりラベル付けされている根付き木をラベル付き根付き木といい、ノード v のラベルを $l(v)$ と表記する。また、各頂点の子に左右の順序を指定しているラベル付き根付き木を、ラベル付き根付き順序木といい、本論文では単に木と呼ぶことにする。

木 T_1 から木 T_2 への編集距離⁸⁾ は、以下の編集操作を用いることで定式化することができる。

定義 1 (編集操作) 以下の三つの操作を木 T_1 から木 T_2 への編集操作 (edit operation) という。

1. 置換 (substitution) : T_1 のノードラベル v を T_2 のノードラベル w に置き換える操作。

$v \mapsto w$ と表記する.

2. 挿入 (insertion): ノード w を T_2 に挿入する操作. $\varepsilon \mapsto w$ と表記する.

3. 削除 (deletion): T_1 のノード v を削除する操作. $v \mapsto \varepsilon$ と表記する.

図 2 は編集操作の直観的な説明図である.

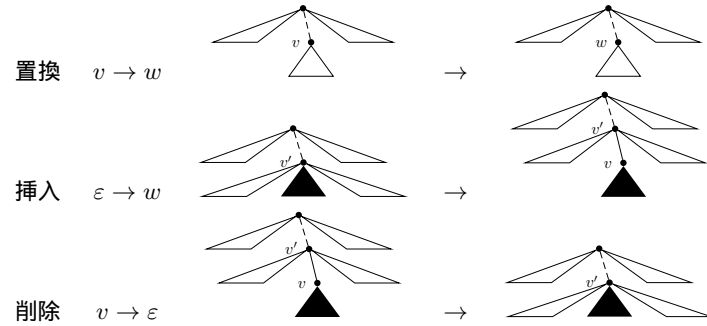


図 2 木の編集操作

空白記号 $\varepsilon \notin \Sigma$ に対して, $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ とする. このとき, ラベルの組に対するコスト (cost) を $\gamma: (\Sigma_\varepsilon \times \Sigma_\varepsilon - \{(\varepsilon, \varepsilon)\}) \mapsto \mathbf{R}$ とする. 頂点 v とそのラベル $l(v)$ を同一視することにより, 編集操作 $v \mapsto w$ を $l(v) \mapsto l(w)$ と表す. そして, 編集操作 $l_1 \mapsto l_2$ ($l_1, l_2 \in \Sigma$) のコストを $\gamma(l_1 \mapsto l_2) = \gamma(l_1, l_2)$ と定義する. 特に断らない限り, コスト γ は距離 (metric) となる, すなわち, 以下が成り立つと仮定する.

- (1) $\gamma(l_1 \mapsto l_2) \geq 0$.
- (2) $\gamma(l_1 \mapsto l_2) = 0 \Leftrightarrow l_1 = l_2$.
- (3) $\gamma(l_1 \mapsto l_2) = \gamma(l_2 \mapsto l_1)$.
- (4) $\gamma(l_1 \mapsto l_3) \leq \gamma(l_1 \mapsto l_2) + \gamma(l_2 \mapsto l_3)$.

定義 2 (編集距離) T_1 から T_2 への編集操作の列 $E = e_1, \dots, e_k$ のコストを $\gamma(E) = \sum_{i=1}^k \gamma(e_i)$ とするとき, T_1 と T_2 の編集距離 (edit distance) $\tau(T_1, T_2)$ を以下のよう定義する.

$$\tau(T_1, T_2) = \min\{\gamma(E) \mid E \text{ は } S \text{ から } T \text{ への編集操作の列}\}.$$

定義より, $\tau(T_1, T_2) = 0$ となるとき, そのときに限り, T_1 と T_2 は同型 (isomorphism) と

なる. T_1 と T_2 が同型のとき, $T_1 \equiv T_2$ と表す.

Tai⁽⁸⁾ は, 編集距離を特徴付けるために, 以下の Tai マッピングを導入した.

定義 3 (Tai マッピング) $M \subseteq V(T_1) \times V(T_2)$ は, 任意の $(v_1, w_1), (v_2, w_2) \in M$ が以下の条件を満たすとき, 木 T_1 から木 T_2 への Tai マッピング (Tai-mapping) という.

- (1) $v_1 = v_2 \Leftrightarrow w_1 = w_2$.
- (2) $v_1 < v_2 \Leftrightarrow w_1 < w_2$.
- (3) $v_1 \prec v_2 \Leftrightarrow w_1 \prec w_2$.

本論文では, Tai マッピングを単にマッピングという. 編集距離と同様に, マッピングにも以下のようにコストを割り振ることができる. ここで, I および J を, マッピング M の要素ではない T_1 の頂点の集合および T_2 の頂点の集合とする.

$$\gamma(M) = \sum_{(v,w) \in M} \gamma(v \mapsto w) + \sum_{v \in I} \gamma(v \mapsto \varepsilon) + \sum_{w \in J} \gamma(\varepsilon \mapsto w).$$

定理 1 (Tai⁽⁸⁾) 木 T_1, T_2 に対して, 以下が成り立つ.

$$\tau(T_1, T_2) = \min\{\gamma(M) \mid M \text{ は } T_1 \text{ と } T_2 \text{ の Tai マッピング}\}.$$

図 2 はマッピングの直観的な説明図である. 破線で結ばれている T_1, T_2 のノード間の対応がマッピングを表し, ラベルが異なる場合は置換と対応する. また, マッピングに含まれていない T_1 のノードは削除を表し, T_2 のノードは挿入を表す.

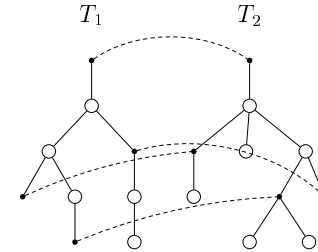


図 3 木 T_1, T_2 間のマッピング

$T_1 \equiv T_2$ のとき, $\gamma(M) = 0$ となるマッピング M は T_1 と T_2 の同型写像となる. 同型写像 M に対して, $M \in \text{Iso}$ と表す.

最後に, マッピングを制限をすることにより得られる, マッピングの変種と, それらの最小コストとして定式化される編集距離の変種を紹介する.

定義 4 (編集距離の変種) $M \subseteq V(T_1) \times V(T_2)$ を木 T_1 と T_2 間のマッピングとする。また、 $M - \{(r(T_1), r(T_2))\}$ を M^- と表す。

- (1) 以下の条件を満たす M を制限マッピング (constrained mapping)^{(10),(13)} といい、 $M \in \text{CST}$ と表す。

$$\forall (v_1, w_1), (v_2, w_2), (v_3, w_3) \in M \left(v_3 < v_1 \sqcup v_2 \iff w_3 < w_1 \sqcup w_2 \right).$$

- (2) 以下の条件を満たす M を調和的マッピング (accordant mapping)^{(4),(5)} といい、 $M \in \text{ACC}$ と表す。

$$\forall (v_1, w_1), (v_2, w_2), (v_3, w_3) \in M \left(v_1 \sqcup v_2 = v_1 \sqcup v_3 \iff w_1 \sqcup w_2 = w_1 \sqcup w_3 \right).$$

- (3) 以下の条件を満たす M を次数 2 マッピング (degree-2 mapping)^{(6),(10),(14)} といい、 $M \in \text{DG2}$ と表す。

$$\forall (v_1, w_1), (v_2, w_2) \in M^- \left((v_1 \sqcup v_2, w_1 \sqcup w_2) \in M \right).$$

- (4) 以下の条件を満たす M をトップダウンマッピング (top-down mapping)^{(2),(7),(12)} といい、 $M \in \text{TOP}$ と表す。

$$\forall (v, w) \in M^- \left((\text{par}(v), \text{par}(w)) \in M \right).$$

- (5) 以下の条件を満たす M をボトムアップマッピング (bottom-up mapping)^{(5),(9),(11)} といい、 $M \in \text{BOT}$ と表す^{*1}。

$$\forall (v, w) \in M \left(\begin{array}{l} \forall v' \in T_1(v) \exists w' \in T_2(w) \left((v', w') \in M \right) \\ \wedge \forall w' \in T_2(w) \exists v' \in T_1(v) \left((v', w') \in M \right) \end{array} \right).$$

また、定理 1 と同様に、制限マッピング、調和的マッピング、次数 2 マッピング、トップダウンマッピング、ボトムアップマッピングの最小コストを、それぞれ、制限距離 (constrained distance)、調和的距離 (accordant distance)、次数 2 距離 (degree-2 distance)、トップダウン距離 (top-down distance)、ボトムアップ距離 (bottom-up distance) という。

例 1 図 4 のマッピング $M_i (i = 1, 2, 3, 4, 5, 6)$ を考える。ここで、点線はノード同士のマッピングを表している。

このとき、 $M_1 \in \text{TOP}$ 、 $M_2 \notin \text{TOP}$ であるが $M_2 \in \text{DG2}$ である。また、 $M_3 \notin \text{DG2}$ であるが $M_3 \in \text{ACC}$ である。 $M_4 \notin \text{ACC}$ であるが $M_4 \in \text{CST}$ である。 $M_5 \notin \text{CST}$ であるが $M_5 \in \text{TAI}$ である。最後に、 $M_6 \in \text{BOT}$ である。しかし、 T_1 において $c \leq a \sqcup b$ となるが、 T_2 において

$c \not\leq a \sqcup b$ となるので、 $M_6 \notin \text{CST}$ である。また、 $M_i \notin \text{BOT} (i = 1, 3, 4, 5)$ である。

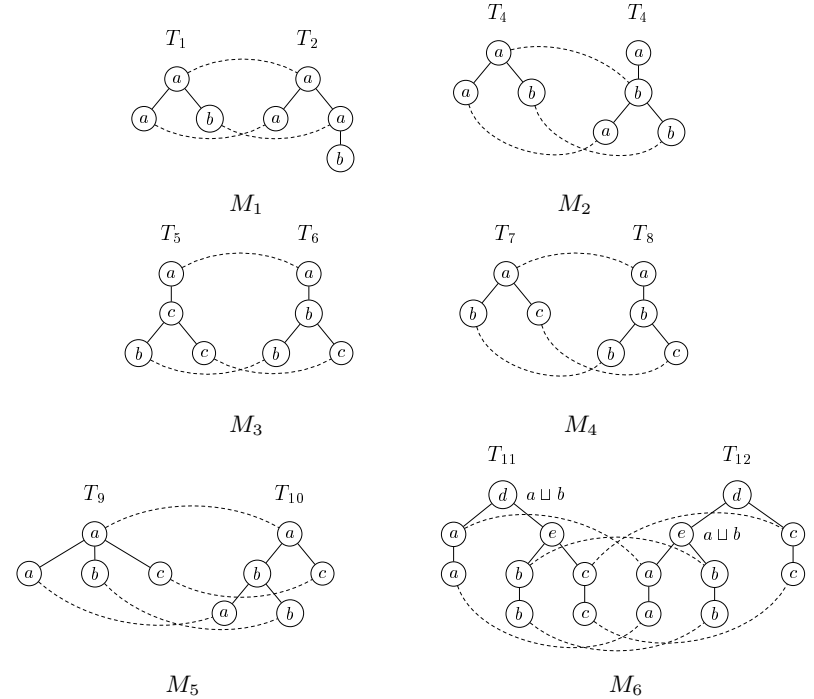


図 4 例 1 のマッピング $M_i (i = 1, 2, 3, 4, 5, 6)$ 。

3. 木の断片距離

本節では、マッピングに含まれているノードの組がその先祖の組を含むときには必ずその親の組も含む、という断片マッピングを新たに導入する。そして、木の断片距離を断片マッピングの最小コストとして定式化する。

定義 5 (断片マッピング) $M \subseteq V(T_1) \times V(T_2)$ を木 T_1 と T_2 間のマッピングとする。

- (1) 以下の条件を満たす M を断片マッピング (segmental mapping) といい、 $M \in \text{SG}$ と表す。

*1 Valiente⁽⁹⁾ のボトムアップマッピングは制限付きマッピングであるという条件が付加されているが、その条件は不要なので、本論文でもこの条件を除いている^{(5),(11)}。

$$\begin{aligned} \forall (v, w) \in M & \left(\left((v', w') \in M \right) \wedge \left(v' \in \text{ans}(v) \right) \wedge \left(w' \in \text{ans}(w) \right) \right) \\ & \implies \left((\text{par}(v), \text{par}(w)) \in M \right). \end{aligned}$$

- (2) $(r(T_1), r(T_2)) \in M$ を満たす断片マッピングをトップダウン断片マッピング (top-down segmental mapping) といい、 $M \in \text{TopSG}$ と表す。
- (3) 以下の条件を満たす断片マッピング M をボトムアップ断片マッピング (bottom-up segmental mapping) といい、 $M \in \text{BotSG}$ と表す。

$$\begin{aligned} \forall (v, w) \in M \exists (v', w') \in M \\ \left(\left(v \in \text{ans}(v') \right) \wedge \left(w \in \text{ans}(w') \right) \wedge \left(v \in \text{lv}(T_1) \right) \wedge \left(w \in \text{lv}(T_2) \right) \right) \end{aligned}$$

例 2 図 5 のマッピング $M_i (i = 1, 2, 3)$ を考える。ここで、点線はノード同士のマッピングを表している。このとき、 $M_1 \in \text{Top}$ 、 $M_1 \in \text{BotSG}$ であるが、 $M_1 \notin \text{Bot}$ である。また、 $M_2 \in \text{BotSG}$ であるが、 $M_2 \notin \text{Top}$ である。 $M_3 \in \text{SG}$ であるが $M_3 \notin \text{BotSG}$ かつ $M_3 \notin \text{Top}$ である。また、 $M_3 \notin \text{CST}$ である。一方、例 1 の M_3 に対しては、 $M_3 \in \text{SG}$ 。さらに、例 1 の M_6 に対しては、 $M_6 \in \text{BotSG}$ かつ $M_6 \in \text{SG}$ である。

木 T_i と T_{i+1} のマッピングを M_i とする ($i = 1, 2$)。このとき、マッピングの合成 (composition) $M_1 \circ M_2$ を以下のように定義する。

$$M_1 \circ M_2 = \{(u, w) \mid (u, v) \in M_1 \text{ かつ } (v, w) \in M_2 \text{ となる } v \in T_2 \text{ が存在する}\}.$$

補題 1 木 T_i と T_{i+1} の断片マッピングを M_i とする ($i = 1, 2$)。

- (1) $M_1 \circ M_2$ は木 T_1 と T_3 の断片マッピングである。
- (2) コスト γ が距離のとき、 $\gamma(M_1 \circ M_2) \leq \gamma(M_1) + \gamma(M_2)$ 。
- (証明) (1) $(u, v) \in M_1$ かつ $(v, w) \in M_2$ となるノード $v \in T_2$ に対して、以下の二つの条件を考える。
- (a) $u_1 \in \text{ans}(u)$ かつ $v_1 \in \text{ans}(v)$ となる $(u_1, v_1) \in M$ が存在する。
- (b) $v_2 \in \text{ans}(v)$ かつ $w_2 \in \text{ans}(w)$ となる $(v_2, w_2) \in M$ が存在する。
- M_1 は断片マッピングなので、(a) が真となるときには $(\text{par}(u), \text{par}(v)) \in M_1$ となる。また、 M_2 は断片マッピングなので、(b) が真となるときには $(\text{par}(v), \text{par}(w)) \in M_2$ となる。したがって、(a) と (b) が共に真となると、 $(\text{par}(u), \text{par}(v)) \in M_1$ かつ $(\text{par}(v), \text{par}(w)) \in M_2$ となる。よって、 $M_1 \circ M_2$ の定義より、 $(\text{par}(u), \text{par}(w)) \in M_1 \circ M_2$ となる。
- (2) $u \in T_1$ と $w \in T_3$ に対して、 $(u, w) \in M_1 \circ M_2$ であるとき、 $(u, *)$ となるマッピングが M_1 に存在しないとき、 $(*, w)$ となるマッピングが M_2 に存在しないとき、の三つの場合

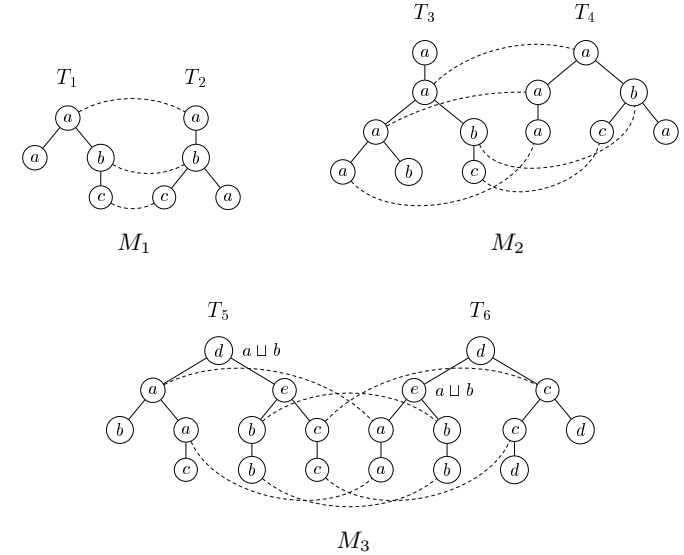


図 5 例 2 のマッピング $M_i (i = 1, 2, 3)$ 。

がある。それぞれの場合は、 $u \mapsto w$ 、 $u \mapsto \epsilon$ 、 $\epsilon \mapsto w$ と対応している。コスト γ は距離なので、 γ は三角不等式 $\gamma(u \mapsto w) \leq \gamma(u \mapsto \epsilon) + \gamma(\epsilon \mapsto w)$ を満たす。したがって、(2) が成り立つ。□

定理 2 以下が成り立つ。

- (1) $\text{Top} = \text{TopSG} \subset \text{SG} \subset \text{TAI}$ 。
- (2) $\text{Bot} \subset \text{BotSG} \subset \text{SG} \subset \text{TAI}$ 。
- (3) $A \in \{\text{DG2}, \text{Acc}, \text{CST}\}$ に対して $A \# \text{BotSG}$ かつ $A \# \text{SG}$ 。

(証明) 定義 5 より、(1) と (2) が成り立つ。また、例 2 より、(3) が成り立つ。□

定理 2 により、SG、TopSG、BotSG は、CST、Acc、DG2、Top、Bot に対して、1 章の図 1(右) のような新しい階層構造を成すことが分かる。

定義 6 (木の断片距離) 木 T_1 と T_2 の断片距離 (segmental distance) $\tau_\Lambda(T_1, T_2)$ を以下のように定義する。

$$\tau_\Lambda(T_1, T_2) = \min\{\gamma(M) \mid M \text{ は } T_1 \text{ と } T_2 \text{ の断片マッピング}\}.$$

同様に、トップダウン断片マッピングおよびボトムアップ断片マッピングの最小コストを、

それぞれ、トップダウン断片距離 (top-down segmental distance), ボトムアップ断片距離 (bottom-up segmental distance) と定義する。

定理 3 木の断片距離 $\tau_{\wedge}$ は距離の公理を満たす。すなわち、 $\tau_{\wedge}$ は木 T_1, T_2, T_3 に対し、 $\tau_{\wedge}(T_1, T_2) \geq 0$, $\tau_{\wedge}(T_1, T_2) = 0 \Leftrightarrow T_1 \equiv T_2$, $\tau_{\wedge}(T_1, T_2) = \tau_{\wedge}(T_2, T_1)$, $\tau_{\wedge}(T_1, T_3) \leq \tau_{\wedge}(T_1, T_2) + \tau_{\wedge}(T_2, T_3)$ を満たす。

(証明) 最初の 3 つの式は、木の断片距離の定義より成り立つ。 M_i ($i = 1, 2$) を T_i と T_{i+1} の断片マッピングとする。このとき、補題 1 より、 $\tau_{\wedge}(T_1, T_3) \leq \gamma(M_1 \circ M_2) \leq \tau_{\wedge}(T_1, T_2) + \tau_{\wedge}(T_2, T_3)$ となり、最後の式も成り立つ。 $\square$

ボトムアップ断片マッピングにおいても同様に、補題 1 と定理 3 が得られる。

4. おわりに

本稿では、新たな階層構造として、断片マッピング、ボトムアップ断片マッピング、トップダウン断片マッピングを導入し、その最小コストとして断片距離、ボトムアップ断片距離、トップダウン断片距離を定義した。そして、これらの距離が距離の公理を満たすことを示した。

今後の課題は、まず、断片距離とボトムアップ断片距離を計算するアルゴリズムの設計が挙げられる。また、それらのアルゴリズムを実装し、他の距離との関係や断片距離の有用性を検証することも今後の課題である。

無順序木のボトムアップ距離の計算は MAX SNP 困難である¹¹⁾ことから^{*1}、無順序木の断片距離とボトムアップ断片距離の計算も MAX SNP 困難となる。そこで、A* アルゴリズムのような、無順序木の断片距離、ボトムアップ断片距離、ボトムアップ距離のヒューリスティックアルゴリズムの開発も今後の課題である。

参考文献

- 1) P.Bille: *A survey on tree edit distance and related problems*, Theoret. Comput. Sci. **337**, 217–239, 2005.
- 2) S.S.Chawathe: *Comparing hierarchical data in external memory*, Proc. VLDB'99, 90–101, 1999.
- 3) E.D.Demaine, S.Mozes, B.Rossmann, O.Weimann: *An optimal decomposition algorithm for tree edit distance*, ACM Trans. Algorithms **6**, 2009.
- 4) T.Kuboyama: *Matching and learning in trees*, Ph.D thesis, University of Tokyo,

2007.

- 5) T.Kuboyama, K.Shin, H.Kashima: *Flexible tree kernels based on counting the number of tree mappings*, Proc. MLG'06, 61–72, 2006.
- 6) S.-Y.Lu: *A tree-to-tree distance and its application to cluster analysis*, IEEE Trans. Pattern Anal. Mach. Intell. **1**, 219–224, 1979.
- 7) S.M.Selkow: *The tree-to-tree editing problem*, Inform. Process. Lett. **6**, 184–186, 1977.
- 8) K.-C. Tai: *The tree-to-tree correction problem*, J. ACM **26**, 422–433, 1979.
- 9) G.Valiente: *An efficient bottom-up distance between trees*, Proc. SPIRE '01, 212–219, 2001.
- 10) J.T.L.Wang, K.Zhang: *Finding similar consensus between trees: An algorithm and a distance hierarchy*, Pattern Recog. **34**, 127–137, 2001.
- 11) Y.Yamamoto, K.Hirata, T.Kuboyama: *A bottom-up edit distance between rooted labeled trees*, Proc. LLLL'11, 26–33, 2011.
- 12) W.Yang: *Identifying syntactic differences between two programs*, Software Pract. Exp. **21**, 739–755, 1991.
- 13) K.Zhang: *Algorithms for the constrained editing distance between ordered labeled trees and related problems*, Pattern Recog. **28**, 463–474, 1995.
- 14) K.Zhang, J.T.L.Wang, D.Shasha: *On the editing distance between undirected acyclic graph*, Int. J. Found. Comput. Sci. **7**, 43–58, 1995.

*1 順序木の場合と同じく、無順序木のボトムアップ距離の計算はインデルコストの場合は $O(n)$ 時間となるが、任意コストの場合は MAX SNP 困難となる¹¹⁾。