人工知能学会研究会資料
SIG-FPAI-B501-01

根付きラベル付き木の Tai マッピングと共通部分森の関係

Tai Mapping Hierarchy for Rooted Labeled Trees Relevant to Common Subforest

芳野 拓也 [1*]　平田 耕一 [2]
Takuya Yoshino[1]　Kouichi Hirata[1,2]

[1] 九州工業大学大学院情報工学府
[1] Graduate School of Computer Science and Systems Engineering,
[2] 九州工業大学情報工学研究院
[2] Department of Artificial Intelligence,
Kyushu Institute of Technology

Abstract: A *Tai mapping* between two rooted labeled trees (trees, for short) is a one-to-one node correspondence preserving ancestors and siblings (if trees are ordered). The variations of the Tai mapping are known to provide a hierarchy, called a *Tai mapping hierarchy*. In this paper, we characterize the Tai mapping hierarchy relevant to a *common subforest* by focusing on the connections of nodes and the arrangements of subtrees in a common subforest. Then, we fill a gap in the Tai mapping hierarchy by introducing several new variations. Furthermore, we summarize and investigate the time complexity of computing the variations of the edit distance as the minimum cost of the variations of the Tai mapping.

1 はじめに

ウェブマイニングにおける HTML, XML データや, バイオインフォマティクスにおける DNA, 糖鎖データのような木構造を計算することは, データマイニングにおける重要なタスクの一つである．木構造刊の距離を測定するもっとも有名なものとして**編集距離** (*edit distance*) [11] が挙げられる．編集距離は, ノードの**削除**, **挿入**, **置換**からなる**編集操作**を用いて, 一方の木から他方の木への変換を行う際の最小コストとして定式化される．編集距離は, 先祖関係を (順序木においては兄弟関係も) 保存するノード間の一対一対応である **Tai マッピング** (*Tai mapping*) と密接な関係があり, 可能なマッピングの最小コストと編集距離は一致する [11].

編集距離は木の比較として一般的であるが, 応用分野によっては一般的過ぎるため, それらの応用分野に対して様々な編集距離の変種が考えられてきた．それらの距離は Tai マッピングの変種である**トップダウンマッピング** (Top) [2, 10], **LCA 保存マッピング** (**次数 2 マッピング**) (Lca) [19], **調和的マッピング** (**Lu マッピング**) (Acc) [7, 9], **孤立部分木マッピング** (**制限マッピング**) (Ilst) [14, 16, 17], **劣制限マッピング** (Less) [7, 8], **アライメント可能マッピング** (Aln) [7], **ボトムアップマッピング** (Bot) [7, 12, 15], **断片マッピング** (Sg) [6], **トップダウン断片マッピング** (TopSg) [6] の最小コストとして定式化されている．ここで, “Less=Aln” [7], “Top=TopSg” [6] となる．図 1 の灰色の線で囲まれたマッピングの階層が知られている．

本稿では, マッピングに含まれるノードの組を**共通部分森**と関連付け, 図 1 の黒色の線で囲まれたマッピングを新たに導入することで, Tai マッピングの階層を特徴づける．そして, 各マッピングの最小コストとして定式化される距離の計算時間について議論する．

*連絡先：九州工業大学大学院情報工学府
〒 820-8502 福岡県飯塚市河津 680-4
E-mail:yoshino@dumbo.ai.kyutech.ac.jp

2 準備

閉路のない連結グラフを**木** (*tree*) という．木 $T = (V, E)$ における V と E をそれぞれ $V(T)$ と $E(T)$ で表す．また, $|V|$ を T の**大きさ** (*size*) といい, $|T|$ で表す．簡単のため, $v \in V(T)$ を単に $v \in T$ で表し, 空の木を $\emptyset$ で表す．

ある特定のノード r を根として持つ木を**根付き木** (*rooted tree*) といい, その r のことを**根** (*root*) という．根付き木 T の根を $r(T)$ で表す．r を根とする根付き木の各ノード v に対して, v から r へのパスを $UP_r(v)$ で表す．$v(\neq r)$ に隣接する $UP_r(v)$ 上のノードを v の**親**

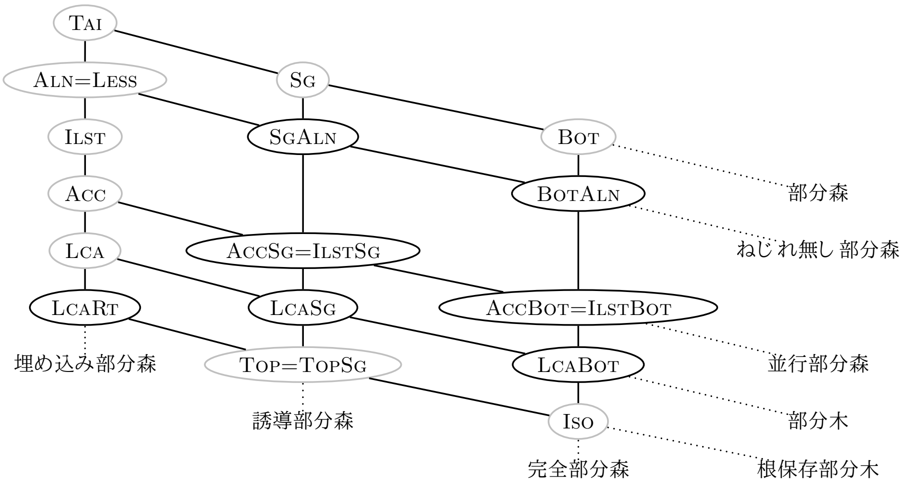

図 1: Tai マッピングの階層. 灰色の線で囲まれたマッピングが既知の階層 [6, 7, 14] をなし, 黒色の線で囲まれたマッピングが本稿で新たに導入するマッピングである.

(*parent*) といい, $par(v)$ で表す. また, $UP_r(v) \setminus \{v\}$ 上のノードを v の**先祖** (*ancestors*) といい, $anc(v)$ で表す. ノード u と v に対して, $UP_r(u) \cap UP_r(v)$ の要素を u と v の共通先祖という. u(あるいは v) へのパスが最も短い共通先祖を u と v の**最近共通先祖** (*least common ancestor*) といい, $u \sqcup v$ で表す. v が u の親であるとき, u を v の**子** (*child*) といい, v の子供の列を $ch(v)$ で表す.

T を根付き木とし, $v \in T$ とする. このとき, v の子の数を**次数** (*degree*) といい $d(v)$ で表す. $d(v) = |ch(v)|$ である. また, $d(T) = \max\{d(v) \mid v \in T\}$ で定義される $d(T)$ を T の**次数**という.

本論文では, v が u の先祖であるときを $u < v$ で表し, $u < v$ か $u = v$ であるときを $u \leq v$ で表す. 木 $T = (V, T)$ に対し, $V' = \{u \in V \mid u \leq v\}$, $E' = \{(u, w) \in E \mid u, v \in V'\}$, $r(T') = v$ となるような木 $T' = (V', E')$ を T の v を根とする**完全部分木** (*complete subtree*) といい, $T[v]$ で表す.

u と v が, 先行走査順 pre と後行走査順 $post$ に対して, $pre(u) \leq pre(v)$ かつ $post(u) \leq post(v)$ を満たすとき, u は v **の左にある**といい, $u \preceq v$ と表す.

根付き木の各ノードに有限文字集合 Σ から文字が割り当てられているとき, その根付き木を**ラベル付き** (*labeled*) といい, 割り当てられた文字をラベルという. ノード v のラベルを $l(v)$ で表し, 場合により v と $l(v)$ を同一視する. また, $\varepsilon \notin \Sigma$ を**空文字** (*blank*) といい, $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ とする. 根付き木の兄弟間の順序が与えられているとき, その根付き木を**順序木** (*ordered*) といい, そうでないときに**無順序木** (*unordered*) という.

木の列 $[T_1, .., T_n]$ を**森** (*forest*) という. 木 T とそのノード $v \in T$ について, $T[v]$ から v を削除してできる森を $T(v)$ で表す.

以降, 距離とマッピングについての議論をするにあたって, 木 T_1, T_2 を用いる. $i \in T_1, j \in T_2$ について, i, j の子をそれぞれ $i_1, \ldots, i_s$ と $j_1, \ldots, j_t$ で表す. 同様に, $T_1(i) = [T_1[i_1], \ldots, T_1[i_s]]$ と $T_2(j) = [T_2[j_1], \ldots, T_2[j_t]]$ をそれぞれ, $F_1(i_1, i_s), F_2(j_1, j_t)$ で表す.

定義 1 (編集操作) 木 T への**編集操作** (*edit operation*) は以下のように定義される.

1. **置換** (*substitution*): ノード $v \in T$ のラベルを変える.

2. **削除** (*deletion*): v' を親に持つノード $v \in T$ を削除し, v の子を v' の子にする. T が順序木の場合は, v があった位置に部分列として v の子を挿入し, 無順序木の場合は, v' の子と和集合を取る.

3. **挿入** (*insertion*): 削除の逆操作. ノード v を $v' \in T$ の子として挿入する. T が順序木の場合は, v' の子の部分列を左右の順序を変えないように v の子にできる. T が無順序木の場合は, v' の子の部分集合を v の子にできる.

ここで, ラベルの組 $(l_1, l_2) \in (\Sigma_\varepsilon \times \Sigma_\varepsilon \setminus \{(\varepsilon, \varepsilon)\})$ に対して, 編集操作を $(l_1 \mapsto l_2)$ と書き, $l_1 \neq \varepsilon$ かつ $l_2 \neq \varepsilon$ のときに置換を, $l_2 = \varepsilon$ のときに削除を, $l_1 = \varepsilon$ のときに挿入をそれぞれ表す. ノード v, w に対して, $(l(v) \mapsto l(w))$ を単に $(v \mapsto w)$ と表す. さらに, ラベルの組に, **コスト関数** (*cost function*)γ: $(\Sigma_\varepsilon \times \Sigma_\varepsilon \setminus \{(\varepsilon, \varepsilon)\}) \to \mathbb{R}^+$ を定義する. 特に断らない限り, コスト関数には $l_1 \neq l_2$ のとき $\gamma(l_1, l_2) = 1$ となる**単一コスト関数** (*unit cost function*) を用いる.

定義 2 (編集距離) γ をコスト関数とする. 編集操作 $e = (l_1 \mapsto l_2)$ に対して, $\gamma(e) = \gamma(l_1, l_2)$ で与えられ

る値を, 編集操作 e のコストという．また, 編集操作列 $E = e_1, \ldots, e_k$ に対して, $\gamma(E) = \sum_{i=i}^{k} \gamma(e_i)$ で与えられる値を, 編集操作列 E のコストという．そして, 木 T_1 と T_2 に対して, 以下のように定義される $\tau_{\mathrm{TAI}}(T_1, T_2)$ を, T_1 と T_2 間の**編集距離** (*edit distance*) という．

$$\tau_{\mathrm{TAI}}(T_1,T_2) = \min\left\{ \gamma(E) \;\middle|\; \begin{array}{l} E \text{ は } T_1 \text{から } T_2 \text{を得る} \\ \text{ための編集操作列} \end{array} \right\}.$$

定義 3 (マッピング) T_1 と T_2 を木とし, $M \subseteq V(T_1) \times V(T_2)$ とする．このとき, 任意の $(u_1, v_1), (u_2, v_2) \in M$ が (1)$u_1 = u_2 \iff v_1 = v_2$(一対一対応), (2)$u_1 \leq u_2 \iff v_1 \leq v_2$(先祖関係保存), (3)$u_1 \preceq u_2 \iff v_1 \preceq v_2$(兄弟関係保存) を満たすとき, 3 つ組 (M, T_1, T_2) を**順序 Tai マッピング** (*ordered Tai mapping*) という．また, 一対一対応と先祖子孫関係を満たす 3 つ組 (M, T_1, T_2) を**無順序 Tai マッピング** (*unordered Tai mapping*) という．誤解のない限り, (M, T_1, T_2) を単に M と書く．また, 順序 Tai マッピングや無順序 Tai マッピングを単に**マッピング**といい, $M \in \mathcal{M}_{\mathrm{TAI}}(T_1, T_2)$ と書く．

特に, 任意の $v \in T_1$ に対して, $(v, w) \in M$ となる $w \in T_2$ が存在するような M を, T_1 から T_2 への**包含マッピング** (*inclusion mapping*) という．このとき, w を $M(v)$ で表す．

M を T_1 から T_2 へのマッピングとし, γ をコスト関数とする．I_M と J_M をそれぞれ T_1 と T_2 のノードのうち, M に含まれないノードの集合, すなわち $I_M = \{u \in T_1 \mid (u, v) \notin M\}, J_M = \{v \in T_2 \mid (u, v) \notin M\}$ とする．このとき, M のコスト $\gamma(M)$ を以下のように与える．

$$\gamma(M) = \sum_{(u,v)\in M} \gamma(u,v) + \sum_{u \in I_M} \gamma(u,\varepsilon) + \sum_{v \in J_M} \gamma(\varepsilon, v).$$

定理 1 (Tai) 以下が成り立つ．

$$\tau_{\mathrm{TAI}}(T_1,T_2) = \min\{\gamma(M) \mid M \in \mathcal{M}_{\mathrm{TAI}}(T_1,T_2)\}.$$

定義 4 (部分森, 共通部分森) $T = (V, E)$ を木とする．

1. $V' \subseteq V$ であり, かつ, 任意の $u, v \in V'$ に対して, $u < v$ であるが $u < w$ かつ $w < v$ となる $w \in V'$ が存在しないときに $(u, v) \in E'$ となるような $F = (V', E')$ を, T の**埋め込み部分森** (*embedded subforest*) という．

2. $V' \subseteq V$, $E' = \{(u, v) \in E \mid u, v \in V'\}$ を満たす $F = (V', E')$ を, T の**誘導部分森** (*induced subforest*) という．

3. T の誘導部分森 $F = (V', E')$ が, 任意の F のノード $v \in V'$ について, その子孫がすべて V' の要素になっているとき, F を T の**完全部分森** (*complete subforest*) という．

さらに, F が木 T_1 と T_2 の埋め込み (誘導, 完全) 部分森であるとき, F を T_1 と T_2 の共通埋め込み (誘導, 完全) 部分森であるという．これら 3 つの区別が必要ないとき, これらを**共通部分森** (*common subforest*) という．

定義 5 (部分森中の部分木の並び) F を T_1 と T_2 の共通部分森とする． $v \in F$ に対して, $v \in (T_1 \cap F)$ と $v \in (T_1 \cap F)$ を, それぞれ v^1, v^2 と表記する．

1. T_1 で $u^1 \sqcup v^1 < v^1 \sqcup w^1$ となり, かつ, T_2 で $v^2 \sqcup w^2 < u^2 \sqcup v^2$ となるノード $u, v, w \in F$ が存在するとき, F を**ねじれている** (*twisting*) といい, そうでないとき**ねじれていない** (*non-twisting*) という．

2. 任意のノード $u, v, w \in F$ について, T_1 で $u^1 \sqcup v_1 = u^1 \sqcup w^1$ となり, かつ, その場合に限り, T_2 で $u^2 \sqcup v_2 = u^2 \sqcup w^2$ となるとき, F を**並行である** (*prallel*) という．

3. F が木であるとき, F を**部分木** (*subtree*) という．

4. F が木であり, $r(F) = r(T_1) = r(T_2)$ であるとき, F を**根保存部分木** (*root-preserving subtree*) という．

3 Tai マッピングの階層

本節では, Tai マッピングの変種を導入する．

定義 6 (マッピングの変種) T_1 と T_2 を木とし, $M \in \mathcal{M}_{\mathrm{TAI}}(T_1, T_2)$ とする． また, $M \setminus \{(r(T_1), r(T_2))\}$ を M^- で表す．

1. M が以下の条件を満たすとき、M を**劣制限マッピング** (*less-constrained mapping*) [8] といい、$M \in \mathcal{M}_{\mathrm{LESS}}(T_1, T_2)$ で表す。

$$\forall (u_1,v_1),(u_2,v_2),(u_3,v_3) \in M \left(\begin{array}{l} u_1 \sqcup u_2 < u_1 \sqcup u_3 \\ \quad \Longrightarrow v_2 \sqcup v_3 = v_1 \sqcup v_3 \end{array} \right).$$

2. M が以下の条件を満たすとき, M を**孤立部分木マッピング** (*isolated-subtree mapping*) [14] または, **制限マッピング** (*constrained mapping*) [16, 17]) といい, $M \in \mathcal{M}_{\mathrm{ILST}}(T_1, T_2)$ で表す。

$$\forall (u_1,v_1),(u_2,v_2),(u_3,v_3) \in M \Big(u_3 < u_1 \sqcup u_2 \iff v_3 < v_1 \sqcup v_2\Big).$$

3. M が以下の条件を満たすとき、M を**調和的マッピング** (*accordant mapping*) [7] または, **Lu マッピング** (*Lu's mapping* [9]) といい, $M \in \mathcal{M}_{\mathrm{ACC}}(T_1, T_2)$ で表す．

$$\forall (u_1,v_1),(u_2,v_2),(u_3,v_3) \in M \left(\begin{array}{l} u_1 \sqcup u_2 = u_1 \sqcup u_3 \\ \quad \iff v_1 \sqcup v_2 = v_1 \sqcup v_3 \end{array} \right).$$

4. M が以下の条件を満たすとき, M を **LCA 保存マッピング** (*LCA-preserving mapping*) または, **次数 2 マッピング** (*degree-2 mapping*) [19]) といい, $M \in \mathcal{M}_{\text{LCA}}(T_1,T_2)$ で表す.

$$\forall (u_1,v_1),(u_2,v_2) \in M^- \Big((u_1 \sqcup u_2, v_1 \sqcup v_2) \in M \Big).$$

5. M が $(r(T_1), r(T_2)) \in M$ を満たす LCA 保存マッピングであるとき, M を **LCA 保存根保存マッピング** (*LCA- and root-preserving mapping*) といい, $M \in \mathcal{M}_{\text{LCART}}(T_1,T_2)$ で表す.

6. M が以下の条件を満たすとき, M を**トップダウンマッピング** (*top-down mapping*) [2, 10] または, **次数 1 マッピング** (*degree-1 mapping*) といい, $M \in \mathcal{M}_{\text{TOP}}(T_1,T_2)$ で表す.

$$\forall (u,v) \in M^- \Big((par(u), par(v)) \in M \Big).$$

7. M が以下の条件を満たすとき, M を**ボトムアップマッピング** (*bottom-up mapping*) [7, 12, 15] といい, $M \in \mathcal{M}_{\text{BOT}}(T_1,T_2)$ で表す.

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

8. M が以下の条件を満たすとき, M を**断片マッピング** (*segmental mapping*) [6] といい, $M \in \mathcal{M}_{\text{SG}}(T_1,T_2)$ で表す.

$$\forall (u,v) \in M^- \left(\begin{array}{l} \exists (u',v') \in M \\ \quad \Big(\big(u' \in anc(u) \big) \wedge \big(v' \in anc(v) \big) \Big) \\ \qquad \Longrightarrow \Big((par(u), par(v)) \in M \Big) \end{array} \right).$$

9. M が $(r(T_1), r(T_2)) \in M$ を満たす断片マッピングであるとき, M を**トップダウン断片マッピング** (*top-down segmental mapping*) [6] といい, $M \in \mathcal{M}_{\text{TOPSG}}(T_1,T_2)$ で表す.

10. 森 F に対し, T_1 から F への包含マッピング M_1 と T_2 から F への包含マッピング M_2 が存在し, 任意の $(u,v) \in M$ に対し, $M_1(u) = M_2(v)$ が成り立つとき, M を**アライメント可能マッピング** (*alignable mapping*) [7] といい, $M \in \mathcal{M}_{\text{ALN}}(T_1,T_2)$ で表す.

11. マッピング $\mathtt{A}, \mathtt{B}$ に対して, $\mathcal{M}_{\mathtt{A}}(T_1,T_2) \cap \mathcal{M}_{\mathtt{B}}(T_1,T_2)$ を $\mathcal{M}_{\mathtt{AB}}(T_1,T_2)$ と表す. 本稿では以下の組み合わせを扱う.

$$\begin{array}{l} \mathcal{M}_{\text{ILSTSG}}(T_1,T_2),\ \mathcal{M}_{\text{ACCSG}}(T_1,T_2), \\ \mathcal{M}_{\text{LCASG}}(T_1,T_2),\ \mathcal{M}_{\text{SGALN}}(T_1,T_2), \\ \mathcal{M}_{\text{BOTALN}}(T_1,T_2),\ \mathcal{M}_{\text{ILSTBOT}}(T_1,T_2), \\ \mathcal{M}_{\text{ACCBOT}}(T_1,T_2),\ \mathcal{M}_{\text{LCABOT}}(T_1,T_2). \end{array}$$

定理 2 マッピング $\mathcal{M}_{\mathtt{A}}(T_1,T_2)$ は図 1 の階層を成す.

注意 1 T_1 と T_2 の共通部分森をマッピング M によって特徴づけることができる. I_M と J_M は, それぞれ T_1 と T_2 のうち, マッピングに含まれないノードの集合である. このとき, $T_1 = (V_1, E_1)$ と $T_2 = (V_2, E_2)$ に対して, M から, $F' = V_1 \setminus I_M = V_2 \setminus J_M$ となるような共通部分森 $F' = (V', E')$ が得られる. このような共通部分森を M **による共通部分森**という.

断片マッピングは親子関係を保存する必要があるため, 断片マッピングおよびその部分マッピング (SG, SGALN, ACCSG, LCASG, TOP) による共通部分森は誘導部分森となる. 操作的には, 削除と挿入を葉あるいは根のみに適用することに対応する. 特に TOP では, 削除と挿入を葉にのみ適用する.

ボトムアップマッピングは完全部分木を保存する必要があるため, ボトムアップマッピングおよびその部分マッピング (BOT, BOTALN, ACCBOT, LCABOT, ISO) による共通部分森は完全部分森となる. 操作的には, 削除と挿入を根のみに適用することに対応する.

注意 2 LCA 保存マッピングは含まれるノードすべてに対してその最近共通先祖がマッピングに含まれるため, LCA 保存マッピングおよびその部分マッピング (LCA, LCASG, LCABOT, LCART, TOP, ISO) による共通部分森は部分木となる. 特に, LCA 保存根保存マッピングは与えられた木の根の組を含むため, LCA 保存根保存マッピングおよびその部分マッピング (LCART, TOP, ISO) による共通部分森は根保存部分木になる.

孤立部分木マッピングの定義は共通部分森が並行である条件と同じであるため, 孤立部分木マッピングおよびその部分マッピング (ACC, ACCSG, ACCBOT) による共通部分森は並行になる.

ねじれている共通部分森と同様に, 木 T_1, T_2 間のマッピングに関しても, ねじれを定義することができる. $(v_1 \sqcup v_2 < v_2 \sqcup v_3) \wedge (v_2 \sqcup v_3 < v_1 \sqcup v_2)$ を満たす組 $(v_1,w_1),(v_2,w_2),(v_3,w_3) \in M$ が存在するとき, M はねじれているといい, 組 $(v_1,w_1),(v_2,w_2),(v_3,w_3)$ を, M の**ねじれ** (*twist*) という.

ノード $w_1, w_2, w_3 \in T$ に対して, $(w_1 \sqcup w_2 \leq w_2 \sqcup w_3) \equiv (w_1 \sqcup w_3 = w_2 \sqcup w_3)$ と $\neg(w_2 \sqcup w_3 \leq w_1 \sqcup w_2) \equiv (w \sqcup w_2 \leq w_2 \sqcup w_3)$ が常に成り立つ. よって, マッピ

ング M が劣制限マッピングである場合に限り M はねじれを持たないことを論理式の変形によって示すことができる. したがって, アライメント可能マッピングおよびその部分マッピング (Aln, SgAln, BotAln) による共通部分森はねじれを持たない.

定義 7 (編集距離の変種) 図 1 で示した階層の各マッピング $\mathcal{M}_{\mathtt{A}}(T_1,T_2)$ に対し, 編集距離 τ_{TAI} の変種 $\tau_{\mathtt{A}}(T_1,T_2)$ を, $\mathcal{M}_{\mathtt{A}}(T_1,T_2)$ に含まれるマッピングの最小コストとして定義する:

$$\tau_{\mathtt{A}}(T_1,T_2)=\min\{\gamma(M)\mid M\in\mathcal{M}_{\mathtt{A}}(T_1,T_2)\}.$$

特に, 順序木 (無順序木) 間の $\tau_{\mathtt{A}}(T_1,T_2)$ を, $\tau^o_{\mathtt{A}}(T_1,T_2)$ ($\tau^u_{\mathtt{A}}(T_1,T_2)$) と表す.

4 距離計算の時間計算量

本節では, 順序木と無順序木での編集距離計算の時間計算量について議論する. 以降, $n=|T_1|, m=|T_2|$ とし, $D=\max\{d(T_1),d(T_2)\}, d=\min\{d(T_1),d(T_2)\}$ とする. マッピング $\mathtt{A}$ に対する, 順序木と無順序木の距離をそれぞれ $\tau^o_{\mathtt{A}},\tau^u_{\mathtt{A}}$ で表し, 順序森と無順序森の距離をそれぞれ $\delta^o_{\mathtt{A}},\delta^u_{\mathtt{A}}$ で表す.

トップダウン距離 τ_{Top} に対して以下が成り立つ.

注意 3 [6] より, 任意の組 $(i,j)\in T_1\times T_2(1\le i\le n, 1\le j\le m)$ に対する $T_1[i]$ と $T_2[j]$ の順序木トップダウン距離 $\tau^o_{\mathrm{Top}}(T_1[i],T_2[j])$ は $O(nm)$ 時間で計算可能である. 一方, $\tau^u_{\mathrm{Top}}(T_1[i],T_2[j])$ は図 2 の再帰式で計算できる [15]. ここで, $\mathtt{BM}$ は完全 2 部グラフ $G_M=(X,Y,E)$ における, 重み ω による最大重み付き 2 部マッチングである. ただし, X,Y はそれぞれ T_1 の i, T_2 の j の子ノードの集合であり, 辺 $(i',j')\in E$ の重み $\omega((i',j'))$ は, $\tau^u_{\mathrm{Top}}(T_1[i'],\emptyset)+\tau^u_{\mathrm{Top}}(\emptyset,T_2[j'])-\tau^u_{\mathrm{Top}}(T_1[i'],T_2[j'])$ である [15, 19]. ゆえに, 任意の $(i,j)\in T_1\times T_2$ に対する $\tau^u_{\mathrm{Top}}(T_1[i],T_2[j])$ を $O(nmd)$ 時間で計算可能である.

$$\begin{aligned}&\tau^u_{\mathrm{Top}}(T_1[i],T_2[j])=\delta^u_{\mathrm{Top}}(T_1(i),T_2(j))+\gamma(i,j),\\&\delta^u_{\mathrm{Top}}(T_1(i),T_2(j))=\sum_{T\in T_1(i)}\sum_{i'\in T}\gamma(i',\varepsilon)\\&\quad+\sum_{T\in T_2(j)}\sum_{j'\in T}\gamma(\varepsilon,j')-\sum_{(i',j')\in\mathtt{BM}}\omega((i',j')).\end{aligned}$$

図 2: $\tau^u_{\mathrm{Top}}(T_1,T_2)$ を計算する再帰式.

注意 3 より, 任意の $(i,j)\in T_1\times T_2$ に対する $\tau^o_{\mathrm{Top}}(T_1[i],T_2[j])$ を事前計算することで, 以下の定理が成り立つ.

定理 3 $\tau^o_{\mathrm{LcaRt}}(T_1,T_2)$ と $\tau^u_{\mathrm{LcaRt}}(T_1,T_2)$ は $O(nm)$ 時間と $O(nmd)$ 時間で計算できる.

定理 4 $\tau^o_{\mathrm{LcaSg}}(T_1,T_2)$ と $\tau^u_{\mathrm{LcaSg}}(T_1,T_2)$ は $O(nm)$ 時間と $O(nmd)$ 時間で計算できる.

定理 5 $\tau^o_{\mathrm{AccSg}}(T_1,T_2)$ と $\tau^o_{\mathrm{SgAln}}(T_1,T_2)$ は $O(nm)$ 時間と $O(nmD^2)$ 時間で計算できる.

定理 6 $\tau^u_{\mathrm{AccSg}}(T_1,T_2)$ は $O(nmd)$ 時間で計算できる.

ボトムアップ距離 τ_{Bot} に対して以下が成り立つ.

注意 4 2 つの木 T_1 と T_2 に対して, $I_M=J_M=\emptyset$ となるようなマッピング $M\in\mathcal{M}^o_{\mathrm{TAI}}(T_1,T_2)$(あるいは $M\in\mathcal{M}^u_{\mathrm{TAI}}(T_1,T_2)$) を, ラベル無視同型マッピングという. ラベル無視同型マッピング $M\in\mathcal{M}^o_{\mathrm{TAI}}(T_1,T_2)$($M\in\mathcal{M}^o_{\mathrm{TAI}}(T_1,T_2)$) が存在する T_1 と T_2 を, ラベル無視順序木 (無順序木) 同型といい, $T_1\equiv^o_l T_2$($T_1\equiv^u_l T_2$) で表す. また, $\mathit{diff}^r(T_1,T_2)(r\in\{o,u\})$ を以下のように定義する. ただし, M はラベル無視同型マッピングである.

$$\begin{aligned}&\mathit{diff}^r(T_1,T_2)\\&=\begin{cases}\sum_{(i,j)\in M}\gamma(i,j) & (T_1\equiv^r_l T_2),\\ \sum_{i\in T_1}\gamma(i,\varepsilon)+\sum_{j\in T_2}\gamma(\varepsilon,j) & (T_1\not\equiv^r_l T_2).\end{cases}\end{aligned}$$

$T_1\equiv^r_l T_2$ のチェックや $\mathit{diff}^r(T_1,T_2)$ の計算は, $O(n+m)$ 時間で可能である [12].

注意 4 の関数 diff を利用することで, 以下の定理が成り立つ.

定理 7 $\tau^o_{\mathrm{LcaBot}}(T_1,T_2)$ と $\tau^u_{\mathrm{LcaBot}}(T_1,T_2)$ は $O(nm)$ 時間で計算できる.

定理 8 $\tau^o_{\mathrm{AccBot}}(T_1,T_2)$ と $\tau^o_{\mathrm{BotAln}}(T_1,T_2)$ は $O(nm)$ 時間と $O(nmD^2)$ 時間で計算できる.

定理 9 $\tau^u_{\mathrm{AccBot}}(T_1,T_2)$ は $O(nmd)$ 時間で計算できる.

一方, ねじれ無し部分森に関連する無順序木距離に対して, 以下の定理が成り立つ.

定理 10 $\tau^u_{\mathrm{SgAln}}(T_1,T_2)$ と $\tau^u_{\mathrm{BotAln}}(T_1,T_2)$ を計算する問題は MAX SNP 困難である.

定理 11 T_1 と T_2 の次数がある定数以下となるとき, $\tau^u_{\mathrm{SgAln}}(T_1,T_2)$ と $\tau^u_{\mathrm{BotAln}}(T_1,T_2)$ は多項式時間で計算可能である.

5 まとめ

本稿では, 共通部分森を通してマッピングの階層を考察し, 新たなマッピングを導入することで図 1 の階層を与えた. そして, 上記マッピングの最小コストとして編集距離の変種を定式化し, その計算時間について考察した. 結果を表 1 にまとめる.

表 1: 図 1 の $\mathcal{M}_{\mathtt{A}}(T_1, T_2)$ に対する距離 $\tau_{\mathtt{A}}(T_1, T_2)$ の時間計算量. ここで, $n = \max\{|T_1|, |T_2|\}$, $m = \min\{|T_1|, |T_2|\}$, $D = \max\{d(T_1), d(T_2)\}$, $d = \min\{d(T_1), d(T_2)\}$ である.

マッピング $\mathcal{M}_{\mathtt{A}}(T_1, T_2)$	順序木 $\tau^o_{\mathtt{A}}$		無順序木 $\tau^u_{\mathtt{A}}$		(次数が定数以下)	
Tai	$O(nm^2(1+\log\frac{n}{m}))$	[3]	MAX SNP	[1, 4, 18]	MAX SNP	[4]
Aln	$O(nmD^2)$	[5]	MAX SNP	[5]	多項式時間	[5]
Ilst	$O(nm)$	[16]	$O(nmd)$	[15]	$O(nm)$	←
Acc	$O(nm)$	[7]	$O(nmd)$	[15]	$O(nm)$	←
Lca	$O(nm)$	[19]	$O(nmd)$	[19]	$O(nm)$	←
LcaRt	$O(nm)$	定理 3	$O(nmd)$	定理 3	$O(nm)$	←
Sg	$O(nm)$	[6]	MAX SNP	[15]	MAX SNP	[15]
SgAln	$O(nmD^2)$	定理 5	MAX SNP	定理 10	多項式時間	定理 11
AccSg	$O(nm)$	定理 5	$O(nmd)$	定理 6	$O(nm)$	←
LcaSg	$O(nm)$	定理 5	$O(nmd)$	定理 5	$O(nm)$	←
Top	$O(nm)$	[2, 10]	$O(nmd)$	[15]	$O(nm)$	←
Bot	$O(nm)$	[15]	MAX SNP	[15]	MAX SNP	[15]
BotAln	$O(nmD^2)$	定理 8	MAX SNP	定理 10	多項式時間	定理 11
AccBot	$O(nm)$	定理 8	$O(nmd)$	定理 9	$O(nm)$	←
LcaBot	$O(nm)$	定理 7	$O(nm)$	定理 7	←	
Iso	$O(n+m)$	[13]	$O(n+m)$	[13]	←	

参考文献

[1] T. Akutsu, D. Fukagawa, M. M. Halldórsson, A. Takasu, K. Tanaka: *Approximation and parameterized algorithms for common subtrees and edit distance between unordered trees*, Theoret. Comput. Sci. **470**, 10–22, 2013.

[2] S. S. Chawathe: *Comparing hierarchical data in external memory*, Proc. VLDB'99, Proc. of the 25th International Conference on Very Large Data Bases (VLDB'99), 90–101, 1999.

[3] E. D. Demaine, S. Mozes, B. Rossman, O. Weimann: *An optimal decomposition algorithm for tree edit distance*, ACM Trans. Algo. **6**, 2009.

[4] K. Hirata, Y. Yamamoto, T. Kuboyama: *Improved* MAX SNP*-hard results for finding an edit distance between unordered trees*, Proc. CPM 2011, LNCS **6661**, 402–415, 2011.

[5] T. Jiang, L. Wang, K. Zhang: *Alignment of trees – an alternative to tree edit*, Theoret. Comput. Sci. **143**, 137–148, 1995.

[6] T. Kan, S. Higuchi, K. Hirata: *Segmental mapping and distance for rooted ordered labeled trees*, Fundamenta Informaticae **132**, 1–23, 2014.

[7] T. Kuboyama: *Matching and learning in trees*, Ph.D thesis, University of Tokyo, 2007.

[8] C. L. Lu, Z.-Y. Su, C. Y. Yang: *A new measure of edit distance between labeled trees*, Proc. COCOON'01, LNCS **2108**, 338–348, 2001.

[9] S.-Y. Lu: *A tree-to-tree distance and its application to cluster analysis*, IEEE Trans. Pattern Anal. Mach. Intell. **1**, 219–224, 1979.

[10] S. M. Selkow: *The tree-to-tree editing problem*, Inform. Process. Lett. **6**, 184–186, 1977.

[11] K.-C. Tai: *The tree-to-tree correction problem*, J. ACM **26**, 422–433, 1979.

[12] G. Valiente: *An efficient bottom-up distance between trees*, Proc. SPIRE'01, 212–219, 2001.

[13] G. Valiente: *Algorithms on trees and graphs*, Springer, 2002.

[14] J. T. L. Wang, K. Zhang: *Finding similar consensus between trees: An algorithm and a distance hierarchy*, Pattern Recog. **34**, 127–137, 2001.

[15] Y. Yamamoto, K, Hirata, T. Kuboyama: *Tractable and intractable variations of unordered tree edit distance*, Internat. J. Found. Comput. Sci. **25**, 307–330, 2014.

[16] K. Zhang: *Algorithms for the constrained editing distance between ordered labeled trees and related problems*, Pattern Recog. **28**, 463–474, 1995.

[17] K. Zhang: *A constrained edit distance between unordered labeled trees*, Algorithmica **15**, 205-222, 1996.

[18] K. Zhang, T. Jiang: *Some* MAX SNP*-hard results concerning unordered labeled trees*, Inform. Process. Lett. **49**, 249–254, 1994.

[19] K. Zhang, J. Wang, D. Shasha: *On the editing distance between undirected acyclic graphs*, Internat. J. Found. Comput. Sci. **7**, 43-58. 1996.