人工知能学会研究会資料
SGI-FPAI-B503-03

根付きラベル付き順序木のトップダウン距離計算のSETH困難性

On SETH-Hardness of Computing Top-Down Distance between Rooted Ordered Labeled Trees

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

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

Abstract: In this paper, we show that, if the problem of computing the top-down distance between either (1) height-1 or (2) binary rooted ordered labeled trees can be computed in $O(n^{2-\varepsilon})$ time for a constant $\varepsilon > 0$, then SETH (Strongly Exponential Time Hypothesis) refutes.

1 はじめに

ウェブマイニングにおける HTML 文書や XML データ, および, バイオインフォマティクスにおける DNA データや糖鎖データなどにおける木構造データの比較は, データマイニングにおける重要な問題の一つである. 本論文では, 木構造データを**根付きラベル付き順序木**(rooted labeled ordered trees, 以後, 単に**木**(tree) という) として定式化し, 木間の距離に着目する.

編集距離(edit distance) は, 最も代表的な木間の距離である [8]. この編集距離は, **置換**(substitution), **削除**(deletion), **挿入**(insertion) という**編集操作**を適用することで, 一方の木から他方の木へ変換する最小コストとして定式化される. 編集距離は, 先祖子孫関係と兄弟関係を保存する 2 つの木間のノードの対応関係である **Tai マッピング**(Tai mapping, 以後, 単に**マッピング**(mapping) という) [8] に対して, 可能な Tai マッピングの最小コストと一致する [8]. 順序木の編集距離は, 2 つの木のノード数の最大値 n に対して $O(n^3)$ 時間で計算可能である [4].

編集距離は木を比較する際に一般的な尺度であるが, 応用先によっては一般的過ぎる場合がある. そこで, より構造を考慮した距離として, **トップダウン**(top-down, または**次数** 1(degree-1)) 距離 [3, 7], **LCA**(least common ancestor) **保存**(LCA-preserving, または**次数** 2(degree-2)) 距離 [12], **制限**(constrained, または**孤立部分木**(isolated-subtree)) 距離 [11], **ボトムアップ**(bottom-up) 距離 [5, 6, 9], **断片**(segmental) 距離 [5] などの距離が用途に応じて考案されてきた. これらの距離は, それぞれ Tai マッピングの変種である, **トップダウンマッピング**(top-down mapping) [3, 7], **LCA 保存マッピング**(LCA-preserving mapping) [12], **制限マッピング**(constrained mapping) [11], **ボトムアップマッピング**(bottom-up mapping) [5, 6, 9], **断片マッピング**(segmental mapping) [5] の最小コストとして定式化されている. これらの距離はすべて $O(n^2)$ 時間で計算可能である [3, 5, 7, 11, 12].

本論文では, 最も制限された編集距離の一つであり, トップダウンマッピングの最小コストであるトップダウン距離 [3, 7] に着目する. ここで, トップダウンマッピングとは, 2 つの木の根の組と, 根からのパスに含まれるノードの組からなる Tai マッピングであり, 常にボトムアップマッピング以外の上記のマッピングの条件を満たす [5, 6]. 操作的には, トップダウン距離は削除と挿入を葉にしか適用できない編集距離である.

そして本論文では, (1) 高さ 1 の木, もしくは, (2) 二分木のトップダウン距離がある定数 $\varepsilon > 0$ に対して $O(n^{2-\varepsilon})$ 時間で計算可能ならば, SETH (Strongly Exponential Time Hypothesis) [1, 2, 10] に反すること, つまり, SETH の下ではトップダウン距離は $O(n^{2-\varepsilon})$ 時間で計算不可能 (SETH 困難) であることを示す.

木の問題に対する SETH 困難性として, Abboud ら [1] は, **部分木同型**(subtree isomorphism) について, 特に, **無順序木**(unordered tree) の困難性を扱っている. ただし, [1] における無順序二分木の部分木同型の SETH 困難性は, その証明中, 誘導部分木の部分木同型が埋め込

*連絡先：九州工業大学情報工学研究院知能情報工学研究系
〒 820-8502 福岡県飯塚市川津 680-4
E-mail: hirata@ai.kyutech.ac.jp

み部分木の部分木同型となっている箇所があり, このままではこの SETH 困難性は成り立たない. 一方, 本論文のトップダウン距離計算の SETH 困難性は, 共通トップダウン誘導部分木同型と対応するため, Abboud ら [1] の結果よりも強い順序二分木の最大共通トップダウン誘導部分木の SETH 困難性を導くことになる.

2 準備

非巡回連結グラフを**木**(tree) という. 木 $T = (V, E)$ に対して, V と E をそれぞれ $V(T)$ と $E(T)$ と表す. また, T の**サイズ**(size)$|V(T)|$ を, 単に $|T|$ と表す. さらに, $v \in V(T)$ を単に $v \in T$ と表す. 空の木を $\emptyset$ で表す.

ノード r を**根**(root) として選んだ木を**根付き木**(rooted tree) という. 根付き木 T の根を $r(T)$ と表す. ノード r を根とする根付き木の各ノード v に対して, v から r への一意なパスを $UP_r(v)$ で表す.

ノード $v(\neq r)$ に対して, $UP_r(v) - \{v\}$ 上のノードを v の**先祖**(ancestor) という. また, v に隣接する先祖のノードを v の**親**(parent) といい, $par(v)$ で表す. v が u の親であるとき, u を v の**子**(child) といい, v が u の先祖であるとき, u を v の**子孫**(descendant) という. 本論文では, v が u の先祖であるとき $u < v$ と表し, $u < v$ または $u = v$ であるとき $u \leq v$ と表す.

子を持たないノードを**葉**(leaf) という. 木 T のすべての葉の集合を $lv(T)$ で表す. 木 T とノード $v \in T$ に対して, v を根とし, v の子孫をすべて含む木を v を根とする T の**完全部分木**といい, $T(v)$ と表す.

$v \in T$ に対して, $UP_r(v)$ の辺の数を v の**高さ**(height) といい $h(v)$ で表す. また, $\max\{h(v) \mid v \in T\}$ を木 T の**高さ**(height) といい $h(T)$ で表す. $v \in T$ の子の数を v の**次数**(degree) といい $d(v)$ で表す. また, $\max\{d(v) \mid v \in T\}$ を木 T の**次数**(degree) といい $d(T)$ で表す. 特に, $h(T) = 1$ となる木 T を**高さ 1 木**(height-1 tree), $d(T) = 2$ となる木 T を**二分木**(binary tree) という.

根付き木 T に対して, $v \in T$ の子ノードを $v_1, \ldots, v_i$ とする. このとき, $T(v)$ において, v を訪問した後に $T(v_k)$ $(1 \leq k \leq i)$ を再帰的に訪問する走査を $T(v)$ の**先行順走査**(preorder traversal) という. 同様に, $T(v_k)$ $(1 \leq k \leq i)$ を訪問した後に v を訪問する走査を $T(v)$ の**後行順走査**という. 木 T に対して, 先行順走査で T のすべてのノードを訪問した際の順番を $pre(v)$, 後行順走査で訪問した際の順番を $post(v)$ と表す. T のノード u と v に対して, $pre(u) \leq pre(v)$ かつ $post(u) \leq post(v)$ となるとき, u は v の**左である**(to the left), もしくは, v は u の**右である**(to the right) といい, $u \preceq v$ と表す.

順序 $\preceq$ が固定された根付き木を**順序木**(ordered tree) といい, そうでない木を**無順序木**(unordered tree) という. 各ノード v にアルファベット Σ の要素 $l(v)$ が割り当てられている根付き木を**ラベル付き木**(labeled tree) という. 以後, v と $l(v)$ を同一視する. 本論文では, 根付きラベル付き順序木を単に木という.

定義 1 (編集操作) 木 T の**編集操作**(edit operations) を以下のように定義する. 図 1 は編集操作の直観的な説明である.

1. **置換**(substitution): T のノード v のラベルを書き換える.

2. **削除**(deletion): T の, 親 v' を持つノード v を削除し, v の子を v' の子として接続する. このとき, 子は v の位置に接続し, v' の子の部分列として, 左右順序関係を保存したまま接続する.

3. **挿入**(insertion): 削除の逆操作. T のノード v' の子の連続部分列の位置にノード v を子ノードとして接続し, 連続部分列を v の子ノードとして接続する.

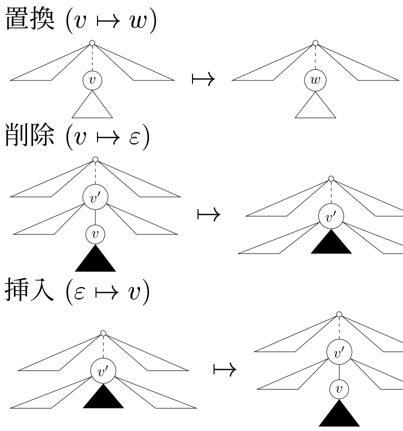

図 1: 木に対する編集操作.

$\varepsilon \notin \Sigma$ を**空白**(blank) 記号とし, $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ とする. このとき, 編集操作を $(l_1 \mapsto l_2)$ で表す. ただし, $(l_1, l_2) \in (\Sigma_\varepsilon \times \Sigma_\varepsilon - \{(\varepsilon, \varepsilon)\})$ である. 編集操作としては, $l_1 \neq \varepsilon$ かつ $l_2 \neq \varepsilon$ のときに置換を, $l_2 = \varepsilon$ のときに削除を, $l_1 = \varepsilon$ のときに挿入を表す. ノード v と w に対して, $(l(v) \mapsto l(w))$ を単に $(v \mapsto w)$ と表す.

ラベルの対に対して, $\gamma : (\Sigma_\varepsilon \times \Sigma_\varepsilon - \{(\varepsilon, \varepsilon)\}) \mapsto \mathbf{R}$ を**コスト関数**(cost function) という. コスト関数 γ は**メトリック**(metric), すなわち, $\gamma(l_1, l_2) \geq 0$, $\gamma(l_1, l_2) = 0 \iff l_1 = l_2$, $\gamma(l_1, l_2) = \gamma(l_2, l_1)$, $\gamma(l_1, l_3) \leq \gamma(l_1, l_2) + \gamma(l_2, l_3)$ を満たす. 特に, $\gamma(l_1, l_2) = \gamma(l_1, \varepsilon) + \gamma(\varepsilon, l_2)$ となるコスト関数 γ を**インデルコスト関数**(indel cost function) といい, $\gamma(l_1, \varepsilon) = \gamma(\varepsilon, l_2) = 1$ となるインデルコスト関数 γ を**単一インデルコスト関数**(unit indel cost function) という.

定義 2 (編集距離) コスト関数 γ に対して, 編集操作 $e = l_1 \mapsto l_2$ の**コスト**(cost) は $\gamma(e) = \gamma(l_1, l_2)$ で与えられる. また, 編集操作の列 $E = e_1, \ldots, e_k$ の**コスト**(cost) は $\gamma(E) = \sum_{i=1}^{k} \gamma(e_i)$ で与えらえる. このとき, 木 T と S の**編集距離**(edit distance)$\textsc{Dist}(T, S)$ を以下のように定義する.

$$\textsc{Dist}(T,S) = \min \left\{ \gamma(E) \;\middle|\; \begin{array}{l} E \text{ は } T \text{ を } S \text{ に} \\ \text{変換するための} \\ \text{編集操作の列} \end{array} \right\}.$$

編集距離は, 以下のマッピングと密接な関係がある [8].

定義 3 (マッピング) T と S を木とする. このとき, 任意の $(v_1, w_1), (v_2, w_2) \in M$ が以下の条件を満たすような集合 $M \subseteq V(T) \times V(S)$ を T と S の **Tai マッピング**(Tai mapping, 以後, 単に**マッピング**(mapping)) といい, $M \in \mathcal{M}_{\textsc{Tai}}(T, S)$ と表す.

1. $v_1 = v_2 \iff w_1 = w_2$.
2. $v_1 \leq v_2 \iff w_1 \leq w_2$.
3. $v_1 \preceq v_2 \iff w_1 \preceq w_2$.

$M \in \mathcal{M}_{\textsc{Tai}}(T, S)$ に対して, M_I を T のノードのうち M に含まれないものの集合, M_J を S のノードのうち M に含まれないものの集合とする. このとき, M の**コスト**(cost) を以下のように定義する.

$$\begin{aligned} \gamma(M) &= \sum_{(v,w)\in M} \gamma(v \mapsto w) \\ &\quad + \sum_{v \in M_I} \gamma(v \mapsto \varepsilon) + \sum_{w \in M_J} \gamma(\varepsilon \mapsto w). \end{aligned}$$

定理 1 (Tai [8]) 木 T と S に対して, 以下が成り立つ.

$$\textsc{Dist}(T,S) = \min\{\gamma(M) \mid M \in \mathcal{M}_{\textsc{Tai}}(T,S)\}.$$

編集距離の変種として, 本論文ではトップダウン距離について議論する.

定義 4 (トップダウン距離 [3, 7]) 木 T と S に対して, $M \in \mathcal{M}_{\textsc{Tai}}(T, S)$ とし, $M^- = M - \{(r(T), r(S))\}$ とする. 以下の条件を満たすような M を T と S の**トップダウンマッピング**(top-down mapping) といい, $M \in \mathcal{M}_{\textsc{Top}}(T, S)$ と表す.

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

さらに, T と S の**トップダウン距離**(top-down distance, または**次数 1 距離**(degree-1 distance)) を以下のように定義する.

$$\textsc{TopDown}(T,S) = \min\{\gamma(M) \mid M \in \mathcal{M}_{\textsc{Top}}(T,S)\}.$$

定義 4 より, トップダウン距離は削除と挿入を葉のみに適用する編集距離である.

定義 5 (部分木) $T = (V, E)$ と $T' = (V', E')$ を木とし, $V' \subseteq V$ とする.

1. 任意の $u, v \in V'$ に対して, $u < v$ であり, $u < w$ かつ $w < v$ なる $w \in V'$ が存在しないときに $(u, v) \in E'$ となるとき, T' を T の**埋め込み部分木**(embedded subtree) という.
2. 任意の $u, v \in V'$ に対して, $(u, v) \in E$ のときに $(u, v) \in E'$ となるとき, T' を T の**誘導部分木**(induced subtree) という.

完全部分木は誘導部分木の特別な場合である. 以後, 埋め込み部分木, 誘導部分木, 完全部分木を単に部分木ともいう.

定義 6 (最大共通部分木) 木 T と S に対して, 木 T' が T と S の部分木となるとき, T' を T と S の**共通部分木**(common subtree) という. また, T と S の共通部分木のうち, ノード数が最大の木を, T と S の**最大共通部分木**(largest common subtree) という. さらに, 2 つの木の共通部分木がそれぞれの根を含むとき, **トップダウン**(top-down) であるという.

木 T と S の共通部分木はマッピング M を用いて特徴づけることができる. M_I (M_J) は T (S) のノードのうち M に含まれないものの集合であることに注意すると, $T = (V_1, E_1)$ と $S = (V_2, E_2)$ に対して, $V' = V_1 - M_I = V_2 - M_J$ であり, M が T と S の共通部分木 $T' = (V', E')$ を含むとき, T' を T と S の M **による共通部分木**(common subtree through M) という. したがって, インデルコスト関数における, トップダウンマッピングによる共通部分木は, 共通トップダウン誘導部分木である.

3 SETH 困難性

以下の 2 つの問題は, SETH(Strongly Exponential Time Hypothesis) の下で, 定数 $\varepsilon > 0$ に対して $O(n^{2-\varepsilon})$ 時間では計算不可能 (SETH 困難) である [2, 10].

> $\textsc{StringEditDistance}$ [2]
> 入力: Σ 上の 2 つの文字列 t と s.
> 出力: 文字列編集距離 $\textsc{StrDist}(t, s)$.

> $\textsc{BinaryOthogonalVectors}$ [10]
> 入力: $\{0,1\}^D$ 上の N 個の 2 値ベクトルからなる 2 つのリスト A と B.
> 出力: $\alpha \cdot \beta = 0$ を満たす対 $(\alpha, \beta) \in A \times B$ が存在するならば真を返し, そうでなければ偽を返す.

本節では, これらの SETH 困難問題からトップダウン距離計算問題へ還元する.

定理 2 定数 $\varepsilon > 0$ に対して, 木 T と S がたとえ高さ 1 木としても, $\textsc{TopDown}(T, S)$ は SETH の下で $O(n^{2-\varepsilon})$ 時間では計算不可能である.

証明. 問題 $\textsc{StringEditDistance}$ から還元する. Σ 上の文字列 $t = t_1 \cdots t_m$, $s = s_1 \cdots s_n$ に対して, 図 2 の木 T と S を考える. ここで, T の $t_i (1 \le i \le m)$ と S の $s_j (1 \le j \le n)$ は, そのノードのラベルを表し, 根のラベルは a とする. このとき, 明らかに $\textsc{TopDown}(T, S) = \textsc{StrDist}(t, s)$ である. □

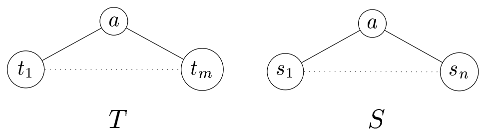

図 2: 定理 2 の証明に用いる木 T と S.

定理 3 定数 $\varepsilon > 0$ に対して, 木 T と S がたとえ二分木としても, $\textsc{TopDown}(T, S)$ は SETH の下で $O(n^{2-\varepsilon})$ 時間では計算不可能である.

証明. 問題 $\textsc{BinaryOthogonalVectors}$ から還元する. 証明では, 0, 1, a のみでラベル付けされた木を用いる. また, 単一インデルコスト関数をコスト関数として用いる. u を根に持ち, u の子が左から木 V と木 W となるような木を $u(V, W)$ で表す. また, u を根に持ち, u の子が木 V となるような木を $u(V)$ で表す.

$A = \{\alpha_1, \ldots, \alpha_N\}$, $B = \{\beta_1, \ldots, \beta_N\}$ とする. α_i および β_i の j 番目 $(1 \le j \le D)$ の要素を, それぞれ $\alpha_i[j]$ および $\beta_i[j]$ と表す.

$\alpha_i \in A$ とする. $\alpha_i[j]$ $(1 \le j \le D)$ に対して, 木 A_i^j を $\alpha_i[j] = 0$ ならば $a(1, 0)$, $\alpha_i[j] = 1$ ならば $a(0, 0)$ として構成する. そして, A_i を以下のように構成する.

$$A_i = a(A_i^1, a(A_i^2, \cdots, a(A_i^D, a(a(a))))).$$

$\beta_i \in B$ とする. $\beta_i[j] (1 \le j \le D)$ に対して, 木 B_i^j を $\beta_i[j] = 0$ ならば $a(0)$, $\beta_i[j] = 1$ ならば $a(1)$ として構成する. そして, B_i を以下のように構成する.

$$B_i = a(B_i^1, a(B_i^2, \cdots, a(B_i^D, a(a(a))))).$$

さらに, $C^j (1 \le j \le D)$ を $a(0, 1)$ とし, C を以下のように構成する.

$$C = a(C^1, a(C^2, \cdots, a(C^D, a(a)))).$$

補題 1 $|A_i| = 4D + 3$ かつ $|C| = 4D + 2$.

証明. A_i と C の構成方法より自明である. □

補題 2 $\alpha_i \cdot \beta_i = 0$ ならば $\textsc{TopDown}(A_i, B_i) = D$ であり, そうでなければ $\textsc{TopDown}(A_i, B_i) \ge D + 2$ である. また, $\textsc{TopDown}(C, B_i) = D + 1$ である.

証明. A_i と B_i の最小コストのトップダウンマッピングは, 根から葉までの $D + 3$ 個の組 (a, a) を要素とし, そのコストは 0 となる. また, 以下の式が成り立つ.

$$\begin{aligned}
&\textsc{TopDown}(a(1,0), a(0)) = 1, \ (\alpha_i[j] = \beta_i[j] = 0)\\
&\textsc{TopDown}(a(1,0), a(1)) = 1, \ (\alpha_i[j] = 0, \beta_i[j] = 1)\\
&\textsc{TopDown}(a(0,0), a(0)) = 1, \ (\alpha_i[j] = 1, \beta_i[j] = 0)\\
&\textsc{TopDown}(a(0,0), a(1)) = 3. \ (\alpha_i[j] = \beta_i[j] = 1)
\end{aligned}$$

これより, 以下の式が成り立つ.

$$\textsc{TopDown}(A_i^j, B_i^j) = \begin{cases} 3 & \alpha_i[j] = \beta_i[j] = 1, \\ 1 & \text{それ以外のとき}. \end{cases}$$

$\textsc{TopDown}(A_i, B_i) = \sum_{j=1}^{D} \textsc{TopDown}(A_i^j, B_i^j)$ なので, $\alpha_i \cdot \beta_i = 0$ ならば $\textsc{TopDown}(A_i, B_i) = D$ となり, そうでなければ $\textsc{TopDown}(A_i, B_i) \ge D + 2$ となる.

さらに, C と B_i の最小コストのトップダウンマッピングは, 根から葉まで $D + 2$ 個の組 (a, a) を要素とし, そのコストは 0 である. また, M は, B_i の部分木 $a(a(a))$ に含まれる a の一つを要素として含まず, そのコストは 1 である. $\textsc{TopDown}(a(0, 1), a(0)) = \textsc{TopDown}(a(0, 1), a(1)) = 1$ より $\textsc{TopDown}(C^j, B_i^j) = 1$ となる. ゆえに, $\textsc{TopDown}(C, B_i) = \sum_{j=1}^{D} \textsc{TopDown}(C^j, B_i^j) + 1 = D + 1$ となる. □

A_i と B_i に対して, 木 T_i と S_i をそれぞれ $a(A_i, C)$ と $a(B_i)$ と構成する.

補題 3 $\alpha_i \cdot \beta_i = 0$ ならば $\textsc{TopDown}(T_i, S_i) = 5D + 2$ であり, そうでなければ $\textsc{TopDown}(T_i, S_i) = 5D + 4$ である.

証明. M を T_i と S_i の最小コストのトップダウンマッピングとする. このとき, M は A_i と C のどちらかのノードと B_i のノードを対応付ける.

$\alpha_i \cdot \beta_i = 0$ と仮定する. M が A_i と B_i のノードの組を含むならば, 補題 1 と 2 より, $\textsc{TopDown}(T_i, S_i) = \textsc{TopDown}(A_i, B_i) + |C| = D + 4D + 2 = 5D + 2$ となる. M が C と B_i のノードの組を含むならば, 補題 1 と 2 より, $\textsc{TopDown}(T_i, S_i) = \textsc{TopDown}(C, B_i) + |A_i| = D + 1 + 4D + 3 = 5D + 4$ となる. ゆえに, M は常に A_i と B_i のノードの組を含み, したがって, $\textsc{TopDown}(T_i, S_i) = 5D + 2$ が成り立つ.

$\alpha_i \cdot \beta_i = 1$ と仮定する．M が C と B_i のノードの対を含むならば, 補題 1 と 2 より, $\textsc{TopDown}(T_i, S_i) \geq \textsc{TopDown}(A_i, B_i) + |C| = D+2+4D+2 = 5D+4$ となる．M が A_i と B_i のノードの組を含むならば, 補題 1 と 2 より, $\textsc{TopDown}(T_i, S_i) = \textsc{TopDown}(C, B_i) + |A_i| = D + 1 + 4D + 3 = 5D + 4$ となる．ゆえに, M は常に C と B_i のノードの組を含み，したがって, $\textsc{TopDown}(T_i, S_i) = 5D + 4$ が成り立つ． □

最後に, 木 T と S を以下のように構成する．

$$T = a(T_1, a(T_2, \ldots, a(T_N, \underbrace{a(\cdots a(a)}_{D+3})))),$$
$$S = a(S_1, a(S_2, \ldots, a(S_N, \underbrace{a(\cdots a(a)}_{D+3})))).$$

補題 4 $\alpha \cdot \beta = 0$ となる対 $(\alpha, \beta) \in A \times B$ が存在するならば $\textsc{TopDown}(T, S) \leq 5DN + 4N - 2$ となり, 存在しないならば $\textsc{TopDown}(T, S) = 5DN + 4N$ となる．

証明. M を T と S の最小コストのトップダウンマッピングとする．このとき, M は根から葉までの $N + D + 3$ 個の組 (a, a) を要素とし, そのコストは 0 である．また, M は任意の $i(1 \leq i \leq N)$ に対して T_i と S_i を対応付ける．$\alpha \cdot \beta = 0$ となる対 $(\alpha, \beta) \in A \times B$ が存在するならば, 補題 3 より, $\textsc{TopDown}(T, S) \leq 5D+2+(N-1)(5D+4) = 5DN + 4N - 2$ となる．存在しないならば, 補題 3 より, $\textsc{TopDown}(T, S) = N(5D + 4) = 5DN + 4N$ となる． □

□

定理 3 の証明で単一インデルコスト関数を利用しているので, 共通部分木問題に対して以下の系が成り立つ．

系 1 定数 $\varepsilon > 0$ に対して, 木 T と S がたとえ二分木としても, 最大共通トップダウン誘導部分木は SETH の下で $O(n^{2-\varepsilon})$ 時間では計算不可能である．

おわりに

本論文では, (1) 高さ 1 の木, もしくは, (2) 二分木のトップダウン距離は, SETH の下で, 定数 $\varepsilon > 0$ に対して $O(n^{2-\varepsilon})$ 時間では計算不可能 (SETH 困難) であることを証明した．また, (2) の系として, 二分木の最大共通トップダウン誘導部分木の計算も SETH 困難であることが言える．

1 節で述べたように, トップダウン距離は, 編集距離の最も制限された距離の一つである．別の最も制限された距離は, **ボトムアップ LCA 保存距離**(bottom-up LCA-preserving distance) である．これは, ボトムアップマッピング [5, 6, 9] と LCA 保存マッピング [12] の共通部分であるボトムアップ LCA 保存マッピングの最小コストとなり, このマッピングは共通完全部分木 (共通ボトムアップ誘導部分木) と対応する．Abboud ら [1] は, 無順序二分木に対する最大共通完全部分木問題の計算が SETH 困難であることを証明した．この結果が順序木でも成り立つか否かについて議論することは今後の課題である．

参考文献

[1] A. Abboud, A. Backurs, T. D. Hansen, V. V. Williams, O. Zamir: *Subtree isomorphism revisited*, Proc. SODA'16, 1256–1271, 2016.

[2] A. Backurs, P. Indyk: *Edit distance cannot be computed in strongly subquadratic time (unless SETH is false)*, Proc. STOC'15, 51–58, 2015.

[3] S. S. Chawathe: *Comparing hierarchical data in external memory*, Proc. VLDB'99, 90–101, 1999.

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

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

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

[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] R. Williams: *A new algorithm for optimal 2-constraint satisfaction and its implication*, Theoret. Comput. Sci. **348**, 357–365, 2005.

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

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