Mapping Kernels Between Rooted Labeled Trees Beyond Ordered Trees

Kouichi Hirata[1(✉)], Tetsuji Kuboyama[2], and Takuya Yoshino[1]

[1] Kyushu Institute of Technology, Kawazu 680-4, Iizuka 820-8502, Japan
{hirata,yoshino}@dumbo.ai.kyutech.ac.jp
[2] Gakushuin University, Mejiro 1-5-1, Toshima, Tokyo 171-8588, Japan
ori-gaba2014@tk.cc.gakushuin.ac.jp

Abstract. In this paper, we investigate several mapping kernels to count all of the mappings between two rooted labeled trees beyond ordered trees, that is, *cyclically ordered trees* such as *biordered trees*, *cyclic-ordered trees* and *cyclic-biordered trees*, and *degree-bounded unordered trees*. Then, we design the algorithms to compute a *top-down mapping kernel*, an *LCA-preserving segmental mapping kernel*, an *LCA-preserving mapping kernel*, an *accordant mapping kernel* and an *isolated-subtree mapping kernel* for biordered trees in $O(nm)$ time and ones for cyclic-ordered and cyclic-biordered trees in $O(nmdD)$ time, where n is the number of nodes in a tree, m is the number of nodes in another tree, D is the maximum value of the degrees in two trees and d is the minimum value of the degrees in two trees. Also we design the algorithms to compute the above kernels for degree-bounded unordered trees in $O(nm)$ time. On the other hand, we show that the problem of computing *label-preserving leaf-extended* top-down mapping kernel and *label-preserving* bottom-up mapping kernel is #P-complete.

1 Introduction

A *tree kernel* is one of the fundamental method to classify *rooted labeled trees* (*trees*, for short) through support vector machines (SVMs). Many researches to design tree kernels for *ordered* trees, in which an order among siblings is fixed, have been developed (*cf.*, [2,6,14–17]). We call them *ordered tree kernels*.

A *mapping kernel* [15–17] is a powerful and general framework for tree kernels based on counting all of the *mappings* (and their variations) as the set of one-to-one node correspondences [18]. It is known that the minimum cost of (Tai) mappings coincides with an edit distance between trees. Also, as the properties of mapping kernels, almost ordered tree kernels are classified into the framework of mapping kernels [15], and a mapping kernel is *positive definite* if and only if the mapping is *transitive*, that is, *closed under the composition* [16,17].

This work is partially supported by Grant-in-Aid for Scientific Research 24240021, 24300060, 25540137, 26280085 and 26370281 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

T. Murata et al. (Eds.): JSAI-isAI 2014 Workshops, LNAI 9067, pp. 317–330, 2015.
DOI: 10.1007/978-3-662-48119-6_24

On the other hand, few researches to design tree kernels for *unordered* trees, in which an order among siblings is arbitrary, have been developed. We call them *unordered tree kernels*. One of the reasons is that the problem of counting all of the subtrees for unordered trees is #P-complete [6].

In order to avoid such difficulty, the unordered tree kernel have been developed as counting all of the specific substructures. For example, Kuboyama *et al.* [9] and Kimura *et al.* [7] have designed the unordered tree kernel counting all of the *bifoliate q-grams* and all of the *subpaths*, respectively.

As a tractable mapping kernel for unordered trees, Hamada *et al.* [3] have introduced an *agreement-subtree mapping kernel* for *phylogenetic trees* (leaf-labeled binary unordered trees). Also they have given a new proof of intractability of computing a mapping kernel for unordered trees, simpler than Kashima *et al.* [6], such that the problem of counting the number of leaves with the same labels in leaf-labeled tree is #P-complete, which is based on the problem of counting all of the matchings in a bipartite graph.

It is known that, by introducing several conditions to mappings, we deal with several variations of mappings and they form the hierarchy of mappings [5,8,21,23]. Every variation of mappings provides not only a variation of the edit distance as the minimum cost of all the mappings [5,8,22,23] but also a tree kernel as the number of all the mappings [8,10,15].

Note that the problem of computing the tractable variations of the edit distance between unordered trees such as a top-down distance [1,13], an LCA-preserving segmental distance [23], an LCA-preserving distance [27], an accordant distance [8,10,22] and an isolated-subtree distance [25,26] is essential to solve the minimum weighted maximum matching in a bipartite graph [22,26,27]. On the other hand, it is essential for the above #P-completeness [3,6] to reduce from the problem of counting all of the matchings in a bipartite graph.

Recently, as trees extended from ordered trees and restricted to unordered trees, Yoshino and Hirata [24] have introduced the following three kinds of a *cyclically ordered tree* that is an unordered tree preserving the adjacency among siblings in a tree as possible. Let $v_1, \dots, v_n$ be siblings from left to right. We say that a tree is *biordered* if it allows two orders $v_1, \dots, v_n$ and $v_n, \dots, v_1$. Also we say that a tree is *cyclic-ordered* if it allows a cyclic order $v_i, \dots, v_n, v_1, \dots, v_{i-1}$ for every i $(1 \leq i \leq n)$. Furthermore, we say that a tree is *cyclic-biordered* if it allows cyclic orders $v_i, \dots, v_n, v_1, \dots, v_{i-1}$ and $v_{i-1}, \dots, v_1, v_n, \dots, v_i$ for every i $(1 \leq i \leq n)$. Then, they have designed the algorithm to compute an *alignment distance* [4] between cyclically ordered trees in polynomial time. Note that the algorithm does not use the maximum matching for a bipartite graph. It is a simple extension of the algorithm (or recurrences) of computing the alignment distance between ordered trees [4].

Hence, in this paper, we first investigate several mapping kernels such as a *top-down mapping kernel*, an *LCA-preserving segmental mapping kernel*, an *LCA-preserving mapping kernel*, an *accordant mapping kernel* and an *isolated-subtree mapping kernel* for cyclically ordered trees. Then, we design the algorithms to compute all of the above mapping kernels for biordered trees in $O(nm)$

time and ones for cyclic-ordered and cyclic-biordered trees in $O(nmdD)$ time, where n is the number of nodes in a tree, m is the number of nodes in another tree, D is the maximum value of the degrees in two trees and d is the minimum value of the degrees in two trees.

Next, by focusing that the agreement subtree mapping kernel is applied to full binary trees, we investigate the above kernels for bounded-degree unordered trees. Then, we design the algorithms to compute all of the above mapping kernels in $O(nm)$ time, which follows from the algorithms to compute ones for unordered trees in $O(nmD^D)$ time, which is exponential to D.

On the other hand, for unordered trees, we show that the problem of computing the *label-preserving leaf-extended* top-down mapping kernel and the *label-preserving* bottom-up mapping kernel is #P-complete. Note here that the proof of the above #P-completeness [3,6] cannot apply to top-down and bottom-up mapping kernels for unordered tree directly. Also, the degrees of unordered trees in this proof are not bounded.

2 Preliminaries

A *tree* is a connected graph without cycles. For a tree $T = (V, E)$, we denote V and E by $V(T)$ and $E(T)$, respectively. Also the *size* of T is $|V|$ and denoted by $|T|$. We sometime denote $v \in V(T)$ by $v \in T$. We denote an empty tree by $\emptyset$.

A *rooted tree* is a tree with one node r chosen as its *root*. We denote the root of a rooted tree T by $r(T)$. A(n ordered) *forest* is a sequence $[T_1, \ldots, T_n]$ of trees which we denote by $T_1 \bullet \cdots \bullet T_n$ or $\bullet_{i=1}^{n} T_i$. In particular, for two forests $F_1 = T_1 \bullet \cdots \bullet T_n$ and $F_2 = S_1 \bullet \cdots \bullet S_m$, we denote the forest $T_1 \bullet \cdots \bullet T_n \bullet S_1 \bullet \cdots \bullet S_m$ by $F_1 \bullet F_2$. For a forest F, we denote the tree rooted by v whose children are trees in F by $v(F)$.

For each node v in a rooted tree with the root r, let $UP_r(v)$ be the unique path (as trees) from v to r. The *parent* of $v (\neq r)$, which we denote by $par(v)$, is its adjacent node on $UP_r(v)$ and the *ancestors* of $v (\neq r)$ are the nodes on $UP_r(v) - \{v\}$. We denote the set of all ancestors of v by $anc(v)$. We say that u is a *child* of v if v is the parent of u. The set of children of v is denoted by $ch(v)$. A *leaf* is a node having no children. We denote the set of all leaves in T by $lv(T)$. A node that is neither a leaf nor a root is called an *internal node*. We call the number of children of v the *degree* of v and denote it by $d(v)$, that is, $d(v) = |ch(v)|$. Also we define $d(T) = \max\{d(v) \mid v \in T\}$ and call it the *degree* of T.

In this paper, we use the ancestor orders $<$ and $\leq$, that is, $u < v$ if v is an ancestor of u and $u \leq v$ if $u < v$ or $u = v$. We say that w is the *least common ancestor* (LCA for short) of u and v, denoted by $u \sqcup v$, if $u \leq w$, $v \leq w$ and there exists no w' such that A (*complete*) $w' < w$, $u \leq w'$ and $v \leq w'$. A (*complete*) *subtree of* $T = (V, E)$ *rooted by* v, denoted by $T[v]$, is a tree $T' = (V', E')$ such that $r(T') = v$, $V' = \{u \in V \mid u \leq v\}$ and $E' = \{(u, w) \in E \mid u, w \in V'\}$.

We say that a rooted tree is *labeled* if each node is assigned a symbol from a fixed finite alphabet Σ. For a node v, we denote the label of v by $l(v)$, and sometimes identify v with $l(v)$. Also let $\varepsilon \notin \Sigma$ denote a special *blank* symbol and define $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$.

Let $v \in T$ and $v_i, v_j \in ch(v)$ such that v_i the i-th child of v and v_j the j-th child of v. Then, we say that v_i is *to the left of* v_j if $i \leq j$. Then, for every $u, v \in T$, $u \preceq v$ if either u is to the left of v (when both u and v are the children of the same node in T) or there exist $u', v' \in ch(u \sqcup v)$ such that $u \leq u'$, $v \leq v'$ and u' is to the left of v'. Hence, we say that a rooted tree is *ordered* if a left-to-right order among siblings is fixed; *unordered* otherwise. Furthermore, in this paper, we introduce *cyclically ordered trees* by using the following functions $\sigma^+_{p,n}(i)$ and $\sigma^-_{p,n}(i)$ for $1 \leq i, p \leq n$.

$$\sigma^+_{p,n}(i) = ((i + p - 1) \bmod n) + 1, \;\; \sigma^-_{p,n}(i) = ((n - i - p + 1) \bmod n) + 1.$$

Definition 1 (Cyclically Ordered Trees). Let T be a tree and suppose that $v_1, \ldots, v_n$ are the children of $v \in T$ from left to right.

1. We say that T is *biordered* if T allows the orders of both $v_1, \ldots, v_n$ and $v_n, \ldots, v_1$.
2. We say that T is *cyclic-ordered* if T allows the orders $v_{\sigma^+_{p,n}(1)}, \ldots, v_{\sigma^+_{p,n}(n)}$ for every $1 \leq p \leq n$.
3. We say that T is *cyclic-biordered* if T allows the orders $v_{\sigma^+_{p,n}(1)}, \ldots, v_{\sigma^+_{p,n}(n)}$ and $v_{\sigma^-_{p,n}(1)}, \ldots, v_{\sigma^-_{p,n}(n)}$ for every $1 \leq p \leq n$.

Sometimes we use the scripts o, b, c, cb, u, and the notation of $\pi \in \{o, b, c, cb, u\}$.

It is obvious that the cyclically ordered trees are an extension of ordered trees and a restriction of unordered trees. The number of orders among siblings of a node v in ordered trees, biordered trees, cyclic-ordered trees, cyclic-biordered trees and unordered trees is 1, 2, $d(v)$, $2d(v)$ and $d(v)!$, respectively. Also it holds that, when $d(T) = 2$, T is unordered iff it is biordered, cyclic-ordered or cyclic-biordered, and when $d(T) = 3$, T is unordered iff it is cyclic-biordered.

3 Mapping

In this section, we introduce a *Tai mapping* and its variations, and then the distance as the minimum cost of all the mappings.

Definition 2 (Tai Mapping [18]). Let T_1 and T_2 be trees and $M \subseteq V(T_1) \times V(T_2)$.

1. We say that a triple (M, T_1, T_2) is an *ordered Tai mapping* from T_1 to T_2, denoted by $M \in \mathcal{M}^o_{\mathrm{TAI}}(T_1, T_2)$, if every pair (u_1, v_1) and (u_2, v_2) in M satisfies the following conditions.
   (i) $u_1 = u_2$ iff $v_1 = v_2$ (one-to-one condition).
   (ii) $u_1 \leq u_2$ iff $v_1 \leq v_2$ (ancestor condition).
   (iii) $u_1 \preceq u_2$ iff $v_1 \preceq v_2$ (sibling condition).

2. We say that a triple (M, T_1, T_2) is an *unordered Tai mapping* from T_1 to T_2, denoted by $M \in \mathcal{M}^{u}_{\text{TAI}}(T_1, T_2)$, if M satisfies the conditions (i) and (ii).

In the following, let $u_1, u_2, u_3, u_4 \in ch(u)$ and $v_1, v_2, v_3, v_4 \in ch(v)$.

3. We say that a triple (M, T_1, T_2) is a *biordered Tai mapping* from T_1 to T_2, denoted by $M \in \mathcal{M}^{b}_{\text{TAI}}(T_1, T_2)$, if M satisfies the above conditions (i) and (ii) and the following condition (iv).
   (iv) For every $u \in T_1$ and $v \in T_2$ such that $(u_1, v_1), (u_2, v_2), (u_3, v_3) \in M$, one of the following statements holds.
   1. $u_1 \preceq u_2 \preceq u_3$ iff $v_1 \preceq v_2 \preceq v_3$.
   2. $u_1 \preceq u_2 \preceq u_3$ iff $v_3 \preceq v_2 \preceq v_1$.
4. We say that a triple (M, T_1, T_2) is a *cyclic-ordered Tai mapping* from T_1 to T_2, denoted by $M \in \mathcal{M}^{c}_{\text{TAI}}(T_1, T_2)$, if M satisfies the above conditions (i) and (ii) and the following condition (v).
   (v) For every $u \in T_1$ and $v \in T_2$ such that $(u_1, v_1), (u_2, v_2), (u_3, v_3) \in M$, one of the following statements holds.
   1. $u_1 \preceq u_2 \preceq u_3$ iff $v_1 \preceq v_2 \preceq v_3$.
   2. $u_1 \preceq u_2 \preceq u_3$ iff $v_2 \preceq v_3 \preceq v_1$.
   3. $u_1 \preceq u_2 \preceq u_3$ iff $v_3 \preceq v_1 \preceq v_2$.
5. We say that a triple (M, T_1, T_2) is a *cyclic-biordered Tai mapping* from T_1 to T_2, denoted by $M \in \mathcal{M}^{cb}_{\text{TAI}}(T_1, T_2)$, if M satisfies the above conditions (i) and (ii) and the following condition (vi).
   (vi) For every $u \in T_1$ and $v \in T_2$ such that $(u_1, v_1), (u_2, v_2), (u_3, v_3), (u_4, v_4) \in M$, one of the following statements holds.
   1. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_1 \preceq v_2 \preceq v_3 \preceq v_4$.
   2. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_2 \preceq v_3 \preceq v_4 \preceq v_1$.
   3. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_3 \preceq v_4 \preceq v_1 \preceq v_2$.
   4. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_4 \preceq v_1 \preceq v_2 \preceq v_3$.
   5. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_4 \preceq v_3 \preceq v_2 \preceq v_1$.
   6. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_3 \preceq v_2 \preceq v_1 \preceq v_4$.
   7. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_2 \preceq v_1 \preceq v_4 \preceq v_3$.
   8. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_1 \preceq v_4 \preceq v_3 \preceq v_2$.

We will use M instead of (M, T_1, T_2) simply and call a Tai mapping a *mapping* simply.

Definition 3 (Variations of Tai Mapping). Let T_1 and T_2 be trees, $\pi \in \{o, b, c, cb, u\}$ and $M \in \mathcal{M}^{\pi}_{\text{TAI}}(T_1, T_2)$. Here, we denote $M - \{(r(T_1), r(T_2))\}$ by M^-.

1. We say that M is a *top-down mapping* [1,13] (or a *degree*-1 *mapping*), denoted by $M \in \mathcal{M}^{\pi}_{\text{TOP}}(T_1, T_2)$, if M satisfies the following condition.

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

2. We say that M is an *LCA-preserving segmental mapping* [23], denoted by $M \in \mathcal{M}^{\pi}_{\text{LCASG}}(T_1, T_2)$, if there exists a pair $(u, v) \in T_1 \times T_2$ such that $M \in \mathcal{M}^{\pi}_{\text{TOP}}(T_1[u], T_2[v])$.

3. We say that M is an *LCA-preserving mapping* (or a *degree*-2 *mapping* [27]), denoted by $M \in \mathcal{M}^{\pi}_{\text{LCA}}(T_1, T_2)$, if M satisfies the following condition.
$$\forall (u_1, v_1), (u_2, v_2) \in M \Big((u_1 \sqcup u_2, v_1 \sqcup v_2) \in M \Big).$$
4. We say that M is an *accordant mapping* [8] (or a *Lu's mapping* [12]), denoted by $M \in \mathcal{M}^{\pi}_{\text{ACC}}(T_1, T_2)$, if M satisfies the following condition.
$$\forall (u_1, v_1), (u_2, v_2), (u_3, v_3) \in M \Big(u_1 \sqcup u_2 = u_1 \sqcup u_3 \iff v_1 \sqcup v_2 = v_1 \sqcup v_3 \Big).$$
5. We say that M is an *isolated-subtree mapping* [21] (or a *constrained mapping* [25,26]), denoted by $M \in \mathcal{M}^{\pi}_{\text{ILST}}(T_1, T_2)$, if M satisfies the following condition.
$$\forall (u_1, v_1), (u_2, v_2), (u_3, v_3) \in M \Big(u_3 < u_1 \sqcup u_2 \iff v_3 < v_1 \sqcup v_2 \Big).$$
6. We say that M is a *bottom-up mapping* [8,20,22], denoted by $M \in \mathcal{M}^{\pi}_{\text{BOT}}(T_1, T_2)$, if M satisfies the following condition.
$$\forall (u, v) \in M \; \begin{pmatrix} \forall u' \in T_1[u] \exists v' \in T_2[v] \Big((u', v') \in M \Big) \\ \wedge \forall v' \in T_2[v] \exists u' \in T_1[u] \Big((u', v') \in M \Big) \end{pmatrix}.$$

Proposition 1 (*cf.* [8,23]**).** *For $\pi \in \{o, b, c, cb, u\}$ and trees T_1 and T_2, the following statement holds:*

$$\begin{aligned} \mathcal{M}^{\pi}_{\text{TOP}}(T_1, T_2) \subset \mathcal{M}^{\pi}_{\text{LCASG}}(T_1, T_2) \subset \mathcal{M}^{\pi}_{\text{LCA}}(T_1, T_2) \\ \subset \mathcal{M}^{\pi}_{\text{ACC}}(T_1, T_2) \subset \mathcal{M}^{\pi}_{\text{ILST}}(T_1, T_2). \end{aligned}$$

Furthermore, for $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}\}$, $\mathcal{M}^{\pi}_{\text{BOT}}(T_1, T_2)$ is incomparable with $\mathcal{M}^{\pi}_{\mathtt{A}}(T_1, T_2)$

4 Mapping Kernels

Let $\pi \in \{o, b, c, cb, u\}$ and $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}\}$ unless otherwise noted. A *mapping* between forests F_1 and F_2 is defined as a mapping M between trees $v(F_1)$ and $v(F_2)$ such that $(v, v) \notin M$. We define $\mathcal{M}^{\pi}_{\mathtt{A}}(F_1, F_2)$ as similar as $\mathcal{M}^{\pi}_{\mathtt{A}}(T_1, T_2)$. Let $\sigma : \Sigma \times \Sigma \to \mathbf{R}^+$ be a similarity function. The similarity $\sigma(M)$ of a mapping $M \in \mathcal{M}^{\pi}_{\mathtt{A}}(T_1, T_2)$ between two trees T_1 and T_2 is defined as $\sigma(M) = \prod_{(u,v) \in M} \sigma(l(u), l(v))$. The *similarity* between two forests F_1 and F_2 is defined as follows:

$$\mathcal{K}^{\pi}_{\mathtt{A}}(F_1, F_2) = \sum_{M \in \mathcal{M}^{\pi}_{\mathtt{A}}(F_1, F_2)} \sigma(M).$$

Corollary 1. *For* $\pi \in \{o, b, c, cb, u\}$ *and* $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}\}$, $\mathcal{K}^{\pi}_{\mathtt{A}}$ *is positive definite.*

Proof. Since $\mathcal{M}^{\pi}_{\mathtt{A}}$ is closed under the composition [8,23,27] and by [16], the statement holds. $\square$

Kuboyama [8] has introduced the recurrences to compute $\mathcal{K}^{o}_{\mathtt{A}}(T_1, T_2)$ for $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}\}$ implicitly and $\mathtt{A} \in \{\textsc{Acc}, \textsc{Ilst}\}$ explicitly illustrated in Fig. 1. Note the underlined formulas that denote the difference between similar formulas.

$$
\begin{aligned}
\mathcal{K}^o_{\textsc{Top}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot (1 + \mathcal{F}^o_{\textsc{Top}}(F_1, F_2)),\\
\mathcal{F}^o_{\textsc{Top}}(\emptyset, F) &= \mathcal{F}^o_{\textsc{Top}}(F, \emptyset) = 0,\\
\mathcal{F}^o_{\textsc{Top}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \underline{\mathcal{K}^o_{\textsc{Top}}(T_1, T_2)} \cdot (1 + \mathcal{F}^o_{\textsc{Top}}(F_1, F_2))\\
&\quad + \mathcal{F}^o_{\textsc{Top}}(F_1, T_2 \bullet F_2) + \mathcal{F}^o_{\textsc{Top}}(T_1 \bullet F_1, F_2) - \mathcal{F}^o_{\textsc{Top}}(F_1, F_2).
\end{aligned}
$$

$$
\mathcal{K}^o_{\textsc{LcaSg}}(T_1, T_2) = \sum_{u \in T_1} \sum_{v \in T_2} \mathcal{K}^o_{\textsc{Top}}(T_1[u], T_2[v]).
$$

$$
\begin{aligned}
\mathcal{K}^o_{\textsc{Lca}}(T_1, T_2) &= \sum_{u \in T_1} \sum_{v \in T_2} \mathcal{T}^o_{\textsc{Lca}}(T_1[u], T_2[v]),\\
\mathcal{T}^o_{\textsc{Lca}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot (1 + \mathcal{F}^o_{\textsc{Lca}}(F_1, F_2)),\\
\mathcal{F}^o_{\textsc{Lca}}(\emptyset, F) &= \mathcal{F}^o_{\textsc{Lca}}(F, \emptyset) = 0,\\
\mathcal{F}^o_{\textsc{Lca}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \underline{\mathcal{K}^o_{\textsc{Lca}}(T_1, T_2)} \cdot (1 + \mathcal{F}^o_{\textsc{Lca}}(F_1, F_2))\\
&\quad + \mathcal{F}^o_{\textsc{Lca}}(F_1, T_2 \bullet F_2) + \mathcal{F}^o_{\textsc{Lca}}(T_1 \bullet F_1, F_2) - \mathcal{F}^o_{\textsc{Lca}}(F_1, F_2).
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{K}^o_{\textsc{Acc}}(\emptyset, F) &= \mathcal{K}^o_{\textsc{Acc}}(F, \emptyset) = \mathcal{TF}^o_{\textsc{Acc}}(T, \emptyset) = 0,\\
\mathcal{K}^o_{\textsc{Acc}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^o_{\textsc{Acc}}(T_1, T_2) \cdot (\mathcal{F}^o_{\textsc{Acc}}(F_1, F_2) - 1)\\
&\quad + \mathcal{TF}^o_{\textsc{Acc}}(T_1, T_2 \bullet F_2) - \mathcal{TF}^o_{\textsc{Acc}}(T_1, F_2)\\
&\quad + \mathcal{TF}^o_{\textsc{Acc}}(T_2, T_1 \bullet F_1) - \mathcal{TF}^o_{\textsc{Acc}}(T_2, F_1)\\
&\quad + \mathcal{K}^o_{\textsc{Acc}}(T_1 \bullet F_1, F_2) - \mathcal{K}^o_{\textsc{Acc}}(F_1, F_2),\\
\mathcal{TF}^o_{\textsc{Acc}}(v(F_1), T_2 \bullet F_2) &= \mathcal{T}^o_{\textsc{Acc}}(v(F_1), T_2) - \mathcal{TF}^o_{\textsc{Acc}}(T_2, F_1) + \mathcal{TF}^o_{\textsc{Acc}}(v(F_1), F_2)\\
&\quad - \mathcal{K}^o_{\textsc{Acc}}(F_1, F_2) + \mathcal{K}^o_{\textsc{Acc}}(F_1, T_1 \bullet F_2),\\
\mathcal{T}^o_{\textsc{Acc}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot (1 + \underline{\mathcal{F}^o_{\textsc{Acc}}(F_1, F_2)}) + \mathcal{TF}^o_{\textsc{Acc}}(u(F_1), F_2)\\
&\quad + \mathcal{TF}^o_{\textsc{Acc}}(v(F_2), F_1) - \mathcal{K}^o_{\textsc{Acc}}(F_1, F_2),\\
\mathcal{F}^o_{\textsc{Acc}}(\emptyset, F) &= \mathcal{F}^o_{\textsc{Acc}}(F, \emptyset) = 0,\\
\mathcal{F}^o_{\textsc{Acc}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^o_{\textsc{Acc}}(T_1, T_2) \cdot (1 + \mathcal{F}^o_{\textsc{Acc}}(F_1, F_2))\\
&\quad + \mathcal{F}^o_{\textsc{Acc}}(F_1, T_2 \bullet F_2) + \mathcal{F}^o_{\textsc{Acc}}(T_1 \bullet F_1, F_2) - \mathcal{F}^o_{\textsc{Acc}}(F_1, F_2).
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{K}^o_{\textsc{Ilst}}(\emptyset, F) &= \mathcal{K}^o_{\textsc{Ilst}}(F, \emptyset) = \mathcal{TF}^o_{\textsc{Ilst}}(T, \emptyset) = 0,\\
\mathcal{K}^o_{\textsc{Ilst}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^o_{\textsc{Ilst}}(T_1, T_2) \cdot (\mathcal{F}^o_{\textsc{Ilst}}(F_1, F_2) - 1)\\
&\quad + \mathcal{TF}^o_{\textsc{Ilst}}(T_1, T_2 \bullet F_2) - \mathcal{TF}^o_{\textsc{Ilst}}(T_1, F_2)\\
&\quad + \mathcal{TF}^o_{\textsc{Ilst}}(T_2, T_1 \bullet F_1) - \mathcal{TF}^o_{\textsc{Ilst}}(T_2, F_1)\\
&\quad + \mathcal{K}^o_{\textsc{Ilst}}(T_1 \bullet F_1, F_2) - \mathcal{K}^o_{\textsc{Ilst}}(F_1, F_2),\\
\mathcal{TF}^o_{\textsc{Ilst}}(v(F_1), T_2 \bullet F_2) &= \mathcal{T}^o_{\textsc{Ilst}}(v(F_1), T_2) - \mathcal{TF}^o_{\textsc{Ilst}}(T_2, F_1) + \mathcal{TF}^o_{\textsc{Ilst}}(v(F_1), F_2)\\
&\quad - \mathcal{K}^o_{\textsc{Ilst}}(F_1, F_2) + \mathcal{K}^o_{\textsc{Ilst}}(F_1, T_1 \bullet F_2),\\
\mathcal{T}^o_{\textsc{Ilst}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot (1 + \underline{\mathcal{K}^o_{\textsc{Ilst}}(F_1, F_2)}) + \mathcal{TF}^o_{\textsc{Ilst}}(u(F_1), F_2)\\
&\quad + \mathcal{TF}^o_{\textsc{Ilst}}(v(F_2), F_1) - \mathcal{K}^o_{\textsc{Ilst}}(F_1, F_2),\\
\mathcal{F}^o_{\textsc{Ilst}}(\emptyset, F) &= \mathcal{F}^o_{\textsc{Ilst}}(F, \emptyset) = 0,\\
\mathcal{F}^o_{\textsc{Ilst}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^o_{\textsc{Ilst}}(T_1, T_2) \cdot (1 + \mathcal{F}^o_{\textsc{Ilst}}(F_1, F_2))\\
&\quad + \mathcal{F}^o_{\textsc{Ilst}}(F_1, T_2 \bullet F_2) + \mathcal{F}^o_{\textsc{Ilst}}(T_1 \bullet F_1, F_2) - \mathcal{F}^o_{\textsc{Ilst}}(F_1, F_2).
\end{aligned}
$$

Fig. 1. The recurrences of computing $\mathcal{K}^{o}_{\mathtt{A}}(T_1, T_2)$ for $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}\}$ [8].

Theorem 1 (*cf.*, Kuboyama [8]). *For* $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}\}$, *the recurrences in Fig. 1 correctly compute* $\mathcal{K}^o_{\mathtt{A}}(T_1, T_2)$ *in* $O(nm)$ *time, where* $n = |T_1|$ *and* $m = |T_2|$.

4.1 Mapping Kernels for Cyclically Ordered Trees

In this section, we extend the recurrences in Fig. 1 to the recurrences to compute $\mathcal{K}^\pi_{\mathtt{A}}(T_1, T_2)$ for $\pi \in \{o, b, c, cb\}$ and $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}\}$.

For $u(F_1)$ and $v(F_2)$, let $F_1{=}[T_1[u_1], \ldots, T_1[u_s]]$ and $F_2{=}[T_2[v_1], \ldots, T_2[v_t]]$, that is, $ch(u) = \{u_1, \ldots, u_s\}$, $ch(v) = \{v_1, \ldots, v_t\}$, $d(u) = s$ and $d(v) = t$. Also let $1 \leq p \leq s$ and $1 \leq q \leq t$. We denote the forests $[T_1[u_{\sigma^+_{p,s}(1)}], \ldots, T_1[u_{\sigma^+_{p,s}(s)}]]$ and $[T_2[v_{\sigma^+_{q,t}(1)}], \ldots, T_2[v_{\sigma^+_{q,t}(t)}]]$ by F_1^p and F_2^q. Furthermore, we denote the forests $[T_1[u_{\sigma^-_{p,s}(1)}], \ldots, T_1[u_{\sigma^-_{p,s}(s)}]]$ and $[T_2[v_{\sigma^-_{q,t}(1)}], \ldots, T_2[v_{\sigma^-_{q,t}(t)}]]$ by F_1^{-p} and F_2^{-q}. It is obvious that $F_1 = F_1^1$ and $F_2 = F_2^1$.

Furthermore, the values of p and q in F_1^p, and F_2^q are (1) $p = q = 1$ if $\pi = o$, (2) $p = \pm 1$ and $q = \pm 1$ if $\pi = b$, (3) $1 \leq p \leq s$ and $1 \leq q \leq t$ if $\pi = c$ and (4) $1 \leq p \leq s$, $-s \leq p \leq -1$, $1 \leq q \leq t$ and $-t \leq q \leq -1$ if $\pi = cb$. Hence, we prepare the following sets: (1) $o(s) = o(t) = \{1\}$, (2) $b(s) = b(t) = \{-1, 1\}$, (3) $c(s) = \{1, \ldots, s\}$, $c(t) = \{1, \ldots, t\}$, and (4) $cb(s) = \{-s, \ldots, -1, 1, \ldots, s\}$, $cb(t) = \{-t, \ldots, -1, 1, \ldots, t\}$. We refer these sets to $\pi(s)$ and $\pi(t)$ for $\pi \in \{o, b, c, cb\}$.

Then, we design the recurrences to compute $\mathcal{K}^\pi_{\mathtt{A}}(T_1, T_2)$ illustrated in Fig. 2.

Theorem 2. *For* $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}\}$*, the recurrences in Fig. 2 correctly compute* $\mathcal{K}^b_{\mathtt{A}}(T_1, T_2)$ *in* $O(nm)$ *time and* $\mathcal{K}^c_{\mathtt{A}}(T_1, T_2)$ *and* $\mathcal{K}^{cb}_{\mathtt{A}}(T_1, T_2)$ *in* $O(nmdD)$ *time, where* $n = |T_1|$, $m = |T_2|$, $d = \min\{d(T_1), d(T_2)\}$ *and* $D = \max\{d(T_1), d(T_2)\}$.

Proof. In the formulas of $\mathcal{K}^\pi_{\textsc{Top}}$ and $\mathcal{T}^\pi_{\textsc{Lca}}$, the number of $\mathcal{F}^\pi_{\textsc{Top}}(F_1^p, F_2^q)$ and $\mathcal{F}^\pi_{\textsc{Lca}}(F_1^p, F_2^q)$ is 1 if $\pi = o$, 4 if $\pi = b$, $d(u) \cdot d(v)$ if $\pi = c$ and $2d(u) \cdot 2d(v)$ if $\pi = bc$. Also in the formulas of $\mathcal{T}^\pi_{\textsc{Acc}}$ and $\mathcal{T}^\pi_{\textsc{Ilst}}$, the number of $\mathcal{F}^\pi_{\textsc{Acc}}(F_1^p, F_2^q)$ and $\mathcal{F}^\pi_{\textsc{Ilst}}(F_1^p, F_2^q)$ is 1 if $\pi = o$, $4 + 2 + 2 + 4 = 12$ if $\pi = b$, $d(u) \cdot d(v) + d(u) + d(v) + d(u) \cdot d(v) = 2d(u) \cdot d(v) + d(u) + d(v)$ if $\pi = c$ and $2d(u) \cdot 2d(v) + 2d(u) + 2d(v) + 2d(u) \cdot 2d(v) = 8d(u) \cdot d(v) + 2d(u) + 2d(v)$ if $\pi = bc$. Then, we can compute these recurrences in $O(1)$ time if $\pi \in \{o, b\}$, whereas in $O(d(u) \cdot d(v)) = O(dD)$ time if $\pi \in \{c, cb\}$. Hence, the time complexity in the statement holds. Also we can show the correctness by extending Theorem 1. $\square$

4.2 Mapping Kernels for Bounded-Degree Unordered Trees

In this section, we extend the recurrences in Fig. 1 to the recurrences to compute $\mathcal{K}^u_{\mathtt{A}}(T_1, T_2)$ for $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}\}$.

For nonnegative integers s and t, let $B_{s,t}$ be a complete bipartite graph $(X \cup Y, E)$ such that $X = \{1, \ldots, s\}$ and $Y = \{1, \ldots, t\}$, and $BM(s, t)$ the set of

$$
\begin{aligned}
\mathcal{K}^{\pi}_{\mathrm{TOP}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot \left(1 + \sum\nolimits_{p\in\pi(d(u))} \sum\nolimits_{q\in\pi(d(v))} \mathcal{F}^{\pi}_{\mathrm{TOP}}(F_1^p, F_2^q)\right),\\
\mathcal{F}^{\pi}_{\mathrm{TOP}}(\emptyset, F) &= \mathcal{F}^{\pi}_{\mathrm{TOP}}(F, \emptyset) = 0,\\
\mathcal{F}^{\pi}_{\mathrm{TOP}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{K}^{\pi}_{\mathrm{TOP}}(T_1, T_2) \cdot (1 + \mathcal{F}^{\pi}_{\mathrm{TOP}}(F_1, F_2))\\
&\quad + \mathcal{F}^{\pi}_{\mathrm{TOP}}(F_1, T_2 \bullet F_2) + \mathcal{F}^{\pi}_{\mathrm{TOP}}(T_1 \bullet F_1, F_2) - \mathcal{F}^{\pi}_{\mathrm{TOP}}(F_1, F_2).
\end{aligned}
$$

$$
\mathcal{K}^{\pi}_{\mathrm{LCASG}}(T_1, T_2) = \sum_{u\in T_1} \sum_{v\in T_2} \mathcal{K}^{\pi}_{\mathrm{TOP}}(T_1[u], T_2[v]).
$$

$$
\begin{aligned}
\mathcal{K}^{\pi}_{\mathrm{LCA}}(T_1, T_2) &= \sum_{u\in T_1} \sum_{v\in T_2} \mathcal{T}^{\pi}_{\mathrm{LCA}}(T_1[u], T_2[v]),\\
\mathcal{T}^{\pi}_{\mathrm{LCA}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot \left(1 + \sum\nolimits_{p\in\pi(d(u))} \sum\nolimits_{q\in\pi(d(v))} \mathcal{F}^{\pi}_{\mathrm{LCA}}(F_1^p, F_2^q)\right),\\
\mathcal{F}^{\pi}_{\mathrm{LCA}}(\emptyset, F) &= \mathcal{F}^{\pi}_{\mathrm{LCA}}(F, \emptyset) = 0,\\
\mathcal{F}^{\pi}_{\mathrm{LCA}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{K}^{\pi}_{\mathrm{LCA}}(T_1, T_2) \cdot (1 + \mathcal{F}^{\pi}_{\mathrm{LCA}}(F_1, F_2))\\
&\quad + \mathcal{F}^{\pi}_{\mathrm{LCA}}(F_1, T_2 \bullet F_2) + \mathcal{F}^{\pi}_{\mathrm{LCA}}(T_1 \bullet F_1, F_2) - \mathcal{F}^{\pi}_{\mathrm{LCA}}(F_1, F_2).
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{K}^{\pi}_{\mathrm{ACC}}(\emptyset, F) &= \mathcal{K}^{\pi}_{\mathrm{ACC}}(F, \emptyset) = \mathcal{TF}^{\pi}_{\mathrm{ACC}}(T, \emptyset) = 0,\\
\mathcal{K}^{\pi}_{\mathrm{ACC}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^{\pi}_{\mathrm{ACC}}(T_1, T_2) \cdot (\mathcal{F}^{\pi}_{\mathrm{ACC}}(F_1, F_2) - 1)\\
&\quad + \mathcal{TF}^{\pi}_{\mathrm{ACC}}(T_1, T_2 \bullet F_2) - \mathcal{TF}^{\pi}_{\mathrm{ACC}}(T_1, F_2)\\
&\quad + \mathcal{TF}^{\pi}_{\mathrm{ACC}}(T_2, T_1 \bullet F_1) - \mathcal{TF}^{\pi}_{\mathrm{ACC}}(T_2, F_1)\\
&\quad + \mathcal{K}^{\pi}_{\mathrm{ACC}}(T_1 \bullet F_1, F_2) - \mathcal{K}^{\pi}_{\mathrm{ACC}}(F_1, F_2),\\
\mathcal{TF}^{\pi}_{\mathrm{ACC}}(v(F_1), T_2 \bullet F_2) &= \mathcal{T}^{\pi}_{\mathrm{ACC}}(v(F_1), T_2) - \mathcal{TF}^{\pi}_{\mathrm{ACC}}(T_2, F_1) + \mathcal{TF}^{\pi}_{\mathrm{ACC}}(v(F_1), F_2)\\
&\quad - \mathcal{K}^{\pi}_{\mathrm{ACC}}(F_1, F_2) + \mathcal{K}^{\pi}_{\mathrm{ACC}}(F_1, T_1 \bullet F_2),\\
\mathcal{T}^{\pi}_{\mathrm{ACC}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot \left(1 + \sum\nolimits_{p\in\pi(d(u))} \sum\nolimits_{q\in\pi(d(v))} \mathcal{F}^{\pi}_{\mathrm{ACC}}(F_1^p, F_2^q)\right)\\
&\quad + \sum\nolimits_{q\in\pi(d(v))} \mathcal{TF}^{\pi}_{\mathrm{ACC}}(u(F_1), F_2^q) + \sum\nolimits_{p\in\pi(d(u))} \mathcal{TF}^{\pi}_{\mathrm{ACC}}(v(F_2), F_1^p)\\
&\quad - \sum\nolimits_{p\in\pi(d(u))} \sum\nolimits_{q\in\pi(d(v))} \mathcal{K}^{\pi}_{\mathrm{ACC}}(F_1^p, F_2^q),\\
\mathcal{F}^{\pi}_{\mathrm{ACC}}(\emptyset, F) &= \mathcal{F}^{\pi}_{\mathrm{ACC}}(F, \emptyset) = 0,\\
\mathcal{F}^{\pi}_{\mathrm{ACC}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^{\pi}_{\mathrm{ACC}}(T_1, T_2) \cdot (1 + \mathcal{F}^{\pi}_{\mathrm{ACC}}(F_1, F_2))\\
&\quad + \mathcal{F}^{\pi}_{\mathrm{ACC}}(F_1, T_2 \bullet F_2) + \mathcal{F}^{\pi}_{\mathrm{ACC}}(T_1 \bullet F_1, F_2) - \mathcal{F}^{\pi}_{\mathrm{ACC}}(F_1, F_2).
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{K}^{\pi}_{\mathrm{ILST}}(\emptyset, F) &= \mathcal{K}^{\pi}_{\mathrm{ILST}}(F, \emptyset) = \mathcal{TF}^{o}_{\mathrm{ILST}}(T, \emptyset) = 0,\\
\mathcal{K}^{\pi}_{\mathrm{ILST}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^{\pi}_{\mathrm{ILST}}(T_1, T_2) \cdot (\mathcal{F}^{\pi}_{\mathrm{ILST}}(F_1, F_2) - 1)\\
&\quad + \mathcal{TF}^{\pi}_{\mathrm{ILST}}(T_1, T_2 \bullet F_2) - \mathcal{TF}^{\pi}_{\mathrm{ILST}}(T_1, F_2)\\
&\quad + \mathcal{TF}^{\pi}_{\mathrm{ILST}}(T_2, T_1 \bullet F_1) - \mathcal{TF}^{\pi}_{\mathrm{ILST}}(T_2, F_1)\\
&\quad + \mathcal{K}^{\pi}_{\mathrm{ILST}}(T_1 \bullet F_1, F_2) - \mathcal{K}^{\pi}_{\mathrm{ILST}}(F_1, F_2),\\
\mathcal{TF}^{\pi}_{\mathrm{ILST}}(v(F_1), T_2 \bullet F_2) &= \mathcal{T}^{\pi}_{\mathrm{ILST}}(v(F_1), T_2) - \mathcal{TF}^{\pi}_{\mathrm{ILST}}(T_2, F_1) + \mathcal{TF}^{\pi}_{\mathrm{ILST}}(v(F_1), F_2)\\
&\quad - \mathcal{K}^{\pi}_{\mathrm{ILST}}(F_1, F_2) + \mathcal{K}^{\pi}_{\mathrm{ILST}}(F_1, T_1 \bullet F_2),\\
\mathcal{T}^{\pi}_{\mathrm{ILST}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot \left(1 + \sum\nolimits_{p\in\pi(d(u))} \sum\nolimits_{q\in\pi(d(v))} \mathcal{K}^{\pi}_{\mathrm{ILST}}(F_1^p, F_2^q)\right)\\
&\quad + \sum\nolimits_{q\in\pi(d(v))} \mathcal{TF}^{\pi}_{\mathrm{ILST}}(u(F_1), F_2^q) + \sum\nolimits_{p\in\pi(d(u))} \mathcal{TF}^{\pi}_{\mathrm{ILST}}(v(F_2), F_1^p)\\
&\quad - \sum\nolimits_{p\in\pi(d(u))} \sum\nolimits_{q\in\pi(d(v))} \mathcal{K}^{\pi}_{\mathrm{ILST}}(F_1^p, F_2^q),\\
\mathcal{F}^{\pi}_{\mathrm{ILST}}(\emptyset, F) &= \mathcal{F}^{\pi}_{\mathrm{ILST}}(F, \emptyset) = 0,\\
\mathcal{F}^{\pi}_{\mathrm{ILST}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^{\pi}_{\mathrm{ILST}}(T_1, T_2) \cdot (1 + \mathcal{F}^{\pi}_{\mathrm{ILST}}(F_1, F_2))\\
&\quad + \mathcal{F}^{\pi}_{\mathrm{ILST}}(F_1, T_2 \bullet F_2) + \mathcal{F}^{\pi}_{\mathrm{ILST}}(T_1 \bullet F_1, F_2) - \mathcal{F}^{\pi}_{\mathrm{ILST}}(F_1, F_2).
\end{aligned}
$$

Fig. 2. The recurrences of computing $\mathcal{K}^{\pi}_{\mathtt{A}}(T_1, T_2)$ for $\pi \in \{o, b, c, cb\}$ and $\mathtt{A} \in \{\mathrm{TOP}, \mathrm{LCASG}, \mathrm{LCA}, \mathrm{ACC}, \mathrm{ILST}\}$.

all maximum matchings in $B_{s,t}$. For every $M \in BM(s,t)$, it holds that $M \subset E$ and $|M| = \min\{s,t\}$.

For $u(F_1)$ and $v(F_2)$, let $F_1 = [T_1[u_1], \ldots, T_1[u_s]]$ and $F_2 = [T_2[v_1], \ldots, T_2[v_t]]$, that is, $ch(u) = \{u_1, \ldots, u_s\}$, $ch(v) = \{v_1, \ldots, v_t\}$, $d(u) = s$ and $d(v) = t$. Then, for $M \in BM(s,t)$, we denote the ordered forests $\bullet_{(i,j)\in M} T_1[u_i]$ and $\bullet_{(i,j)\in M} T_2[v_j]$ by F_1^M and F_2^M, where we assume that trees in a forest are ordered along the order of M. Furthermore, for an ordered forest F, let $pm(F)$ be the set of all

$$
\begin{aligned}
\mathcal{K}^u_{\text{Top}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot \left(1 + \sum\nolimits_{M \in BM(d(u),d(v))} \mathcal{F}^u_{\text{Top}}(F_1^M, F_2^M)\right), \\
\mathcal{F}^u_{\text{Top}}(\emptyset, F) &= \mathcal{F}^u_{\text{Top}}(F, \emptyset) = 0, \\
\mathcal{F}^u_{\text{Top}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{K}^u_{\text{Top}}(T_1, T_2) \cdot (1 + \mathcal{F}^u_{\text{Top}}(F_1, F_2)) \\
&\quad + \mathcal{F}^u_{\text{Top}}(F_1, T_2 \bullet F_2) + \mathcal{F}^u_{\text{Top}}(T_1 \bullet F_1, F_2) - \mathcal{F}^u_{\text{Top}}(F_1, F_2).
\end{aligned}
$$

$$
\mathcal{K}^u_{\text{LcaSg}}(T_1, T_2) = \sum_{u \in T_1} \sum_{v \in T_2} \mathcal{K}^u_{\text{Top}}(T_1[u], T_2[v]).
$$

$$
\begin{aligned}
\mathcal{K}^u_{\text{Lca}}(T_1, T_2) &= \sum_{u \in T_1} \sum_{v \in T_2} \mathcal{T}^u_{\text{Lca}}(T_1[u], T_2[v]), \\
\mathcal{T}^u_{\text{Lca}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot \left(1 + \sum\nolimits_{M \in BM(d(s),d(t))} \mathcal{F}^u_{\text{Lca}}(F_1^M, F_2^M)\right), \\
\mathcal{F}^u_{\text{Lca}}(\emptyset, F) &= \mathcal{F}^u_{\text{Lca}}(F, \emptyset) = 0, \\
\mathcal{F}^u_{\text{Lca}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{K}^u_{\text{Lca}}(T_1, T_2) \cdot (1 + \mathcal{F}^u_{\text{Lca}}(F_1, F_2)) \\
&\quad + \mathcal{F}^u_{\text{Lca}}(F_1, T_2 \bullet F_2) + \mathcal{F}^u_{\text{Lca}}(T_1 \bullet F_1, F_2) - \mathcal{F}^u_{\text{Lca}}(F_1, F_2).
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{K}^u_{\text{Acc}}(\emptyset, F) &= \mathcal{K}^u_{\text{Acc}}(F, \emptyset) = \mathcal{TF}^u_{\text{Acc}}(T, \emptyset) = 0, \\
\mathcal{K}^u_{\text{Acc}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^u_{\text{Acc}}(T_1, T_2) \cdot (\mathcal{F}^u_{\text{Acc}}(F_1, F_2) - 1) \\
&\quad + \mathcal{TF}^u_{\text{Acc}}(T_1, T_2 \bullet F_2) - \mathcal{TF}^u_{\text{Acc}}(T_1, F_2) \\
&\quad + \mathcal{TF}^u_{\text{Acc}}(T_2, T_1 \bullet F_1) - \mathcal{TF}^u_{\text{Acc}}(T_2, F_1) \\
&\quad + \mathcal{K}^u_{\text{Acc}}(T_1 \bullet F_1, F_2) - \mathcal{K}^u_{\text{Acc}}(F_1, F_2), \\
\mathcal{TF}^u_{\text{Acc}}(v(F_1), T_2 \bullet F_2) &= \mathcal{T}^u_{\text{Acc}}(v(F_1), T_2) - \mathcal{TF}^u_{\text{Acc}}(T_2, F_1) + \mathcal{TF}^u_{\text{Acc}}(v(F_1), F_2) \\
&\quad - \mathcal{K}^u_{\text{Acc}}(F_1, F_2) + \mathcal{K}^u_{\text{Acc}}(F_1, T_1 \bullet F_2), \\
\mathcal{T}^u_{\text{Acc}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot \left(1 + \sum\nolimits_{M \in BM(d(s),d(t))} \mathcal{F}^u_{\text{Acc}}(F_1', F_2')\right) \\
&\quad + \sum\nolimits_{F_2' \in pm(F_2)} \mathcal{TF}^u_{\text{Acc}}(u(F_1), F_2') + \sum\nolimits_{F_1' \in pm(F_1)} \mathcal{TF}^u_{\text{Acc}}(v(F_2), F_1') \\
&\quad - \sum\nolimits_{M \in BM(d(s),d(t))} \mathcal{K}^u_{\text{Acc}}(F_1^M, F_2^M), \\
\mathcal{F}^u_{\text{Acc}}(\emptyset, F) &= \mathcal{F}^u_{\text{Acc}}(F, \emptyset) = 0, \\
\mathcal{F}^u_{\text{Acc}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^u_{\text{Acc}}(T_1, T_2) \cdot (1 + \mathcal{F}^u_{\text{Acc}}(F_1, F_2)) \\
&\quad + \mathcal{F}^u_{\text{Acc}}(F_1, T_2 \bullet F_2) + \mathcal{F}^u_{\text{Acc}}(T_1 \bullet F_1, F_2) - \mathcal{F}^u_{\text{Acc}}(F_1, F_2).
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{K}^u_{\text{Ilst}}(\emptyset, F) &= \mathcal{K}^u_{\text{Ilst}}(F, \emptyset) = \mathcal{TF}^u_{\text{Ilst}}(T, \emptyset) = 0, \\
\mathcal{K}^u_{\text{Ilst}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^u_{\text{Ilst}}(T_1, T_2) \cdot (\mathcal{F}^u_{\text{Ilst}}(F_1, F_2) - 1) \\
&\quad + \mathcal{TF}^u_{\text{Ilst}}(T_1, T_2 \bullet F_2) - \mathcal{TF}^u_{\text{Ilst}}(T_1, F_2) \\
&\quad + \mathcal{TF}^u_{\text{Ilst}}(T_2, T_1 \bullet F_1) - \mathcal{TF}^u_{\text{Ilst}}(T_2, F_1) \\
&\quad + \mathcal{K}^u_{\text{Ilst}}(T_1 \bullet F_1, F_2) - \mathcal{K}^u_{\text{Ilst}}(F_1, F_2), \\
\mathcal{TF}^u_{\text{Ilst}}(v(F_1), T_2 \bullet F_2) &= \mathcal{T}^u_{\text{Ilst}}(v(F_1), T_2) - \mathcal{TF}^u_{\text{Ilst}}(T_2, F_1) + \mathcal{TF}^u_{\text{Ilst}}(v(F_1), F_2) \\
&\quad - \mathcal{K}^u_{\text{Ilst}}(F_1, F_2) + \mathcal{K}^u_{\text{Ilst}}(F_1, T_1 \bullet F_2), \\
\mathcal{T}^u_{\text{Ilst}}(u(F_1), v(F_2)) &= \sigma(l(u), l(v)) \cdot \left(1 + \sum\nolimits_{M \in BM(d(s),d(t))} \mathcal{K}^u_{\text{Ilst}}(F_1^M, F_2^M)\right) \\
&\quad + \sum\nolimits_{F_2' \in pm(F_2)} \mathcal{TF}^u_{\text{Ilst}}(u(F_1), F_2') + \sum\nolimits_{F_1' \in pm(F_1)} \mathcal{TF}^u_{\text{lst}}(v(F_2), F_1') \\
&\quad - \sum\nolimits_{M \in BM(d(s),d(t))} \mathcal{K}^u_{\text{Ilst}}(F_1^M, F_2^M), \\
\mathcal{F}^u_{\text{Ilst}}(\emptyset, F) &= \mathcal{F}^u_{\text{Ilst}}(F, \emptyset) = 0, \\
\mathcal{F}^u_{\text{Ilst}}(T_1 \bullet F_1, T_2 \bullet F_2) &= \mathcal{T}^u_{\text{Ilst}}(T_1, T_2) \cdot (1 + \mathcal{F}^u_{\text{Ilst}}(F_1, F_2)) \\
&\quad + \mathcal{F}^u_{\text{Ilst}}(F_1, T_2 \bullet F_2) + \mathcal{F}^u_{\text{Ilst}}(T_1 \bullet F_1, F_2) - \mathcal{F}^u_{\text{Ilst}}(F_1, F_2).
\end{aligned}
$$

Fig. 3. The recurrences of computing $\mathcal{K}^u_{\mathtt{A}}(T_1, T_2)$ for $\mathtt{A} \in \{\text{Top}, \text{LcaSg}, \text{Lca}, \text{Acc}, \text{Ilst}\}$.

permuted forests of F. Then, Fig. 3 illustrates the recurrences of computing $\mathcal{K}^u_{\mathtt{A}}(T_1, T_2)$.

Theorem 3. *For $\mathtt{A} \in \{\text{Top}, \text{LcaSg}, \text{Lca}, \text{Acc}, \text{Ilst}\}$, the recurrences in Fig. 3 correctly compute $\mathcal{K}^u_{\mathtt{A}}(T_1, T_2)$ in $O(nmD^D)$ time, where $n = |T_1|$, $m = |T_2|$ and $D = \max\{d(T_1), d(T_2)\}$. Hence, if the degrees of unordered trees are bounded by some constant, then we can compute $\mathcal{K}^u_{\mathtt{A}}(T_1, T_2)$ in $O(nm)$ time.*

Proof. Since $|BM(s,t)| = {}_sP_t$ and the number of all permuted forests of F_1 (*resp.*, F_2) is ${}_sP_1$ (*resp.*, ${}_tP_1$), the number of occurrences of the formula $\mathcal{F}^u_{\mathtt{A}}(F_1^M, F_2^M)$ is bounded by D^D and the number of occurrences of the formulas $\mathcal{F}^u_{\mathtt{A}}(u(F_1), F_2')$ and $\mathcal{F}^u_{\mathtt{A}}(F_1', v(F_2))$ for $\mathtt{A} \in \{\textsc{Acc}, \textsc{Ilst}\}$ is bounded by D^D. In both cases, the number of occurrences of the formulas is $O(D^D)$. Since every pair $(u,v) \in T_1 \times T_2$ is called just once, we can compute $K^u_{\mathtt{A}}(T_1,T_2)$ in $O(nmD^D)$ time by using dynamic programming. Hence, the time complexity in the statement holds. Also we can show the correctness by extending Theorem 1. $\square$

5 #P-Completeness for Unordered Trees

Since we cannot apply the #P-completeness of [3,6] to the top-down mapping kernel for unordered trees directly, in this section, we show that the problem of counting all the specific top-down mappings (or bottom-up mappings) is #P-complete.

Let M be a mapping between T_1 and T_2. We say that M is *label-preserving* (or an *indel mapping*) if it always holds that $l(u) = l(v)$ for every $(u,v) \in M$. Also we say that M is *leaf-extended* if, for every $(u,v) \in M$, there exists $(u',v') \in M$ such that $u \in anc(u')$, $v \in anc(v')$, $u' \in lv(T_1)$ and $v' \in lv(T_2)$. Then, we deal with a label-preserving leaf-extended top-down mapping M between unordered trees T_1 and T_2, which we denote $M \in \mathcal{M}^u_{\textsc{llTop}}(T_1,T_2)$.

Theorem 4 (*cf.*, [3]). *The problem of counting all the mappings in $\mathcal{M}^u_{\textsc{llTop}}(T_1,T_2)$ is #P-complete.*

Proof. Valiant [19] has shown that the problem of counting all the matchings in a bipartite graph, which we denote #BipartiteMatching, is #P-complete. Then, we give two trees such that the number of all the label-preserving leaf-extended top-down mapping between them is equal to the output of #Bipartite Matching. Here, for a forest F and a node v such that $l(v) = a$, we denote $v(F)$ by $a(F)$.

Let $G = (X \cup Y, E)$ be a bipartite graph. For $v \in X \cup Y$, we denote a neighbor of v by $N(v)$. It is obvious that $N(v) \subseteq Y$ if $v \in X$ and $N(v) \subseteq X$ if $v \in Y$. Then, we construct $T_x = a(\{xy \mid y \in N(x)\})$ for every $x \in X$ and $T_1 = a(\{T_x \mid x \in X\})$. Similarly, we construct $T_y = a(\{xy \mid x \in N(y)\})$ for every $y \in Y$ and $T_2 = a(\{T_y \mid y \in Y\})$. Here, we regard an edge xy in G as the label of a leaf in T_x and T_y. Figure 4 illustrates an example of the above construction of T_1 and T_2 from a bipartite graph G.

For a matching $B \subseteq E$ in G we construct the label-preserving leaf-extended top-down mapping M between T_1 and T_2 such that:

$$M = \begin{cases} \emptyset & \text{if } B = \emptyset, \\ \{(r(T_1), r(T_2))\} \cup \bigcup_{xy \in B} M_{xy} & \text{if } B \neq \emptyset, \end{cases}$$

$$M_{xy} = \left\{ \begin{array}{c} (u_1,v_1),(u_2,v_2) \\ \in V(T_x) \times V(T_y) \end{array} \middle| \begin{array}{l} u_1 = par(u_2), v_1 = par(v_2), \\ u_2 \in lv(T_x), v_2 \in lv(T_y) \\ l(u_1) = l(v_1) = a, l(u_2) = l(v_2) = xy \end{array} \right\}.$$

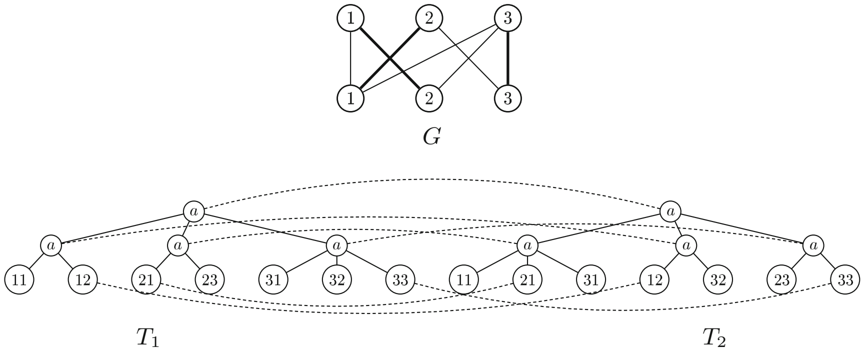

Fig. 4. A bipartite graph G and the trees T_1 and T_2.

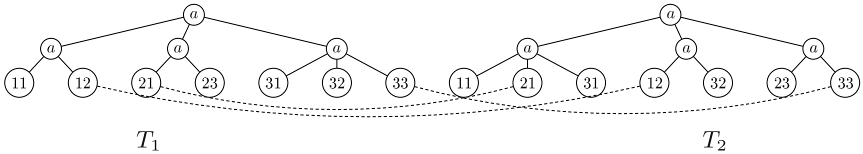

Fig. 5. Trees T_1 and T_2 in Corollary 2.

For example, let B be a matching $\{12, 21, 33\}$ in G illustrated in Fig. 4 as think lines. Then, the label-preserving leaf-extended top-down mapping M between T_1 and T_2 is illustrated by dashed lines.

Note that, by the definition of T_x and T_y, M_{xy} is a label-preserving leaf-extended top-down mapping between T_x and T_y. Also M_{xy} is corresponding to an element xy in a matching of G. Furthermore, no label-preserving leaf-extended top-down mapping M_{xy} between T_1 and T_2 contains more than one path from the root to leaves in T_x or T_y, that is, M_{xy} contains zero or one path in T_x and T_y.

Hence, a matching B in G determines the label-preserving leaf-extended top-down mapping M between T_1 and T_2 uniquely and vice versa. Then, the number of all the matchings in G which is the output of #BipartiteMatching is equal to the number of all the label-preserving leaf-extended top-down mappings between T_1 and T_2. Hence, the statement holds. □

Finally, we denote all the *label-preserving bottom-up mappings* between unordered trees T_1 and T_2 by $\mathcal{M}^u_{\text{LBOT}}(T_1, T_2)$. Then, the proofs of [3,6] or the above proof imply the following corollary. Here, it is sufficient to construct a matching B in Fig. 4 to a mapping $\bigcup_{xy \in B} \{(u, v) \in lv(T_x) \times lv(T_y) \mid l(u) = l(v) = xy\}$ as Fig. 5, for example.

Corollary 2 (*cf.*, [3,6]). *The problem of counting all the mappings in* $\mathcal{M}^u_{\text{LBOT}}(T_1, T_2)$ *is* #P*-complete.*

6 Conclusion

In this paper, for mapping $\mathtt{A} \in \{\textsc{Top}, \textsc{LcaSg}, \textsc{Lca}\}$, we have designed the recurrences to compute $\mathcal{K}^{o}_{\mathtt{A}}(T_1, T_2)$ and $\mathcal{K}^{b}_{\mathtt{A}}(T_1, T_2)$ in $O(nm)$ time and to compute $\mathcal{K}^{c}_{\mathtt{A}}(T_1, T_2)$ and $\mathcal{K}^{cb}_{\mathtt{A}}(T_1, T_2)$ in $O(nmdD)$ time. Also, we have designed the recurrences to compute $\mathcal{K}^{u}_{\mathtt{A}}(T_1, T_2)$ in $O(nmD^D)$ time, which implies that we can compute $\mathcal{K}^{u}_{\mathtt{A}}(T_1, T_2)$ in $O(nm)$ time if the degrees of T_1 and T_2 are bounded by some constant. On the other hand, we show that the problem of computing $\mathcal{K}^{u}_{\mathrm{LLTop}}(T_1, T_2)$ and $\mathcal{K}^{u}_{\mathrm{LBot}}(T_1, T_2)$ are #P-complete.

For $\mathcal{M}_{\mathrm{Aln}}$ (alignable mapping [8], less-constrained mapping [11]), from [4,24], we conjecture that we can compute $\mathcal{K}^{b}_{\mathrm{Aln}}(T_1, T_2)$ in $O(nmD^2)$ time, $\mathcal{K}^{\pi}_{\mathrm{Aln}}(T_1, T_2)$ in $O(nmdD^3)$ time ($\pi \in \{c, cb\}$) and $\mathcal{K}^{u}_{\mathrm{Aln}}(T_1, T_2)$ in polynomial time if the degrees of T_1 and T_2 are bounded by some constant. Hence, it is a future work to investigate whether or not the above conjecture is correct.

In the proof of Theorem 4 and Corollary 2, the condition of label-preserving and leaf-extended are essential. If these condisions are not met, we must count all the other (standard) top-down or bottom-up mappings that are not label-preserving or leaf-extended. In order to show that the problem of counting all the mappings in $\mathcal{M}^{u}_{\mathrm{Top}}(T_1, T_2)$, $\mathcal{M}^{u}_{\mathrm{Bot}}(T_1, T_2)$ and then $\mathcal{K}^{u}_{\mathtt{A}}(T_1, T_2)$ for $\mathtt{A} \in \{\textsc{LcaSg}, \textsc{Lca}, \textsc{Acc}, \textsc{Ilst}, \textsc{Aln}\}$ are all #P-complete, we must use the Cook-reduction [6,19] from #BipartiteMatching, which is more complex than the proof of Theorem 4. On the other hand, this paper has shown that we can compute $\mathcal{K}^{u}_{\mathtt{A}}(T_1, T_2)$ for bounded-degree unordered trees. Hence, it is an important future work to investigate whether or not the problem of computing $\mathcal{K}^{u}_{\mathtt{A}}(T_1, T_2)$ is #P-complete when degrees are unbounded.

References

1. Chawathe, S.S.: Comparing hierarchical data in external memory. In: Proceedings of the VLDB 1999, pp. 90–101 (1999)
2. Gärtner, T.: Kernels for Structured Data. World Scientific Publishing, Singapore (2008)
3. Hamada, I., Shimada, T., Nakata, D., Hirata, K., Kuboyama, T.: Agreement subtree mapping kernel for phylogenetic trees. In: Nakano, Y., Satoh, K., Bekki, D. (eds.) JSAI-isAI 2013. LNCS, vol. 8417, pp. 321–336. Springer, Heidelberg (2014)
4. Jiang, T., Wang, L., Zhang, K.: Alignment of trees - an alternative to tree edit. Theoret. Comput. Sci. **143**, 137–148 (1995)
5. Kan, T., Higuchi, S., Hirata, K.: Segmental mapping and distance for rooted ordered labeled trees. Fundamenta Informaticae **132**, 1–23 (2014)
6. Kashima, H., Sakamoto, H., Koyanagi, T.: Tree kernels. J. JSAI **21**, 1–9 (2006). (in Japanese)
7. Kimura, D., Kuboyama, T., Shibuya, T., Kashima, H.: A subpath kernel for rooted unordered trees. J. JSAI **26**, 473–482 (2011). (in Japanese)
8. Kuboyama, T.: Matching and learning in trees. Ph.D. thesis, University of Tokyo (2007)

9. Kuboyama, T., Hirata, K., Aoki-Kinoshita, K.F.: An efficient unordered tree kernel and its application to glycan classification. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 184–195. Springer, Heidelberg (2008)
10. Kuboyama, T., Shin, K., Kashima, H.: Flexible tree kernels based on counting the number of tree mappings. In: Proceedings of the MLG 2006, pp. 61–72 (2006)
11. Lu, C.L., Su, Z.-Y., Tang, C.Y.: A new measure of edit distance between labeled trees. In: Wang, J. (ed.) COCOON 2001. LNCS, vol. 2108, pp. 338–348. Springer, Heidelberg (2001)
12. Lu, S.-Y.: A tree-to-tree distance and its application to cluster analysis. IEEE Trans. Pattern Anal. Mach. Intell. **1**, 219–224 (1979)
13. Selkow, S.M.: The tree-to-tree editing problem. Inform. Process. Lett. **6**, 184–186 (1977)
14. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge University Press, Cambridge (2004)
15. Shin, K.: Engineering positive semedefinite kernels for trees - a framework and a survey. J. JSAI **24**, 459–468 (2009). (in Japanese)
16. Shin, K., Cuturi, M., Kuboyama, T.: Mapping kernels for trees. In: Proceedings of ICML 2011 (2011)
17. Shin, K., Kuboyama, T.: A generalization of Haussler's convolutioin kernel - Mapping kernel and its application to tree kernels. J. Comput. Sci. Tech. **25**, 1040–1054 (2010)
18. Tai, K.-C.: The tree-to-tree correction problem. J. ACM **26**, 422–433 (1979)
19. Valiant, L.G.: The complexity of enumeration and reliablity problems. SIAM J. Comput. **8**, 410–421 (1979)
20. Valiente, G.: An efficient bottom-up distance between trees. In: Proceedings of SPIRE 2001, pp. 212–219 (2001)
21. Wang, J.T.L., Zhang, K.: Finding similar consensus between trees: an algorithm and a distance hierarchy. Pattern Recogn. **34**, 127–137 (2001)
22. Yamamoto, Y., Hirata, K., Kuboyama, T.: Tractable and intractable variations of unordered tree edit distance. Int. J. Found. Comput. Sci. **25**, 307–329 (2014)
23. Yoshino, T., Hirata, K.: Hierarchy of segmental and alignable mapping for rooted labeled trees. In: Procedings of DDS 2013, pp. 62–69 (2013)
24. Yoshino, T., Hirata, K.: Alignment of cyclically ordered trees. In: Proceedings of ICPRAM 2015 (2015, to appear)
25. Zhang, K.: Algorithms for the constrained editing distance between ordered labeled trees and related problems. Pattern Recogn. **28**, 463–474 (1995)
26. Zhang, K.: A constrained edit distance between unordered labeled trees. Algorithmica **15**, 205–222 (1996)
27. Zhang, K., Wang, J., Shasha, D.: On the editing distance between undirected acyclic graphs. Int. J. Found. Comput. Sci. **7**, 43–58 (1996)