

Approximation: Theory and Algorithms

The Tree Edit Distance (I)

Nikolaus Augsten

Free University of Bozen-Bolzano
Faculty of Computer Science
DIS

Unit 6 – April 3, 2009

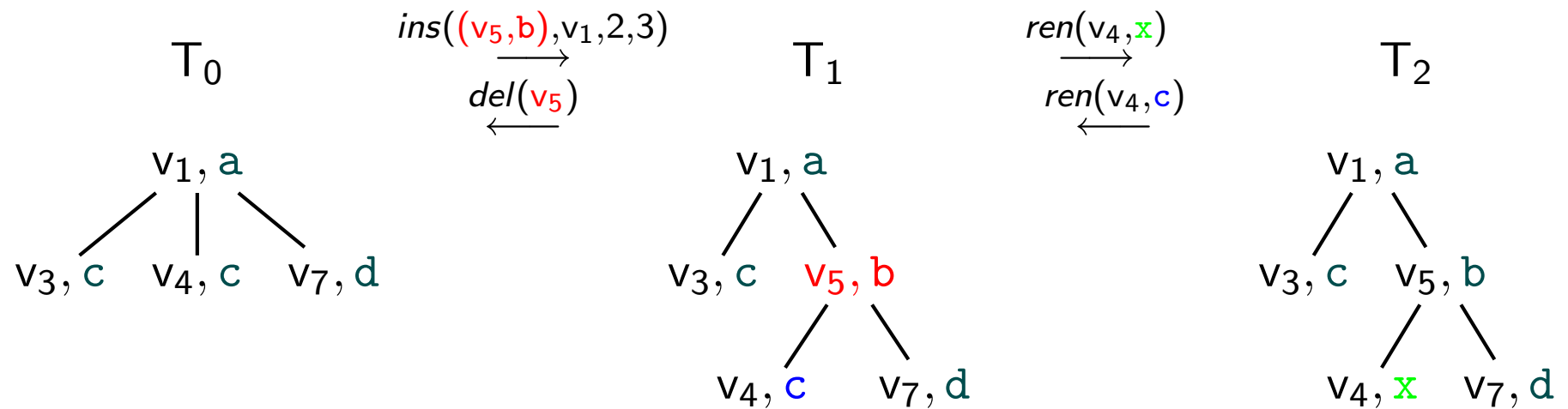
Outline

- 1 Tree Edit Distance
 - Preliminaries and Definition
 - Forest Distance
- 2 Conclusion

Edit Operations

- We assume **ordered, labeled trees**
- **Rename node**: $ren(v, l')$
 - change label l of v to $l' \neq l$
- **Delete node**: $del(v)$ (v is not the root node)
 - remove v
 - connect v 's children directly to v 's parent node (preserving order)
- **Insert node**: $ins(v, p, k, m)$
 - insert new node v as a child of p at position k
 - substitute children c_k, c_{k+1}, \dots, c_m of p with v
 - insert c_k, c_{k+1}, \dots, c_m as children of the new node v (preserving order)
- Insert and delete are **inverse** edit operations (i.e., insert undoes delete and vice versa)

Example: Edit Operations



Edit Cost Function

- Represent **edit operation as node pair** $(a, b) \neq (\varepsilon, \varepsilon)$
(written also as $a \rightarrow b$, ε is the null node)
 - $a \rightarrow \varepsilon$: delete a
 - $\varepsilon \rightarrow b$: insert b
 - $a \rightarrow b$: rename a to b
- **Cost function** $\alpha(a \rightarrow b)$:
 - assign to each edit operation a non-negative real
 - cost can be different for different nodes
 - we use constant costs $\omega_{ins}, \omega_{del}, \omega_{ren}$
- We constrain α to be a **distance metric**:
 - (i) triangle inequality: $\alpha(a \rightarrow b) + \alpha(b \rightarrow c) \geq \alpha(a, c)$
 - (ii) symmetry: $\alpha(a \rightarrow b) = \alpha(b \rightarrow a)$
 - (iii) identity: $\alpha(a \rightarrow b) = 0 \Leftrightarrow \lambda(a) = \lambda(b)$

Definition

Definition (Tree Edit Distance)

The tree edit distance between two trees is the minimum cost sequence of node edit operations (node deletion, node insertion, node rename) that transforms one tree into the other.

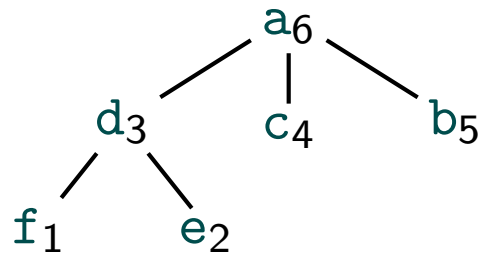
- Cost of a sequence $S = \{s_1, \dots, s_n\}$ of edit operations:

$$\alpha(S) = \sum_{i=1}^{i=n} \alpha(s_i)$$

- As the cost function is a metric, also the tree edit distance is a metric.

Postorder Traversal

- **Postorder traversal** of an ordered tree:
 - traverse subtrees rooted in children of current node (from left to right) in postorder
 - visit current node
- **Example:** postorder = (f, e, d, c, b, a)



- **Observations:** The postorder number of a node is **larger than**
 - the postorder numbers of all its **descendants**
 - the postorder numbers of all its **left siblings**

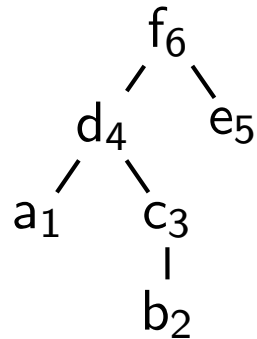
Subtrees and Subforests

- A **subtree** T' of T is a tree that consists of:
 - a subset of the nodes of T : $N(T') \subseteq N(T)$
 - all edges in T that connect these nodes: $E(T') \subseteq E(T)$
- **Ordered Forests**:
 - a forest is a set of trees
 - an *ordered* forest is a sequence of trees
- **Ordered Subforests** of a tree T :
 - formed by subtrees of T with disjoint nodes
 - subtrees ordered by the postorder number in T of their root

Example: Subtrees and Subforests

- Example tree (postorder numbers are node IDs):

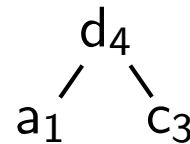
$$T = (\{v_1, v_2, v_3, v_4, v_5, v_6\}, \{(v_6, v_4), (v_6, v_5), (v_4, v_1), (v_4, v_3), (v_3, v_2)\})$$



- Two subtrees of T :

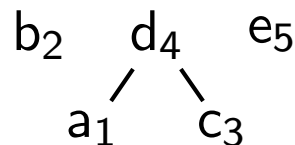
$$T'_1 = (\{v_3\}, \{\}) \quad T'_2 = (\{v_4, v_1, v_3\}, \{(v_4, v_1), (v_4, v_3)\})$$

c_3



- Ordered subforest of T :

$$F = ((\{v_2\}, \{\}), (\{v_4, v_1, v_3\}, \{(v_4, v_1), (v_4, v_3)\}), (\{v_5\}, \{\}))$$



Notation I/II

- We use the following notation:
 - $T[i]$ is the i -th node of T in **postorder** (we say: $T[i]$ is node i of T)
 - $T[i..j]$ is the subforest formed by the nodes $T[i]$ to $T[j]$
 - $l(i)$ is the left-most leaf descendant of node $T[i]$
 - $desc(T[i])$ is the set of all descendants of $T[i]$ including $T[i]$ itself (elements of $desc(T[i])$ are usually denoted with d_i)
- Node identifiers:
 - we assume that the node IDs correspond to their postorder number
 - we refer to a node simply by its ID, if the context is clear

Notation II/II

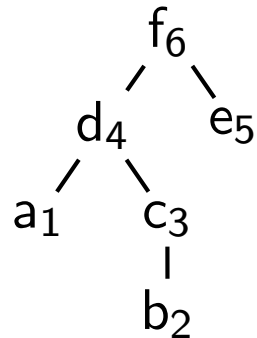
- We consider only **special subforests** of the form:

$$T[l(i)..d_i]$$

- **Observations:**
 - If a node k is in $T[l(i)..d_i]$, also all its descendants are in $T[l(i)..d_i]$.
 - $T[l(i)..i]$ is the subtree consisting of node i and all its descendants
 - $T[l(r)..r] = T$ if r is the root node
- We call $T[l(i)..i]$ the **subtree rooted in $T[i]$**

Example: Subtrees and Subforests

- Example tree:



- Descendants: $\text{desc}(T[4]) = \{T[1], T[2], T[3], T[4]\}$
- Left-most leaf descendants: $l(1) = l(4) = l(6) = T[1]$
- Some ordered subforests of the form $T[l(i)..d_i]$, $d_i \in \text{desc}(i)$:

$T[l(4)..3]$	$T[l(4)..4]$	$T[l(6)..5]$	$T[l(5)..5]$
<pre> a1 c3 b2 </pre>	<pre> d4 / \ a1 c3 b2 </pre>	<pre> d4 e5 / \ a1 c3 b2 </pre>	<pre> e5 </pre>

Edit Mapping

Definition (Edit Mapping)

An **edit mapping** M between T_1 and T_2 is a set of node pairs that satisfy the following conditions:

- (1) $(a, b) \in M \Rightarrow a \in N(T_1), b \in N(T_2)$
- (2) $a = \text{root}(T_1)$ and $b = \text{root}(T_2) \Leftrightarrow (a, b) \in M$
- (3) for any two pairs (a, b) and (x, y) of M :
 - (i) $a = x \Leftrightarrow b = y$ (one-to-one)
 - (ii) a precedes x in preorder¹ $\Leftrightarrow b$ precedes y in preorder (sibling order preserved)
 - (iii) a is an ancestor of $x \Leftrightarrow b$ is an ancestor of y (ancestor order preserved)

¹i.e., a is to the left of x

Edit Mapping

- The **cost of the mapping** is

$$\alpha(M) = \sum_{(a,b) \in M} \alpha(a \rightarrow b) + \sum_{a \in D} \alpha(a \rightarrow \varepsilon) + \sum_{b \in I} \alpha(\varepsilon \rightarrow b),$$

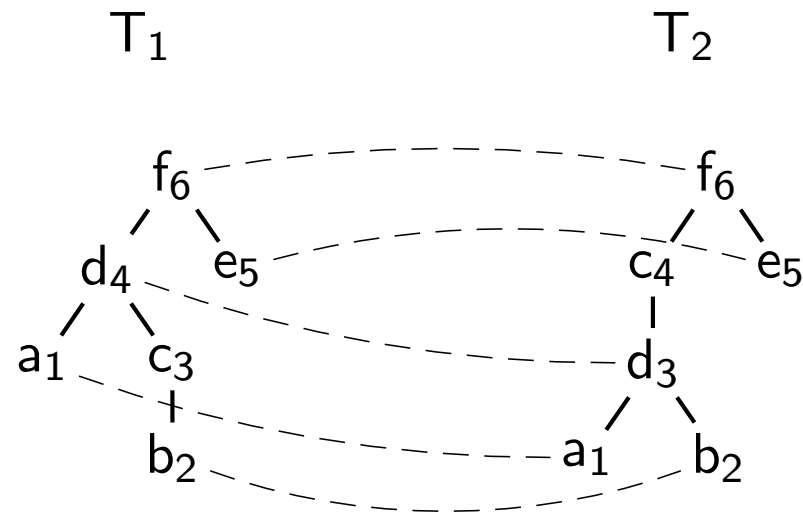
where D and I are the nodes of T_1 and T_2 , respectively, not touched by a line in M .

- Alternative definition of the **tree edit distance** $ted(T_1, T_2)$:

$$ted(T_1, T_2) = \min\{\alpha(M) \mid M \text{ is a mapping from } T_1 \text{ to } T_2\}$$

Example: Mapping

- $M = \{(T_1[6], T_2[6]), (T_1[5], T_2[5]), (T_1[4], T_2[3]), (T_1[1], T_2[1]), (T_1[2], T_2[2])\}$
 - $T_1[3]$ is deleted from T_1
 - $T_2[4]$ is inserted into T_2
 - no proper rename (only rename to the same label with cost 0)



Forest Distance

Definition (Forest Distance)

The forest distance between two ordered forests is the minimum cost sequence of node edit operations (node deletion, node insertion, node rename) that transforms one forest into the other.

- **Edit mapping and edit operations** in a forest:
 - Each tree in the forest has a root node.
 - We imagine a dummy node that is the parent of all these root nodes.
 - The sibling order in the imaginary tree is the tree order in the forest.
 - The dummy node connects the forest to become a tree.
 - Then all edit operations and edit mappings valid between two imaginary trees are valid also between the respective forests.
- The tree edit distance is a **special case** of the forest distance, where the forest has the form $T[l(i)..i]$, i.e. it consists of a single tree.

Recursive Formula: Distance to the Empty Forest

Lemma (Empty Forest [ZS89, AG97])

Given two trees T_1 and T_2 , $i \in N(T_1)$ and $d_i \in \text{desc}(i)$, $j \in N(T_2)$ and $d_j \in \text{desc}(j)$, then:

- (i) $f\text{dist}(\emptyset, \emptyset) = 0$
- (ii) $f\text{dist}(T_1[l(i)..d_i], \emptyset) = f\text{dist}(T_1[l(i)..d_i - 1], \emptyset) + \omega_{del}$
- (iii) $f\text{dist}(\emptyset, T_2[l(j)..d_j]) = f\text{dist}(\emptyset, T_2[l(j)..d_j - 1]) + \omega_{ins}$

Proof.

Case (i) requires no edit operation. In cases (ii), the distance corresponds to the cost of deleting all nodes in $T_1[l(i)..d_i]$. In cases (iii), the distance corresponds to the cost of inserting all nodes in $T_2[l(j)..d_j]$. \square

Summary

- XML as an ordered, labeled tree
 - DOM and SAX for parsing XML
- Tree Edit Distance
 - definition
 - edit distance mapping
 - recursive formula for empty forests

What's Next?

- Tree Edit Distance (II):
 - recursive formula for non-empty forests
 - dynamic programming algorithm
 - edit distance example



Alberto Apostolico and Zvi Galil, editors.

Pattern Matching Algorithms, chapter Tree Pattern Matching, pages 341–371.

Oxford University Press, 1997.



Kaizhong Zhang and Dennis Shasha.

Simple fast algorithms for the editing distance between trees and related problems.

SIAM Journal on Computing, 18(6):1245–1262, 1989.