TREE CORRESPONDENCE PROBLEMS*

by

J. Albert[1] and K. Culik II[2]

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Research Report CS-80-45

October 1980

# Faculty of Mathematics

University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

# TREE CORRESPONDENCE PROBLEMS*

by

J. Albert[1] and K. Culik II[2]

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Research Report CS-80-45

October 1980

*) This research was supported by the Natural Sciences and Engineering Council of Canada, Grant No. A7403.

[1] Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe, 7500 Karlsruhe, W-Germany.

[2] Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

## List of Symbols

| | |
|---|---|
| $\geq$ | greater or equal |
| $\leq$ | less or equal |
| $<$ | strictly less than |
| $\neq$ | not equal |
| $\cap$ | intersection |
| $\emptyset$ | empty set |
| $\{\ \}$ | set brackets |
| $\circ$ | little circle (function symbol) |
| $\uparrow$ | little vertical arrow (function symbol) |
| $\in$ | element of |
| $\notin$ | not element of |
| $\bigcup$ | union of several sets |
| $\cup$ | union of two sets |
| $\underset{\ell m}{\circ}$ | little circle with $\ell m$ below (abbreviation for leftmost), to be treated as one function symbol |
| $\underset{\ell m}{\uparrow}$ | little vertical arrow with $\ell m$ below, to be treated as one function symbol |
| # | number symbol |
| $\vert\ \vert$ | vertical bars, cardinality |
| $\rightarrow$ | small horizontal arrow, "production arrow" |
| $\Sigma$ | capital greek letter "sigma" |
| $\alpha$ | lower case greek letter "alpha" |
| $\sigma$ | lower case greek letter "sigma" |
| $\tau$ | lower case greek letter "tau" |
| $\rho$ | lower case greek letter "rho" |
| $\mu$ | lower case greek letter "mu" |
| $\omega$ | lower case greek letter "omega" |
| $\ell$ | lower case script "$\ell$" |
| $*$ | small star as super script |

## Abstract

We investigate decision problems for the concatenation of rooted trees, considering different types of trees as well as different modes of concatenation.

Our main theorem establishes the undecidability of the following problem: Given two lists

$$A = (x_1, \ldots, x_k) \text{ and } B = (y_1, \ldots, y_k)$$

of rooted ordered unlabeled trees, does there exist a sequence of indices $i_1, \ldots, i_n$ such that the concatenation of $x_{i_1}, x_{i_2}, \ldots, x_{i_n}$ - in that order - equals the concatenation of $y_{i_1}, y_{i_2}, \ldots, y_{i_n}$, in that order?

## 1. Introduction

The well-known Post Correspondence Problem - PCP for short - asks whether for two lists of strings there is a unifying sequence of indices, i.e. concatenating the words from the first list in the given order of the index-sequence yields the same string as the concatenation of the words of the second list in that order. This problem is known to be equivalent to the halting problem of Turing machines and hence undecidable, c.f. [2].

On the other hand we get a decidable problem if we drop the restriction of the common order of indices. Obviously, this problem can be reduced to the emptiness-problem of the intersection of two regular sets. The fact that trees are the most natural and important generalization of strings motivates us to formulate the above mentioned problems for several types of trees. The concatenation of two trees is done by attaching the root of the second tree to a leaf of the first one. Thus, concatenation of trees becomes a nondeterministic operation and therefore we have to deal with sets of trees, where we had single concatenated strings in the original problems.

For the generalization of the Post Correspondence Problem to (rooted ordered) labeled trees we get a straightforward undecidability result, since strings can easily be encoded as 'string-like' labeled trees.

If we do not require the same sequences of indices for the concatenation of the trees, the decidability of that problem is not that obvious as it is for strings but can be established by an effective reduction to the intersection problem of two parenthesis languages.

More interesting is the tree version of the PCP over a one-letter alphabet, where we have lists of unlabeled rooted trees. Since the one-letter case of the PCP is decidable, the undecidability for the corresponding problem for trees is of interest and actually quite non-trivial as our proof will indicate. Our proof-technique is then carried over to a different mode of concatenation and also to 'oriented' trees, where the relative order of the subtrees is not considered.

## 2. Preliminaries

For the definitions concerning trees we follow the terminology of [1] and refer the reader to this book for all notions not explicitely explained in the sequel.

Definition

A directed graph $G$ is a pair $(A,R)$, where $A$ is a set of elements called nodes and $R$ is a relation on $A$. An element of $R$ is called an edge of $G$. A labeling of the graph is a pair of functions $f$ and $g$, mapping $A$ and $R$ resp. to some (possibly distinct) sets.

A sequence of nodes $(a_0,a_1,\dots,a_n)$, $n \geq 1$, is a path of length $n$ from $a_0$ to node $a_n$, if there is an edge which leaves node $a_{i-1}$ and enters $a_i$ for $1 \leq i \leq n$.

The in-degree of a node $p$ is the number of edges entering $p$ and the out-degree of $p$ is the number of edges leaving $p$.

Definition

An oriented tree $T$ is a directed graph $G = (A,R)$ with a specific node $r$ in $A$, called root, such that

i) $r$ has in-degree zero,

ii) All other nodes of $T$ have in-degree 1, and

iii) For every node $p$ there is a path from $r$ to $p$.

In a tree a node with out-degree zero will be called a leaf.

Definition

An ordered graph is a pair $(A,R)$ where $A$ is a set of nodes as before and $R$ is a set of linearly ordered lists of edges such that each element of $R$ is of the form $((a,b_1),(a,b_2),\dots,(a,b_n))$ where $a$ is a distinct member of $A$, and this element of $R$ indicates that there are $n$ edges leaving $a$, the first entering $b_1$, the second entering $b_2$, and so forth.

Definition

A labeling of an ordered graph $G = (A,R)$ is a pair of mappings $f$ and $g$ such that

i) $f : A \to S$ for some set $S$,

ii) $g$ maps $R$ to sequences of symbols from some set $T$ such that $g$ maps $((a,b_1),\ldots,(a,b_n))$ to a sequence of $n$ symbols of $T$.

Definition

An ordered tree is an ordered graph $(A,R)$ whose underlying graph is a tree and such that if $((a,b_1),\ldots,(a,b_n))$ is in $R$ then $b_i \neq b_j$ if $i \neq j$.

For an ordered (oriented) graph $g = (A,R)$ and a specific pair of labeling functions $(f,g)$ for $G$ we will write the quadruple $G' = (A,R,f,g)$ and call it a labeled ordered (oriented) graph.

We will now introduce two different types of concatenation of trees.

Definition

Let $T_1 = (A_1,R_1,f_1,g_1)$, $T_2 = (A_2,R_2,f_2,g_2)$ be two labeled ordered (oriented) trees with roots $r_1$ and $r_2$ resp. and $A_1 \cap A_2 = \emptyset$.

a) For each leaf $q$ of $T_1$ such that $f_1(q) = f_2(r_2)$ we define the labeled ordered (oriented) tree $S_q = (A_1 \cup A_2 - \{q\}, R_1 \cup R_2 \cup \{z_{r_2}\} - \{z_q\}, f_q, g_q)$ where $z_q$ is the unique element in $R_1$ in which $q$ occurs and $z_{r_2}$ is obtained by replacing $q$ by $r_2$ in $z_q$.
(Remember that $z_q$ is an ordered list of pairs of nodes for an ordered tree $T_1$ and a single pair for an oriented tree $T_1$.)
Furthermore, let

$$f_q(x) := \begin{cases} f_1(x) & \text{for } x \in A_1 - \{q\} \ , \\ f_2(x) & \text{for } x \in A_2 \ , \end{cases}$$

$$g_q(z) := \begin{cases} g_1(z) & \text{for } z \in R_1 - \{z_q\} \ , \\ g_1(z_q) & \text{for } z = z_{r_2} \ , \\ g_2(z) & \text{for } z \in R_2 \ . \end{cases}$$

Now define

$$\circ(T_1,T_2) = \{S_q \mid q \text{ leaf of } T_1, \text{s.t.} f_1(q) = f_2(r_2)\}.$$

b) Let $E_1$, $E_2$ be the sets of edge-labels of $T_1$, $T_2$ resp. For each leaf $q$ of $T_1$ and each edge-label $a \in E_1 \cup E_2$ let $S_{q,a}$ be the following labeled ordered (oriented) tree $S_{q,a} = (A_1 \cup A_2, R_1 \cup R_2 \cup \{(q,r_2)\}, f_{q,a}, g_{q,a})$ where

$$f_{q,a}(x) = \begin{cases} f_1(x) & \text{for } x \in A_1 \ , \\ f_2(x) & \text{for } x \in A_2 \ , \end{cases}$$

$$g_{q,a}(z) = \begin{cases} g_1(z) & \text{for } z \in R_1 \ , \\ g_2(z) & \text{for } z \in R_2 \ , \\ a & \text{for } z = (q,r_2) \ . \end{cases}$$

Then let

$$\uparrow(T_1,T_2) = \{S_{q,a} \mid q \text{ leaf of } T_1, \ a \in E_1 \cup E_2\}.$$

Thus in $\circ(T_1,T_2)$ a leaf of $T_1$ can be replaced by $T_2$ if leaf-label and root-label of $T_2$ match, whereas in $\uparrow(T_1,T_2)$ a new labeled edge is created between a leaf of $T_1$ and the root of $T_2$ .

Now we extend the definitions for $\circ$ and $\uparrow$ in the following manner.

Definition

Let $S_1,S_2,\ldots,S_n$ be sets of labeled ordered (oriented) trees and $F \in \{\circ,\uparrow\}$.

We define

i) $F(S_1,S_2) = \bigcup\limits_{T_1\in S_1, T_2\in S_2} F(T_1,T_2)$

ii) $F(S_1,S_2,\ldots,S_n) = F(F(S_1,S_2),S_3,\ldots,S_n)$ for $n \geq 3$.

Whenever some set $S_i$ is a singleton $\{T\}$ we drop the set brackets and write $T$ only.

Now we still need the notion of leftmost concatenation for the proof of our main result.

Definition

Let $T_1 = (A_1,R_1)$, $T_2 = (A_2,R_2)$ be unlabeled ordered trees with roots $r_1$ and $r_2$ resp. . Let $s$ be the (unique) leaf of $T_1$ such that for the path $(r_1 = q_0,q_1,q_2,\ldots,q_m = s)$ every pair $(q_i,q_{i+1})$ is the first one in the corresponding list of $R_1$. Then $s$ is called the leftmost leaf of $T_1$.

a) Now we define the leftmost concatenation of $T_1$, $T_2$ by

$$\underset{\ell m}{\circ}(T_1,T_2) = (A_1 \cup A_2 - \{s\}, R_1' \cup R_2)$$

where $R_1'$ equals $R_1$ except for the replacement of $s$ by $r_2$.

b) Analogously, let $\underset{\ell m}{\uparrow}(T_1,T_2)$ be the leftmost concatenation of $T_1$, $T_2$ with a new edge between $s$ and $r_2$.

$$\underset{\ell m}{\uparrow}(T_1,T_2) = (A_1 \cup A_2, R_1 \cup R_2 \cup \{(s,r_2)\})$$

c) For $T_1,T_2,\ldots,T_n$, $n \geq 3$ and $H \in \{\underset{\ell m}{\circ}, \underset{\ell m}{\uparrow}\}$ define

$$H(T_1,T_2,\ldots,T_n) = H(H(T_1,T_2),T_3,\ldots,T_n) \; .$$

Now we recall the formulation of the Post Correspondence Problem.

The Post Correspondence Problem (PCP for short) is to determine for two lists $A = (x_1, \ldots, x_k)$, $B = (y_1, \ldots, y_k)$ of non-empty words from a common alphabet $\Sigma$ whether or not there exist $i_1, i_2, \ldots, i_n$, $1 \le i_j \le k$ such that

$$x_{i_1} x_{i_2} \ldots x_{i_n} = y_{i_1} y_{i_2} \ldots y_{i_n} \quad .$$

For the undecidability of the PCP see [2].

## 3. Results

We start our main section with a decidability result, proven by a modification of a well-known encoding of trees in bracketed strings and a reduction of the problem to the emptiness problem of the intersection of two parenthesis languages.

### Theorem 1

Let $A = (s_1, s_2, \ldots, s_k)$, $B = (t_1, t_2, \ldots, t_k)$ be two lists of labeled ordered (oriented) trees and

$$T_o(A) = \bigcup_{\substack{m \geq 1, \\ 1 \leq i_j \leq k}} o(s_{i_1}, s_{i_2}, \ldots, s_{i_m}) \quad ,$$

$$T_o(B) = \bigcup_{\substack{m \geq 1, \\ 1 \leq i_j \leq k}} o(t_{i_1}, t_{i_2}, \ldots, t_{i_m}) \quad .$$

Then it is decidable whether or not

$$T_o(A) \cap T_o(B) = \emptyset \quad .$$

### Proof

We will show Theorem 1 first for the case of lists of ordered trees. Let $N$ be the set of node-labels and $E$ be the set of edge-labels occurring in the trees of the lists $A$ and $B$. We will construct two parenthesis languages which give us a string-representation of all the trees which can be generated by the lists $A$ and $B$. For the definition of parenthesis grammars the reader is referred to [3] rather than to the original definition found in [4].

Let ( , ), # be new symbols, not occurring as labels in the trees of $A$ and $B$ and furthermore for each $a \in N$ let $S_a, S_a'$ be new symbols. For a symbol $Z$ and a tree $T$ we define now a set of context free productions recursively as follows:

a) Let $T$ be a tree with root labeled by $a$ and $n \geq 1$ edges leaving the root, labeled by $b_1,\dots,b_n$, leading to the subtrees $T_1,\dots,T_n$.

$$P(Z,T,) = \{Z \to (a\#b_1\#X_1\#\dots\#b_n\#X_n) \mid X_i = S_c \text{ if } T_i \text{ consists of only one node labeled } c \text{ and } X_i \text{ is a new symbol } Z_i \text{ otherwise}\} \cup \bigcup_{|T_i|>1} P(Z_i,T_i)$$

, where for a tree $t$, the number of nodes of $t$ is denoted by $|t|$.

b) Let $T$ be a tree consisting of only one node labeled $a$, then

$$P(Z,T) = \{S_a \to (a)\} \quad .$$

For the motivation of this construction c.f. the left-bracketed representation of a tree given in [1].

According to the lists of trees

$$A = (s_1,s_2,\dots,s_k),\ B = (t_1,t_2,\dots,t_k)$$

we define production-sets $P_A$ and $P_B$ by:

$$P_A = \bigcup_{i=1}^{k} P(S'_{a_i},s_i) \cup \bigcup_{i=1}^{k} P(S_{a_i},s_i) \quad ,$$

$$P_B = \bigcup_{i=1}^{k} P(S'_{b_i},t_i) \cup \bigcup_{i=1}^{k} P(S_{b_i},t_i)$$

where $a_i$, $b_i$ are the node-labels of the roots of $s_i$, $t_i$ resp. .

The two parenthesis grammars $G_A$, $G_B$ are now defined by

$$G_A = (\Sigma,V_A,\sigma,P_A)$$

$$G_B = (\Sigma,V_B,\sigma,P_B) \text{ where}$$

$$\Sigma = E \cup N \cup \{(\ ,\ ),\#\}\ ,$$

$$\sigma = \{S'_a \mid a \in N\} \text{ and}$$

$V_A$, $V_B$ contain $\Sigma$, $\sigma$, $\{S_a \mid a \in N\}$ and all the new symbols introduced during the construction of $P_A$, $P_B$ resp. . Obviously, each terminal word generated by $G_A$, $G_B$ represents a concatenation of trees from A, B resp. and vice versa. Thus, there exist $i_1,\ldots,i_m$ and $j_1,\ldots,j_n$ such that

$$o(s_{i_1},\ldots,s_{i_m}) \cap o(t_{j_1},\ldots,t_{j_n}) \neq \emptyset$$

iff $L(G_A) \cap L(G_B) \neq \emptyset$ .

Since parenthesis languages are effectively closed under intersection [3], and the emptiness of parenthesis languages is trivially decidable, this proves our Theorem 1 for the case of ordered labeled trees.

For A, B being lists of oriented labeled trees we only have to modify the definition of $P(Z,T)$ to

$$P(Z,T) = \{Z \to (a\#b_{i_1}\#X_{i_1}\#\ldots\#b_{i_n}\#X_{i_n}) \mid$$

$X_{i_j} = S_c$ if $T_{i_j}$ consists of only one node labeled c and $X_{i_j}$ is a new symbol $Z_{i_j}$ otherwise, $(i_1,\ldots,i_n)$ is a permutation of $(1,\ldots,n)\} \cup \bigcup\limits_{|T_i|>1} P(Z_i,T_i)$ .

It should be clear that the above proof for the ordered trees can be carried over for the oriented trees without further changes. □

Corollary 1

Let $A = (s_1,s_2,\ldots,s_k)$, $B = (t_1,t_2,\ldots,t_k)$ be two lists of labeled ordered (oriented) trees and

$$T_{\uparrow}(A) = \bigcup_{\substack{m\geq 1,\\ 1\leq i_j\leq k}} \uparrow(s_{i_1},s_{i_2},\ldots,s_{i_m})$$

$$T_{\uparrow}(B) = \bigcup_{\substack{m\geq 1,\\ 1\leq i_j\leq k}} \uparrow(t_{i_1},t_{i_2},\ldots,t_{i_m}) \quad .$$

Then it is decidable whether or not

$$T_{\uparrow}(A) \cap T_{\uparrow}(B) = \emptyset \quad .$$

The proof of this corollary follows the lines of the proof of Theorem 1.

It is slightly more complicated in the construction of $P_A$ and $P_B$, where the parts $P(S_{a_i},s_i)$, $P(S_{b_i},t_i)$ have to be modified according to the insertion of a new labeled edge for the concatenation of the trees. But the details of the changes are obvious and left to the reader.

Corollary 2

Let $A = (s_1,s_2,\ldots,s_k)$, $B = (t_1,t_2,\ldots,t_k)$ be two lists of labeled ordered (oriented) trees and let $T_o(A)$, $T_o(B)$, $T_{\uparrow}(A)$, $T_{\uparrow}(B)$ be defined as in Theorem 1, Corollary 1 resp. . Then it is decidable whether or not

$$T_o(A) = T_o(B) \ ,$$
$$T_{\uparrow}(A) = T_{\uparrow}(B) \ .$$

This follows immediately from the theorems given in [3].

We turn now to our main results, namely, the undecidability results for the concatenation of unlabeled ordered and unlabeled oriented trees, our Tree Correspondence Problem. Since labeled trees are generalizations of the unlabeled ones, the corresponding results for the labeled case are hereby implied.

Theorem 2

Let $U = (s_1,\dots,s_\ell)$, $V = (t_1,\dots,t_\ell)$ be two lists of unlabeled ordered trees whose nodes have out-degrees of at most 2.

Then it is undecidable whether or not there exists a sequence of integers $i_1, i_2, \dots, i_m$ such that

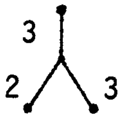
$$\circ(s_{i_1},\dots,s_{i_m}) \cap \circ(t_{i_1},\dots,t_{i_m}) \neq \emptyset \quad .$$

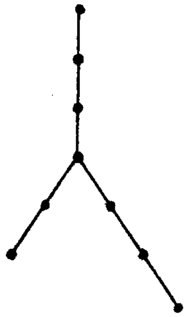
Proof

We will show the undecidability by encoding each instance A, B of the Post Correspondence Problem in two lists U, V of unlabeled trees such that A, B has a solution if and only there exists a solution for U, V.

We can restrict ourselves to instances A, B of the PCP over a two-letter alphabet $\{a,b\}$ by using the mapping $a_i \to ab^i$; which also has the effect that each word in the lists A, B starts with an a.

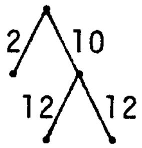
To simplify the representation of the trees we will only draw roots, nodes of out-degree 2 and leaves and mark their connections by the number of edges occurring in the original tree.

So  is an abbreviation for the tree

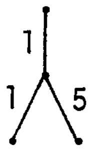


Now, let $\tau_1$, $\tau_2$ be two homomorphisms defined on $\{a,b\}^*$ by

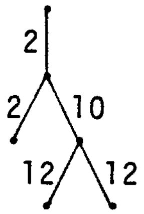
$\tau_1(a) =$



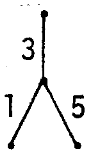
, $\tau_2(a) =$



$\tau_1(b) =$



, $\tau_2(b) =$



,

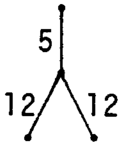
and for words $u,v \in \{a,b\}^*$ let

$$\tau_1(uv) = \underset{\ell m}{\circ} (\tau_1(u),\tau_1(v))$$

$$\tau_2(uv) = \underset{\ell m}{\circ} (\tau_2(u),\tau_2(v)) .$$

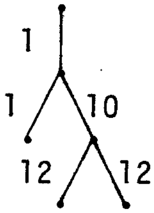
Note that $\tau_2(a)$, $\tau_2(b)$ can be completed by

$\rho =$

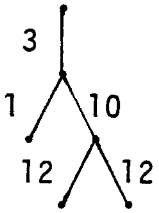


at their righthand leaves to become

$\tau_2'(a) =$



and $\tau_2'(b) =$



.

Hence, by $\tau_2'(uv) = \underset{\ell m}{\circ}(\tau_2'(u),\tau_2'(v))$ for $u,v \in \{a,b\}^*$ we get

$$\underset{\ell m}{\circ}\,(r{+}1\Big|,\tau_1(x),s\Big|) = \underset{\ell m}{\circ}\,(r\Big|,\tau_2'(x'),s{+}1\Big|)$$

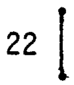
for any $r,s \geq 0$ if and only if $x = x'$, which is the essential idea of our following construction of U and V.

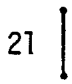
For the lists $A = (x_1,\dots,x_k)$, $B = (y_1,\dots,y_k)$ of nonempty strings over $\{a,b\}$ let

$$U = (\underset{\ell m}{\circ}(\alpha_1,\tau_1(x_1)),\dots,\underset{\ell m}{\circ}(\alpha_1,\tau_1(x_k)),\ \tau_1(x_1),\dots,\tau_1(x_k),\sigma_1,\mu_1,\mu_1,\omega_1),$$

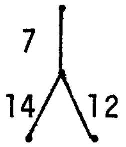
$$V = (\underset{\ell m}{\circ}(\alpha_2,\tau_2(y_1)),\dots,\underset{\ell m}{\circ}(\alpha_2,\tau_2(y_k)),\ \tau_2(y_1),\dots,\tau_2(y_k),\sigma_2,\rho,\mu_2,\omega_2)\ ,$$

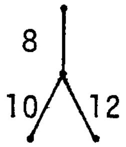
where the pairs $(\alpha_1,\alpha_2)$, $(\sigma_1,\sigma_2)$, $(\mu_1,\rho)$, $(\mu_1,\mu_2)$, $(\omega_1,\omega_2)$ are defined as follows:

$\alpha_1 =$ 

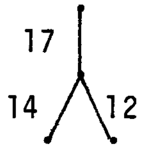
, $\alpha_2 =$ 

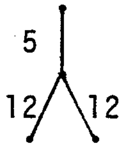
,

$\sigma_1 =$ 

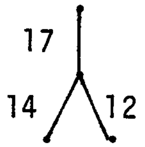
, $\sigma_2 =$ 

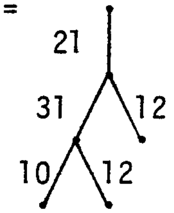
,

$\mu_1 =$ 

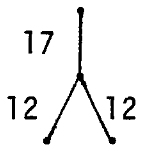
, $\rho =$ 

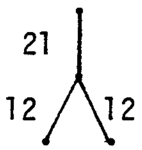
,

$\mu_1 =$ 

, $\mu_2 =$ 

,

$\omega_1 =$ 

, $\omega_2 =$ 

.

Now we assume, that for A, B there exists a sequence $i_1,i_2,\dots,i_m$ such that

$$x_{i_1} x_{i_2} \cdots x_{i_m} = y_{i_1} y_{i_2} \cdots y_{i_m} = z$$

and let $n$ be the length of $z$.

Then it is fairly easy to see that the sequence

$$i_1,(k+i_2),\dots,(k+i_m),(2k+1),\underbrace{(2k+2),\dots,(2k+2)}_{n\ \text{times}},\underbrace{(2k+3),\dots,(2k+3)}_{n\ \text{times}},(2k+4)$$

gives a solution for $U$ and $V$, just by using the facts

$$\underset{\ell m}{\circ}(22 \,\Big|\, ,\tau_1(z),7 \,\Big|\,) = \underset{\ell m}{\circ}(21 \,\Big|\, ,\tau_2'(z),8 \,\Big|\,) \quad \text{and}$$

$$\underset{\ell m}{\circ}(14 \bigwedge 12 \overbrace{,\mu_1,\dots,\mu_1}^{2n\ \text{times}},\omega_1) = \underset{\ell m}{\circ}(10 \bigwedge 12 \overbrace{,\mu_2,\dots,\mu_2}^{n\ \text{times}},\omega_2) \quad .$$

For the converse let us assume now, that for the lists of trees

$$U = (s_1,s_2,\dots,s_\ell) = (\underset{\ell m}{\circ}(\alpha_1,\tau_1(x_1)),\dots,\omega_1),$$
$$V = (t_1,t_2,\dots,t_\ell) = (\underset{\ell m}{\circ}(\alpha_2,\tau_2(y_1)),\dots,\omega_2)$$

there exists a sequence of indices $j_1,j_2,\dots,j_q$ such that $1 \le j_r \le 2k+4 = \ell$ for $r = 1,\dots,q$ and

$$\circ(s_{j_1},s_{j_2},\dots,s_{j_q}) \cap \circ(t_{j_1},t_{j_2},\dots,t_{j_q}) \ni T.$$

From the construction of the pairs $(s_i,t_i)$ it is clear that $j_1 \in \{1,\dots,k\}$, i.e.

$$(s_{j_1},t_{j_1}) = (\underset{\ell m}{\circ}(\alpha_1,\tau_1(x_{j_1})),\underset{\ell m}{\circ}(\alpha_2,\tau_2(y_{j_1})))$$

since those are the only pairs with matching lengths from the root to the first node with out-degree 2. Remember that all words $x_i,y_i$ start with an $a$. For the concatenation of further trees it is now essential which numbers of edges between two nodes of out-degree 2 can occur in a solution T.

For U, V let D(U), D(V) be the sets of numbers of edges between two nodes of out-degree 2, such that all nodes in between have only out-degree 1, occurring in the concatenations of the trees of U, V resp. . In other words, D(U), D(V) are the sets of chain-lengths occurring between inner nodes in $T_o(U), T_o(V)$.

Thus, $D(U) = \{2,4,9,10,12,14,19,21,24,29,31,34,36\}$, $D(V) = \{2,4,6,9,10,11,13,15,17,18,20,22,23,26,27,31,32,33,34\}$.

Of course, only numbers from $D(U) \cap D(V) = \{2,4,9,10,31,34\}$ can appear as such distances in a solution T.

Let us consider now the possible choices for $(s_{j_2}, t_{j_2})$:

(1) $j_2 \notin \{1,\dots,k\}$ since each tree in $\circ(\circ_{\ell m}(\alpha_2,\tau_2(y_{j_1})),\circ_{\ell m}(\alpha_2,\tau_2(y_{j_2})))$ would contain a path of 23 or 27 edges between two nodes of out-degree 2.

(2) $j_2 \notin \{2k+2,2k+3,2k+4\}$, since each tree in $\circ(\circ_{\ell m}(\alpha_1,\tau_1(x_{j_1})),\mu_1)$ and in $\circ(\circ_{\ell m}(\alpha_1,\tau_1(x_{j_1})),\omega_1)$ contains a path of 19 or 29 edges between two nodes of out-degree 2.

(3) Now assume $j_2 \in \{k+1,\dots,2k\}$. Then $s_{j_2} = \tau_1(x_{j_2-k})$, $t_{j_2} = \tau_2(y_{j_2-k})$ must be attached to the leftmost leaves of $\circ_{\ell m}(\alpha_1,\tau(x_{j_1})),\circ_{\ell m}(\alpha_2,\tau_2(y_{j_1}))$ resp. since all other concatenations would result in path-lengths 12, 6 resp. which are not in $D(U) \cap D(V)$.

By definition of $\tau_1,\tau_2$ it holds

$\underset{\ell m}{\circ}(\alpha_1,\tau_1(x_{j_1}),\tau_1(x_{j_2-k})) = \underset{\ell m}{\circ}(\alpha_1,\tau_1(x_{j_1},x_{j_2-k}))$ and

$\underset{\ell m}{\circ}(\alpha_2,\tau_2(y_{j_1},\tau_2(y_{j_2-k})) = \underset{\ell m}{\circ}(\alpha_2,\tau_2(y_{j_1},y_{j_2-k}))$ .

For the trees

$\underset{\ell m}{\circ}(\alpha_1,\tau_1(x_{j_1},x_{j_2-k})), \underset{\ell m}{\circ}(\alpha_2,\tau_2(y_{j_1},y_{j_2-k}))$

the arguments of (1) and (2) apply, i.e.

$j_3 \notin \{1,\dots,k\} \cup \{2k+2,2k+3,2k+4\}$, and if

$s_{j_3} = \tau_1(x_{j_3-k})$, $t_{j_3} = \tau_1(y_{j_3-k})$ then these trees must be attached to the leftmost leaves of $\underset{\ell m}{\circ}(s_{j_1},s_{j_2})$, $\underset{\ell m}{\circ}(t_{j_1},t_{j_2})$, etc. .

Thus, there exists a $p \geq 1$ such that $j_2,j_3,\dots,j_p$ are in $\{k+1,\dots,2k\}$. Furthermore, $p < q$ by the definitions of $\tau_1(x_i)$, $\tau_2(y_i)$ and by (1) and (2) $j_{p+1} = 2k+1$.

(4) Again, $s_{j_{p+1}} = \sigma_1$, $t_{j_{p+1}} = \sigma_2$ can only be attached to the leftmost leaves of

$\underset{\ell m}{\circ}(\alpha_1,\tau_1(x_{j_1},x_{j_2-k},\dots,x_{j_p-k}))$ ,

$\underset{\ell m}{\circ}(\alpha_2,\tau_2(y_{j_1},y_{j_2-k},\dots,y_{j_p-k}))$ resp. .

Therefore,

$\underset{\ell m}{\circ}(\alpha_1,\tau_1(x_{j_1},x_{j_2-k},\dots,x_{j_p-k}),\sigma_1)$ and

$\underset{\ell m}{\circ}(\alpha_2,\tau_2(y_{j_1},y_{j_2-k},\dots,y_{j_p-k}),\sigma_2)$

are both initial parts of the solution T.

Since the path-length 9 is now occurring for the first time between two nodes of out-degree 2 on both leftmost paths from the roots to the leaves, there must be a complete matching of all previous path-lengths between out-degree 2 nodes on these paths.

After the necessary completion of $\tau_2(y_{j_1}, y_{j_2-k}, \ldots, y_{j_p-k})$ which can only be achieved by concatenations with $\rho$, we have

$$\circ_{\ell m}(\alpha_1, \tau_1(x_{j_1}, x_{j_2-k}, \ldots, x_{j_p-k}), 7 \;\Big|\; ) =$$

$$\circ_{\ell m}(\alpha_2, \tau_2(y_{j_1}, y_{j_2-k}, \ldots, y_{j_p-k}), 8 \;\Big|\; )$$

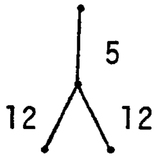
as completely matching initial parts of T. We conclude

$$x_{j_1}\ x_{j_2-k} \cdots x_{j_p-k} = y_{j_1}\ y_{j_2-k} \cdots y_{j_p-k}$$

which gives obviously a solution of the instance A, B of the PCP and thus complete the proof of Theorem 2. □

Note, that the inclusion of subtree

$\rho =$



in $\tau_1(a)$ and $\tau_1(b)$, and the pair $(\mu_1, \rho)$ in the simulating instance of the Tree Correspondence Problem is essential for our proof. Otherwise the instance of the Tree Correspondence Problem might have a solution which does not correspond to a solution of the simulated instance of PCP. Intuitively, we want to enforce that a "solution" on the left-most path is completed before another "solution" is started elsewhere.

The restriction to trees with nodes of out-degree at most one in Theorem 2 would make the problem decidable since it would, clearly, correspond to PCP over a one-letter alphabet.

Corollary 3

Let $U = (s_1,\ldots,s_\ell)$, $V = (t_1,\ldots,t_\ell)$ be two lists of unlabeled ordered (oriented) trees whose nodes have out-degrees of at most 2.

Then it is undecidable whether or not there exists a sequence of integers $i_1, i_2, \ldots, i_m$ such that

a) $\circ(s_{i_1},\ldots,s_{i_m}) \cap \circ(t_{i_1},\ldots,t_{i_m}) \neq \emptyset$

b) $\uparrow(s_{i_1},\ldots,s_{i_m}) \cap \uparrow(t_{i_1},\ldots,t_{i_m}) \neq \emptyset$ .

Proof

a) For the ordered trees this is the result of Theorem 2. For the oriented trees it is easy to see that the definition of the leftmost concatenation can be formulated equivalently for the trees occurring in U and V in terms of the admissible path-lengths of $D(U) \cap D(V)$. This property is of course independent of the relative order of subtrees and thus can be used for oriented trees as well as for ordered ones.

b) Since the operation $\uparrow$ always creates a new edge, we can carry over the proof of Theorem 2 if we redefine the trees of U and V in the following way. In each of these trees insert one new node between each pair of adjacent nodes, except if one of these nodes is a leaf where there is nothing changed. This just doubles each number in $D(U)$ and $D(V)$ but does not affect the proof in other ways. □

## 4. Concluding Remarks

Since trees are intensively studied objects and there are a lot of tree-families investigated which are sub-families of the ones treated in Theorem 2 and Corollary 4, it would be an interesting problem to find nontrivial families for which the question given in Theorem 2 turns out to be decidable. On the other hand, one can try to put severe restrictions on the trees of U and V such that undecidability still holds and by this narrow the gap between decidability and undecidability.

## References

1. A.V. Aho, J.D. Ullman, The Theory of Parsing, Translating and Compiling, Vol. 1, Prentice Hall, 1972.
2. M.A. Harrison, Introduction to Formal Languages, Addison Wesley, 1978.
3. D.E. Knuth, A Characterization of Parenthesis Languages, Information and Control, 11, pp. 269-289, 1968.
4. R. McNaughton, Parenthesis Grammars, J. Assoc. Computing Mach., 14, pp. 490-500, 1967.