Decidability results concerning tree transducers I

By Z. Ésik

A tree transducer is called functional if its induced transformation is a partial mapping. We show that the functionality of tree transducers is decidable. Consequently, the equivalence problem for deterministic tree transducers is also decidable. The latter result was independently achieved by Z. Zachar in [12] for bottom-up tree transducers and a restricted class of top-down tree transducers. The solvability of the equivalence problem for generalized deterministic sequential machines is known from [2] and [4]. It was proved in [11] that this positive result can not be generalized for arbitrary, i.e. generalized nondeterministic, sequential machines. Therefore, the equivalence problem for nondeterministic tree transducers is undecidable.

Our result can be used to minimize deterministic tree transducers in an effective manner. However, the minimal realizations of a deterministic tree transducer are not isomorphic. We investigate conditions assuring the uniqueness (up to isomorphism) of minimal realizations in certain classes of tree transducers.

Part of the results of the present paper have been announced in [8]. The terminology is used in the sense of [5].

1. Notions and notations

By a type $F= \bigcup_{n=0}^{\infty} F_n$ we mean a finite type such that $F_0 \neq \emptyset$. For the type $F$, $v(F)=\max \{n \mid F_n \neq \emptyset\}$. An $F$-algebra is a system $A=(A, ((f)_A \mid f \in F))$, or shortly, $(A, F)$, where for every nonnegative integer $n$ and $f \in F_n (f)_A : A^n \to A$ is the realization of the $n$-ary operational symbol $f$.

Let $Y$ be an arbitrary set. Then $T_{F,Y}= (T_{F,Y}, F)$ denotes the free $F$-algebra generated by $Y$. The elements of $T_{F,Y}$ are called trees and they can be obtained by induction as follows: $T_{F,Y}$ is the smallest set satisfying

(i) $F_0, Y \subseteq T_{F,Y},$

(ii) if $n > 0$, $f \in F_n$, $t_1, \ldots, t_n \in T_{F,Y}$ then $f(p_1, \ldots, p_n) \in T_{F,Y}.$
In particular, if $Y=X_n$, the set of the first $n$ variables $x_1, \ldots, x_n$ for a nonnegative integer $n$, $T_{F,Y}$ is denoted by $T_{F,n}$ and $T_{F,0}$ is written $T_F$. Each $n$-ary tree $p \in T_{F,n}$ induces a mapping $(p)_A: A^n \to A$ in an $A$-algebra $A$. If $A$ is the free algebra $T_{F, Y}$ then $(p)_A(t_1, \ldots, t_n) = p(t_1, \ldots, t_n)$, i.e. the tree obtained by substituting $t_i$ for $x_i$ ($i=1, \ldots, n$) in $p$.

The depth (dp), rank (rn) and frontier (fr) of trees are defined as usually. For a tree $p \in T_{F,Y}$ we have

(i) $dp(p) = 0$, $rn(p) = 1$, $fr(p) = p$ if $p \in Y$,
(ii) $dp(p) = 0$, $rn(p) = 1$, $fr(p) = \lambda$ if $p \in F_0$,
(iii) $dp(p) = 1 + \max \{ dp(p_i) | i = 1, \ldots, n \}$, $rn(p) = 1 + \sum_{i=1}^{n} rn(p_i)$,

$fr(p) = fr(p_1) \ldots fr(p_n)$ if $p = f(p_1, \ldots, p_n)$, $p \in F_n$,

$p_1, \ldots, p_n \in T_{F,Y}$ and $n > 0$. Here $\lambda$ denotes the empty string.

In connection with the elements of $T_{F,n}(n \geq 0)$ we shall also use the concept of path. For an arbitrary $i (1 \leq i \leq n)$ and $p \in T_{F,n}$ path$_i(p)$ is given by

(i) $\text{path}_i(p) = \{ \lambda \}$ if $p = x_i$,
(ii) $\text{path}_i(p) = \emptyset$ if $p \in F_0 \cup X_n - \{ x_i \}$,
(iii) $\text{path}_i(p) = \{ jw | w \in \text{path}_i(p), 1 \leq j \leq m \}$ if $p = f(p_1, \ldots, p_m)$,

$m > 0$, $f \in F_m$, $p_1, \ldots, p_m \in T_{F,n}$. If path$_i(p)$ is a singleton then it is identified with its unique element. For $w \in \text{path}_i(p)$ we denote by $|w|$ the length of $w$. path$(p) = \bigcup_{i=1}^{n} \text{path}_i(p)$. For arbitrary two strings $v$ and $w$ $v/w$ denotes the derivative of $v$ with respect to $w$, i.e. $v/w = u$ if and only if $v = uw$.

Further on we shall often use vector notations to simplify the treatment. Vectors, except possibly the one dimensional ones, are always denoted by boldfaced letters. For each $k$ dimensional vector $a \in A^k$ ($k \geq 0$) and $i$ ($1 \leq i \leq k$) $a_i$ denotes the $i$th component of $a$. Conversely, if $a \in A$ then $a^i \in A^k$ is the $k$ dimensional vector whose each component is equal to $a$. The product ab of the $k$ dimensional vectors $a$ and $b$ is defined by $ab = (a_1 b_1, \ldots, a_k b_k)$ where $a_i b_i$ are short notations for $(a_i, b_j)$ ($i=1, \ldots, k$). For the vectors of trees $p \in T_{F,n}$ and $q \in T_{F,m}$ we denote by $p(q)$ the vector $(p_1(q), \ldots, p_k(q))$.

According to the function fr one can distinguish the subset $T_{F,n}$ of $T_{F,F}$. This consists of those elements of $T_{F,n}$ whose frontier is a permutation of the variables in $X_n$. We may extend this definition to vectors as follows: $T_{F,k}^r = \{ p \in T_{F,n}^r | fr(p_1) \ldots fr(p_i) \text{ is a permutation of } X_n \}$. Observe that $T_{F,k}^r$ is not the $k$th power of $T_{F,F}$.

We now turn to the definition of tree transducers. Following [5] a top-down tree transducer is a system $A = (F, A, G, A_0, \Sigma)$, where $F$ and $G$ are types, $A$ is a finite, nonvoid set, the set of states, $A_0 \subseteq A$ is the set of initial states, finally, $\Sigma$ is a finite set of top-down rewriting rules. A top-down rule has the form $af \rightarrow p$ — or equivalently $af(x_1, \ldots, x_n) \rightarrow p$, where $n \geq 0$, $a \in A$, $f \in F_n$, $p \in T_{G,A \times X_n}$. A bottom-up tree transducer $A = (F, A, G, A_0, \Sigma)$ has a similar structure except $A_0$ is the set of final states and $\Sigma$ contains bottom-up rewriting rules. A typical
bottom-up rewriting rule is of form \( f(a_1 x_1, \ldots, a_n x_n) \rightarrow a p' \) where \( n \geq 0, f \in F_n, p \in T_{G,n}, a, a_1, \ldots, a_n \in A \). By a tree transducer we mean a top-down or bottom-up transducer.

Take an arbitrary tree transducer \( A = (F, A, G, A_0, \Sigma) \) and let \( \Sigma \) be an arbitrary set. \( \Sigma \) can be used to define a binary relation \( \Rightarrow_{A,Y} \) on \( T_{G,A} \times T_{F,Y} \) in the top-down case and on the set \( \mathcal{T}_{G,A} \times T_{F,Y} \) in the bottom-up case. It is called derivation and its exact definition can be found in [5]. If there is no danger of confusion \( A' \) is omitted in \( \Rightarrow \).

It can be seen that if \( Y_1 \subseteq Y_2 \) and \( p, q \in T_{G,A} \times T_{F,Y_1} \) then \( p \Rightarrow_{Y_1} q \) if and only if \( p \Rightarrow_{Y_2} q \). Similar equivalence is valid in the bottom-up case. Thus we may omit \( Y \) in \( \Rightarrow \).

Again take the tree transducer \( A \). This induces a transformation \( \tau_A \subseteq T_F \times T_G \): 

\[
\tau_A = \{(p, q) \mid \exists a_0 \in A_0 \ a_0 p \Rightarrow q\}
\]
in the top-down case, and

\[
\tau_A = \{(p, q) \mid \exists a_0 \in A_0 \ p \Rightarrow a_0 q\}
\]
for bottom-up A. If \( \tau_A \) is a (partial) function \( A \) is called functional. This is always the case if \( A \) is deterministic, i.e. different rules have different left sides, moreover, \( A_0 \) is a singleton in the top-down case. Two tree transducers are called equivalent if their induced transformations coincide. For a tree transducer \( A = (F, A, G, A_0, \Sigma) \) and a state \( a \in A \) we denote by \( A(a) \) the transducer \( A(a) = (F, A, G, \{a\}, \Sigma) \).

The domain of the transformation \( \tau_A \) is denoted by \( \text{dom} \ \tau_A \). It is a regular subset of \( T_F \), i.e. a regular forest. Regular forests are exactly the forests recognized by tree automata. A tree automaton is a system \( B = (F, B, A_0) \) with \( (B, (F)) \) a finite \( F \)-algebra which is denoted by \( B \) too, \( A_0 \subseteq B \) is the set of final states. The forest recognized by \( B \) is determined by \( T(B) = \{p \in T_F \mid (p)_0 \in B_0\} \).

Sometimes we need to restrict a top-down tree transducer to a regular forest. If \( A = (F, A, G, A_0, \Sigma) \) is a top-down tree transducer and \( T \subseteq T_F \) is a regular forest then the system \( B = (F, T, A, G, A_0, \Sigma) \) is called a regularly restricted top-down tree transducer. Its induced transformation is \( \tau_B = \{(p, q) \in \tau_A \mid p \in T\} \). A similar but more general concept is the concept of top-down tree transducer with regular look-ahead introduced in [6]. A top-down tree transducer with regular look-ahead is a system \( A = (F, A, G, A_0, \Sigma) \) where \( F, A, G, A_0 \) are the same as for top-down tree transducers and \( \Sigma \) is a finite set of rules

\[
(af(x_1, \ldots, x_n) \rightarrow p; R_1, \ldots, R_n)
\]
where \( af(x_1, \ldots, x_n) \rightarrow p \) is a top-down rewriting rule, i.e. \( a \in A, f \in F_n \ (n \geq 0), p \in T_{G,A} \times T_n, \) and \( R_i \subseteq T_F \ (1 \leq i \leq n) \) are regular forests. The regular forests \( R_i \) are used to restrict the applicability of the corresponding top-down rule \( af(x_1, \ldots, x_n) \rightarrow p \). The rule \( (af(x_1, \ldots, x_n) \rightarrow p; R_1, \ldots, R_n) \) can be applied for a subtree of a tree in \( T_{G,A} \times T_{F,Y} \) if and only if it is of form \( af(p_1, \ldots, p_n) \) with \( p_i \in R_i \) for each \( i \ (1 \leq i \leq n) \). Apart from this derivation is defined as for top-down transducers. The induced transformation is the relation \( \tau_A = \{(p, q) \mid a_0 p \Rightarrow q \} \) for some \( a_0 \in A_0 \). Again, if it is a function \( A \) is called functional. It is known that every functional bottom-up or top-down tree transducer is equivalent to some deterministic top-down transducer with regular look-ahead (cf. [7]).
2. The decidability of functionality of tree transducers

First we show that the decision of functionality of bottom-up transducers is reducible to the decision of functionality of regularly restricted top-down ones.

Let \( A = (F, A, G, A_0, \Sigma) \) be an arbitrary bottom-up transducer. Define the top-down transducer with regular look-ahead \( A' \) as follows: \( A' = (F, A, G, A_0, \Sigma') \) where

\[
\Sigma' = \{ (af \rightarrow p(a_1 x_1, \ldots, a_n x_n); R_1, \ldots, R_n) \mid f(a_1 x_1, \ldots, a_n x_n) \rightarrow ap \in \Sigma, \\
R_i = \text{dom } \tau_{A(a_i)} (i = 1, \ldots, n) \}
\]

Lemma 1. \( A \) is functional if and only if \( A' \) is functional.

Proof. It is obvious that \( \tau_A \subseteq \tau_{A'} \). Therefore if \( A' \) is functional then \( A \) is functional, too. To prove the converse first we show that if \( ap \Rightarrow a'q \) and \( a'p \Rightarrow a'q' \) where \( a, a' \in A, p \in T_F, q, q' \in T_G \) and \( q \neq q' \) then there exist different trees \( r, r' \in T_G \) such that \( p \Rightarrow a'br \) and \( p \Rightarrow a'b'r' \) are also satisfied for certain choice of states \( b, b' \) with \( \{ b, b' \} \subseteq \{ a, a' \} \). We shall prove this by induction on \( p \). The basis, \( p \in F_0 \), is immediate. Suppose now that \( p = f(p_1, \ldots, p_n) \) where \( n > 0 \), \( f \in F_n, p_1, \ldots, p_n \in T_F \). Since \( ap \Rightarrow q \) and \( a'p \Rightarrow q' \) there exist rules \( f(a_1 x_1, \ldots, a_n x_n) \rightarrow aq_0 \), \( f(a'_1 x_1, \ldots, a'_n x_n) \rightarrow a'q'_0 \in \Sigma \) with \( p_i \in \text{dom } \tau_{A(a_i)} \cap \text{dom } \tau_{A(a'_i)} \) and satisfying \( q_0(a_1 p_1, \ldots, a_n p_n) \Rightarrow q \) and \( q'_0(a'_1 p_1, \ldots, a'_n p_n) \Rightarrow q' \), respectively. We distinguish two cases.

Firstly assume that for each \( i \in \{ 1, \ldots, n \} \) if \( x_i \) appears in \( \text{fr} (q_0) \) then there exists exactly one tree \( q_i \in T_G \) with \( a_i p_i \Rightarrow q_i \). Then also, \( p_i \Rightarrow a_i q_i \). This and \( p_i \in \text{dom } \tau_{A(a_i)} \) \( (i = 1, \ldots, n) \) yield \( p \Rightarrow a'q' \). Similarly, we get \( p \Rightarrow a'q' \) if, for each \( x_i \) occuring in \( \text{fr} (q'_0) \), there is only one tree in \( T_G \) which can be derived from \( a'_i p_i \). This proves our assertion in the first case.

Secondly assume that there is an integer \( i \in \{ 1, \ldots, n \} \) such that \( x_i \) appears in \( \text{fr} (q_0) \) and there are different trees \( q_i, q'_i \in T_G \) with \( a_i p_i \Rightarrow q_i \) and \( a_i p_i \Rightarrow q'_i \), respectively. Then, by the induction hypothesis, there exist trees \( r_i \neq r'_i \in T_G \) satisfying both \( p_i \Rightarrow a_i r_i \) and \( p_i \Rightarrow a_i r'_i \). For each index \( j (j \neq i) \) choose \( r_j \in T_G \) in such a way that we have \( p_j \Rightarrow a_j r_j \). This can be done by \( p_i \in \text{dom } \tau_{A(a_j)} \). Now let \( r = q_0(r_1, \ldots, r_n), r' = q'_0(r_1, \ldots, r_{i-1}, r'_i, r_{i+1}, \ldots, r_n) \). \( r \neq r' \) because \( r_i \neq r'_i \). On the other hand \( p \Rightarrow ar \) and \( p \Rightarrow ar' \).

Now assume that \( A' \) is not functional. Then there exist trees \( p \in T_F, q \neq q' \in T_G \) and initial states \( a_0, a'_0 \in A_0 \) such that both \( a_0 p \Rightarrow a'q \) and \( a'_0 p \Rightarrow a' q' \) are satisfied. By the previous considerations it follows that there are different trees \( r, r' \in T_G \) with \( p \Rightarrow a_0 b_0 r \) and \( p \Rightarrow a'_0 b'_0 r' \) where each of the states \( b_0 \) and \( b'_0 \) denotes either \( a_0 \) or \( a'_0 \). This means that both \( (p, r) \) and \( (p, r') \) are in \( \tau_A \), i.e. \( A \) is not functional.

Lemma 2. The decision of functionality of bottom-up tree transducers is reducible to the decision of functionality of regularly restricted top-down ones.

Proof. Let \( A \) be an arbitrary bottom-up transducer and \( A' \) the top-down transducer with regular look-ahead constructed in the previous lemma. We know that \( A \) is functional if and only if \( A' \) is functional. By Theorem 2.6 in [6] we have
\( \tau_A = \tau \circ \tau_B \) where \( \tau \) is a deterministic bottom-up relabeling, i.e. a transformation induced by a special deterministic bottom-up transducer, and \( B \) is a top-down transducer. Since \( \tau \) is a function \( A' \) is functional if and only if \( B \) restricted to the regular forest \( \text{dom}(\tau_A) \) is functional. Note that \( \text{dom}(\tau_A) = \text{dom}(\tau_A) \). As one can construct the transducers \( A' \) and \( B \) in an effective manner this proves Lemma 2.

Now let us fix an arbitrary regularly restricted top-down tree transducer \( A = (F, T, A, G, A_0, \Sigma) \) and a tree automaton \( B = (F, B, B_0) \) recognizing \( T \). Set \( P = \{ p \in T \mid \exists q \neq q' \in T_G \ (p, q), (p, q') \in \tau_A \}. \)

In the next five lemmas we shall present five reduction rules. Each reduction rule produces a smaller tree \( p' \in P \) for a tree \( p \in T \) if it can be applied for \( p \).

**Lemma 3.** Let \( p_1, p_2 \in \hat{T}_{F,1}, p_3 \in T_F, n_1, n'_1, n_2, n'_2 \geq 0, q_1 \in \hat{T}_{G,n_1}, q'_1 \in \hat{T}_{G,n'_1}, q_2 \in \hat{T}_{G,n_2}, q'_2 \in \hat{T}_{G,n'_2}, q_3 \in T_{G,n}, q'_3 \in T_{G,n'}, a_0, a'_0 \in A, a \in A^n, a' \in A'^n \) \((i = 1, 2) \). Let us denote by \( A_1 \) and \( A'_1 \) the sets \( A_1 = \{ a_{i,j} \mid 1 \leq j \leq n_i \} \) and \( A'_1 = \{ a'_{i,j} \mid 1 \leq j \leq n'_i \} \) \((i = 1, 2) \) respectively. Assume that each of the following conditions is satisfied:

(i) \( p_1(p_2(p_3)) \in T \),

(ii) \( a_0 p_1 \Rightarrow q_1(a_1 x_1^n), \ a'_0 p_1 \Rightarrow q'_1(a_1 x_1^{n'_1}) \),

(iii) \( a_1 p_2 \Rightarrow q_2(a_2 x_2^n), \ a'_1 p_2 \Rightarrow q'_2(a_2 x_2^{n'_1}) \),

(iv) \( a_2 p_3 \Rightarrow q_3, \ a'_2 p_3 \Rightarrow q'_3 \),

(v) \( (p_3)_{B} = (p_2(p_3))_{B}, \ A_1 \subseteq A_2, \ A'_1 \subseteq A'_2 \),

(vi) \( q_1(r) \neq q'_1(r') \) holds for any \( r \in T_{G,n} \) and \( r' \in T_{G,n'} \).

Then \( p_1(p_2(p_3)) \in P \).

**Proof.** First note that our assumptions imply the condition \( p_1(p_2(p_3)) \in P \).

From now on let \([n]\) denote the set of the first \( n \) positive integers for every \( n \geq 0 \). Thus \([0]\) is the empty set. Let \( \varphi : [n_1] \rightarrow [n_2] \) and \( \varphi' : [n_1] \rightarrow [n'_2] \) be mappings with \( a_{i,1} = a_{i,\varphi(i)} \) \((i \in [n_1]) \) and \( a'_{i,1} = a'_{i,\varphi'(i)} \) \((i \in [n_2]) \). Obviously we have \( a_1 p_3 \Rightarrow r \) and \( a'_1 p_3 \Rightarrow r' \) where \( r = (q_3, \varphi(1), ..., q_3, \varphi(n_1)), r' = (q'_3, \varphi'(1), ..., q'_3, \varphi'(n_1)) \).

By (iv) this implies that \( a_0 p_1 \Rightarrow q_1(r) \) and \( a'_0 p_1 \Rightarrow q'_1(r') \). On the other hand \( q_1(r) \neq q'_1(r') \) by our assumption (vi). Furthermore, \( p_1(p_2(p_3)) \in P \) holds by (v). Hence \( p_1(p_2(p_3)) \in P \).

**Lemma 4.** Let \( p_1 \in \hat{T}_{F,1}, p_2 \in T_F, n, n' > 0, q_1 \in \hat{T}_{G,n}, q'_1 \in \hat{T}_{G,n'}, q_2 \in T_{G,n}, q'_2 \in T_{G,n'}, a_0, a'_0 \in A, a \in A^n, a' \in A'^{n'} \). Let \( |A| \) and \( |B| \) denote the cardinality of \( A \) and \( B \) respectively and let \( \|A\| = 2^{|A|}, K = \max \{ dp(q) \mid a \in A, p \in T_F, a \rightarrow q \in \Sigma \} \). Assume that the following conditions are valid:

(i) \( p_1(p_2) \in T \),

(iii) \( a_0 p_1 \Rightarrow q_1(a_1 x_1^n), \ a'_0 p_1 \Rightarrow q'_1(a_1 x_1^{n'_1}) \),
(iii) \( a^p \Rightarrow q \), \( a'^p \Rightarrow q' \), path\(_1\) \((q_1)\) is a prefix of path\(_1\) \((q_1')\),
\(|\text{path}\_1\( (q_1) \)| - |\text{path}\_1\( (q_1') \)| > \( \|A\|^2 |B|K \), dp \((p_2)\) \(\equiv\) \( \|A\|^2 |B| \).

Then there is a tree \( r \in T_F \) such that \( p_1(r) \in P \) and \( \text{rn}(r) < \text{rn}(p_2) \).

**Proof.** Let \( R \) be the forest defined by
\[ R = \{ r \in T \mid p_1(r) \in P, \text{rn}(r) < \text{rn}(p_2) \} \]
Since \( p_2 \in R \) \( R \) is nonvoid. Let \( r \) be an element of \( R \) with minimal rank. We shall show that \( p_1(r) \in P \) and \( \text{dp}(r) < \|A\|^2 |B| \).

Assume that the condition \( \text{dp}(r) < \|A\|^2 |B| \) does not hold. In this case there exist
\[ r_1, r_2 \in \hat{T}_{F,1}, r_3 \in T_F, m_1, m_1', m_2, m_2' \equiv 0, s_i, t_i \in \hat{T}_{G,m_i}, s_i', t_i' \in \hat{T}_{G,m'_i} \text{ such that each of the following five conditions is satisfied:} \]

1. \( r = r_1(r_2(r_3)) \), \( r_2 \neq x_1 \),
2. \( a r_1 \Rightarrow s_1(b_1 x_1^{m_1}) \), \( a' r_1 \Rightarrow s_1'(b_1' x_1^{m_1'}) \),
3. \( b_1 r_2 \Rightarrow s_2(b_2 x_1^{m_2}) \), \( b_1' r_2 \Rightarrow s_2'(b_2' x_1^{m_2'}) \),
4. \( b_2 r_3 \Rightarrow s_3 \), \( b_2' r_3 \Rightarrow s_3' \),
5. \( (r_2)_{B} = (r_2(r_3))_{B}, B_1 \subseteq B_2, B_1' \subseteq B_2', \text{ where} \)
\[ B_i = \{ b_i, j | 1 \leq j \leq m_i \}, B_i' = \{ b_i', j | 1 \leq j \leq m_i' \} \text{ (} i = 1, 2 \) \]

Now let \( \phi: [m_1] \rightarrow [m_2], \phi': [m_1'] \rightarrow [m_2'] \) be mappings satisfying the equalities \( b_{1,i} = b_{2,\phi(i)} \) \( (i \in [m_1]) \), \( b_{1,i} = b_{2,\phi'(i)} \) \( (i \in [m_1']) \). It is immediate that \( a r_1(r_2) \Rightarrow \Rightarrow s_1(s_3, \phi(1), \ldots, s_3, \phi(m_1)) \) and \( a' r_1(r_2) \Rightarrow \Rightarrow s_1'(s_3, \phi'(1), \ldots, s_3, \phi'(m_1)) \). This, together with \( (r_1(r_3))_{B} = (r)_{B} \) yields that \( r_1(r_3) \in R \), which is a contradiction because \( \text{rn}(r_1(r_3)) < \text{rn}(r) \).

Therefore, \( \text{dp}(r) < \|A\|^2 |B| \). This implies that for every \( s \in T_{G}^a \) and \( s' \in T_{G}^a \)
if the derivations \( a r^a \Rightarrow s \) and \( a' r^a \Rightarrow s' \) exist then \( \text{dp}(s), \text{dp}(s') \equiv \|A\|^2 |B|K \), thus, by (iv), \( p_1(r) \in P \). Since \( r \) was of minimal rank this ends the proof of Lemma 4.

**Lemma 5.** Let \( p_1, p_2, p_3 \in \hat{T}_{F,1}, p_4 \in T_F, n_1, n_2, n_3, n_4 \equiv 0 \) \( (i = 1, 2, 3) \), \( q_1 \in \hat{T}_{G,n_1+1}, q'_1 \in \hat{T}_{G,n'_1+1}, r_1 \in \hat{T}_{G,n_2}, r'_1 \in \hat{T}_{G,n'_2}, q_2 \in \hat{T}_{G,n_3}, q'_2 \in \hat{T}_{G,n'_3}, r_2 \in \hat{T}_{G,n_3}, q_3 \in \hat{T}_{G,n_3}, r_3 \in \hat{T}_{G,n_3}, q_4 \in \hat{T}_{G,n_3}, q'_4 \in \hat{T}_{G,n'_3}, r_4 \in T_{G,m_3}, q_0 \in A_0, a \in A, a' \in A', a'_0 \in A_0, b_i \in A^m \) \( (i = 1, 2, 3) \). Finally, let \( v \in T_G \) and \( v' = r_1(r_2(r_3(r_4))) \). Denote by \( A_i, A'_i \) and \( B_i \) \( (i = 1, 2, 3) \) the sets of components of \( a_i, a'_i \) and \( b_i \), respectively. Assume that the following conditions are satisfied:
Decidability results concerning tree transducers I.

(i) \( p_1(p_3(p_3(p_3))) \in \mathcal{T} \),

(ii) \( a_0 p_1 \Rightarrow q_1(ax_1, a_1 x_1^{n_1}), \quad a^*_0 p_1 \Rightarrow q'_1(r_1(b_1 x_1^{m_1}), a'_1 x_1^{n'_1}) \),

(iii) \( a_1 p_2 \Rightarrow q_2(a_2 x_2^{n_2}), \quad a^*_1 p_2 \Rightarrow q'_2(a_2 x_1^{n'_2}), \quad a p_2 \Rightarrow a x_1, \quad b_1 p_3^{m_3} \Rightarrow r_3(b_2 x_1^{m_3}) \),

(iv) \( a_2 p_3 \Rightarrow q_3(a_3 x_3^{n_3}), \quad a^*_2 p_3 \Rightarrow q'_3(a_3 x_1^{n'_3}), \quad a p_3 \Rightarrow a x_1, \quad b_2 p_4^{m_4} \Rightarrow r_3(b_3 x_1^{m_4}) \),

(v) \( a_3 p_4^{m_4} \Rightarrow q_4, \quad a^*_3 p_4^{m_4} \Rightarrow q'_4, \quad a p_4 \Rightarrow v, \quad b_3 p_4^{m_4} \Rightarrow r_4 \),

(vi) \( (p_3(p_4))_B = (p_2(p_3))_B = (p_1(p_2(p_3)))_B \),

\( A_1 \subseteq A_2 \subseteq A_3, A'_1 \subseteq A'_2 \subseteq A'_3, B_1 = B_2 \subseteq B_3 \),

(vii) \( v = v', \quad \text{path}_1(q_1) = \text{path}_1(q'_1) \).

Then at least one of the trees \( p_1(p_2(p_3)), p_1(p_3(p_4)) \) and \( p_1(p_4) \) is in \( P \).

**Proof.** First observe that by the assumptions of the lemma it follows that \( p_1(p_2(p_3(p_4))) \in P \).

Let \( \phi_i : [n] \to [n_{i+1}], \phi'_i : [n_i] \to [n_{i+1}] \) and \( \psi_i : [m] \to [m_{i+1}] \) \((i = 1, 2)\) be mappings such that we have \( a_{i,j} = a_{i+1,j+1}(j) \) \((i = 1, 2, j \in [n_i])\), \( a'_i,j = a'_{i+1,j+1}(j) \) \((i = 1, 2, j \in [n'_i])\), \( b_i,j = b_{i+1,j+1}(j) \) \((i = 1, 2, j \in [m_i])\). Furthermore, let \( \phi_3 = \phi_1 \circ \phi_2, \phi'_3 = \phi'_1 \circ \phi'_2, \psi_3 = \psi_1 \circ \psi_2 \).

Let us introduce the following notations:

\[
\begin{align*}
 s_1 &= (q_3, \phi_1(1), \ldots, q_3, \phi_1(m_3))(q_4), \\
 s'_1 &= (q_3, \phi'_1(1), \ldots, q_3, \phi'_1(n'_3))(q_4), \\
 t_1 &= (r_3, \psi_1(1), \ldots, r_3, \psi_1(m_1))(r_4), \\
 s_2 &= q_2(q_4, \psi_2(1), \ldots, q_4, \psi_2(n_2)), \\
 s'_2 &= q_2(q'_4, \psi'_2(1), \ldots, q'_4, \psi'_2(n'_2)), \\
 t_2 &= r_2(r_4, \psi_2(1), \ldots, r_4, \psi_2(m_2)), \\
 s_3 &= (q_4, \phi_2(1), \ldots, q_4, \phi_2(m_3)), \\
 s'_3 &= (q'_4, \phi'_2(1), \ldots, q'_4, \phi'_2(n'_2)), \\
 t_3 &= (r_4, \psi_2(1), \ldots, r_4, \psi_2(m_2)).
\end{align*}
\]

It is easy to check that each of the following derivations is valid:

\( a_0 p_1(p_3(p_4)) \Rightarrow \Rightarrow q_1(v, s_1), \quad a_0 p_1(p_3(p_4)) \Rightarrow q'_1(r_1(t_1), s'_1), \quad a_0 p_1(p_3(p_4)) \Rightarrow q_1(v, s_2), \quad a_0 p_1(p_3(p_4)) \Rightarrow q'_1(r_1(t_2), s'_2), \quad a_0 p_1(p_3(p_4)) \Rightarrow q_1(v, s_3), \quad a_0 p_1(p_3(p_4)) \Rightarrow q'_1(r_1(t_3), s'_3). \)

On the other hand \( p_1(p_2(p_3)), p_1(p_2(p_4)), p_1(p_3(p_4)) \in \mathcal{T} \).

Assume that \( p_1(p_2(p_3)) \notin P \). Then, by (vii), it follows that \( m_1, m_2, m_3 > 0 \) and there is an integer \( j \in [m_3] \) with \( r_3,t(r_4) \neq r_4,\psi_2(j) \). Without loss of generality we may assume that this integer \( j \) is in the range of \( \psi_1 \), i.e. there exist \( j \in [m_1] \) satisfying
ψ₁(j) = i. Now suppose that neither \( p_i(p_j) \) nor \( p_j(p_i) \) is in \( P \). Then \( r_1(t_1) = r_1(t_2) = r_1(t_3) \) (= t). But this is impossible because \( t, j ≠ t_j \).

Note that Lemma 5 remains valid even if \( A_2 ≤ A_3 \) and \( B_2 ≤ B_3 \) are replaced by \( A'_2 ≤ A'_3 \) and \( B'_2 ≤ B'_3 \).

The proof of the next lemma is similar to the previous one.

**Lemma 6.** Let \( p_1, p_2, p_3 ∈ T_{F,1}, p_4 ∈ T_F, n_1, n'_1, m_1 \equiv 0 \) \((i = 1, 2, 3)\), \( q_1 ∈ T_{G,n_1+1}, q'_1 ∈ T_{G,n_1',+1}, r_1 ∈ T_{G,m_1}, q_2 ∈ T_{G,n_2}, q'_2 ∈ T_{G,n_2'}, r_2 ∈ T_{G,m_2}, q_3 ∈ T_{G,n_3}, q'_3 ∈ T_{G,n_3'}, r_3 ∈ T_{G,m_3}, q_4 ∈ T_{G,n_4}, q'_4 ∈ T_{G,n_4'}, r_4 ∈ T_{G,m_4}, a_0, a_0' ∈ A_0, a_i ∈ A^n, a_i' ∈ A'^n, b_i, b_i' ∈ A^m \((i = 1, 2, 3)\). Furthermore, let \( v' ∈ T_G \) and \( v = r_1(r_2(r_3(r_4))) \). Denote by \( A_i, A_i' \) and \( B_i \((i = 1, 2, 3)\) the sets of components of \( a_i, a_i' \) and \( b_i \), respectively. Assume that

(i) \( p_1(p_2(p_3(p_4))) ∈ T \),

(ii) \( a_0 p_1 ⇒ q_1(r_1(b_1 x_1^n)), a_1 x_1^n, a'_0 p_1 ⇒ q'_1(v', a'_1 x_1'^n) \),

(iii) \( a_1 p_2^{n'} ⇒ q_2(a_2 x_2^n); a'_1 p_2^{n'} ⇒ q'_2(a_2' x_2'^n), b_1 p_2^{n'} ⇒ r_2(b_2 x_2^n) \),

(iv) \( a_2 p_3^{n''} ⇒ q_3(a_3 x_3^n); a'_2 p_3^{n''} ⇒ q'_3(a_3' x_3'^n), b_2 p_3^{n''} ⇒ r_3(b_3 x_3^n) \),

(v) \( a_3 p_4^{n'''} ⇒ q_4; a'_3 p_4^{n'''} ⇒ q'_4, b_3 p_4^{n'''} ⇒ r_4 \),

(vi) \( (p_0 b_0 = (p_3(p_4))_B = (p_2(p_3(p_4)))_B \)

\( A_1 ⊆ A_2 ⊆ A_3, A'_1 ⊆ A'_2 ⊆ A'_3, B_1 = B_2 ⊆ B_3 \),

(vii) \( v ≠ v', \) path \( q_i = \text{path}_i(q_i) \).

Then at least one of the trees \( p_1(p_2(p_3)), p_1(p_3(p_4)), p_1(p_4) \) is in \( P \).

Our last lemma is stated as follows:

**Lemma 7.** Let \( p_1, p_2 ∈ T_{F,1}, p_3 ∈ T_F, k, l, m, m' \equiv 0, q_1 ∈ T_{G,k+1}, q'_1 ∈ T_{G,k'+1}, q_2 ∈ T_{G,l+1}, q'_2 ∈ T_{G,l'+1}, r_1 ∈ T_{G,m}, r'_1 ∈ T_{G,m'}, r_2 ∈ T_{G,l}, q_3 ∈ T_{G,m}, v_1 ∈ T_G, v_2 ∈ T_G \)

\( t ∈ T_G, t' ∈ T_G, t_0 ∈ A_0, a_0' ∈ A_0, a_0' ∈ A'^0, b ∈ A^l, b' ∈ A'^l, c ∈ A^m, c' ∈ A'^m \). Let \( A_1, B_1 \) and \( C_1 \) denote the sets of all components of \( a, b \) and \( c \), respectively. Similarly, denote by \( A'_1, B'_1 \) and \( C'_1 \) the sets of components of \( a', b' \) and \( c' \). Suppose that the following conditions are satisfied:

(i) \( p_1(p_2(p_3)) ∈ T \),

(ii) \( a_0 p_1 ⇒ q_1(ax_1, ax_1^k), a'_0 p_1 ⇒ q'_1(a'x_1, a'x_1^k) \),

(iii) \( a_0 p_2 ⇒ q_2(ax_1, bx_1^k), a'_0 p_2 ⇒ q'_2(a'x_1, b'x_1^k), a_0 b_2^k ⇒ r(cx_1^m), a'_0 b_2^k ⇒ r'(c'x_1'^m) \),

(iv) \( a_0 p_3 ⇒ q_3(v), a'_0 p_3 ⇒ q'_3, b_0 p_3^k ⇒ s, b'_0 p_3^k ⇒ s', c_p^m ⇒ t, c'_p^m ⇒ t' \),

(v) \( A_1 ⊆ B_1 ∪ C_1, A'_1 ⊆ B'_1 ∪ C'_1, (p_3b) = (p_2(p_3))_B \).
(vi) \( \text{path}_1(q'_0) = \text{path}_1(q_0) \text{ path}(q_0), \) \( \text{path}_1(q_2) \text{ path}(q_3) = \) \( = \text{ path}(q_3) \text{ path}_1(q'_2), \) \( v \not= q'_3. \)

Then \( p_1(p_3) \in P. \)

Proof. Let us introduce the following notations: \( d=(b, c), d'=(b', c'), u=(s, t), u'=(s', t'). \) Choose the mappings \( \varphi : [k] \rightarrow [l+m], \varphi' : [k] \rightarrow [l'+m'] \) in such a way that we have \( a_i=d_{\varphi(i)} \) and \( a'_j=d_{\varphi'(j)} \) for every \( i \in [k] \) and \( j \in [k']. \) Obviously, \( a_0 p_1(p_3) \Rightarrow q_0(q_0(v), u_\varphi(1), \ldots, u_\varphi(k)), \) and \( a'_0 p_1(p_3) \Rightarrow q'_0(q'_0, u'_\varphi(1), \ldots, u'_\varphi(k')), \) furthermore, \( p_1(p_3) \in T. \) On the other hand \( \text{path}_1(q'_1(x_1, u'_\varphi(1), \ldots, u'_\varphi(k')), q'_3) \) \( \not= q'_3 \). Therefore, \( q_1(x_1, u_\varphi(1), \ldots, u_\varphi(k)) \) \( \not= q'_1(x_1, u'_\varphi(1), \ldots, u'_\varphi(k')), \) showing that \( p_1(p_3) \in P. \)

We are now able to prove our main result:

Theorem 8. The functionality of top-down as well as bottom-up tree transducers is decidable.

Proof. By Lemma 2 it suffices to prove our statement for regularly restricted top-down transducers. Hence take an arbitrary regularly restricted top-down transducer \( A=(F, T, A, G, A_0, \Sigma) \) with \( T=T(B) \), where \( B \) is the tree automaton \( B=(F, B, B_0). \) Define the set \( P \) and integers \( |A|, |A|, |B| \) and \( K \) as previously (cf. Lemma 4) and let \( L \) denote the number of nonempty strings over \( \{v(G)\} \) with length not exceeding \( ||A||^2|B|K. \) Furthermore, let \( k=||A||^2|A||B|(|A| ||B|K+1) \) and finally, \( m=l+2||A||^2|B|. \)

We shall show that \( P \) is nonvoid if and only if it contains a tree of depth less than \( m. \) It is obvious if \( K=0. \) Therefore let \( K\not=0 \) and assume that \( p \) is an element of \( P \) with minimal rank. Let \( q \) and \( q' \) be different images of \( p \) under \( \tau_A. \)

Assume to the contrary \( dp(p) \equiv m. \) Then there exist \( a_0, a'_0 \in A_0, \) \( p_0, \ldots, p_m \in \hat{T}_{F,1}, \) \( p_{m+1} \in \hat{T}_F, \) \( n_0, n'_0 \equiv 0 \) \( (i=0, \ldots, m), \) \( q_0 \in \hat{T}_{G, n_0}, \) \( q'_0 \in \hat{T}_{G, n'_0}, \) \( q_i \in \hat{T}_{G, n_i} (i=1, \ldots, m), \) \( q_{m+1} \in T_{G, m}, \) \( q'_{m+1} \in T_{G, m}, \) \( a_i \in A^i, \) \( a'_i \in A'^i \) \( (i=0, \ldots, m) \) such that the following three conditions are satisfied:

1. \( p = p_0 p_1(\ldots(p_{m+1})\ldots), \) \( p_i \not= x_i \) \( (i=1, \ldots, m), \)
2. \( q = q_0(q_1(\ldots(q_{m+1})\ldots)), \) \( q' = q'_0(q'_1(\ldots(q'_{m+1})\ldots)), \)
3. \( a_0 p_0 \Rightarrow q_0(a_0 x'_0), \) \( a'_0 p_0 \Rightarrow q'_0(a'_0 x'_0), \)
   \( a_i p'_i \Rightarrow q_i(a_{i+1} x'_{i+1}), \) \( a'_i p'_i \Rightarrow q'_i(a'_{i+1} x'_{i+1}) \) \( (i=0, \ldots, m-1), \)
   \( a_m p'_{m+1} \Rightarrow q_m, \) \( a'_m p'_{m+1} \Rightarrow q'_m. \)

Further on we shall often use the following notations. Let \( i \in \{0, \ldots, m+1\}, \) \( j \in \{0, \ldots, m\}. \) Then \( \tilde{p}_i=p_0(p_1(\ldots(p_i)\ldots)), \) \( \tilde{q}_i=q_0(q_1(\ldots(q_i)\ldots)), \) \( \tilde{q}'_i=q'_0(q'_1(\ldots(q'_i)\ldots)). \) Similarly, \( \tilde{p}_j=p_{j+1}(\ldots(p_{m+1})\ldots), \) \( \tilde{q}_j=q_{j+1}(\ldots(q_{m+1})\ldots), \) \( \tilde{q}'_j=q'_{j+1}(\ldots(q'_{m+1})\ldots). \)

Furthermore, for each \( i=0, \ldots, m, \) \( A_i \) and \( A'_i \) denotes the set of all components of \( a_i \) and \( a'_i, \) respectively.

If for any \( v' \in T_{G, m} \) and \( v' \in T'_{G, m} \) we have \( \tilde{q}_i(v) \not= \tilde{q}'_i(v) \) then, by Lemma 3 and the fact that the cardinality of the set \( \{i, \ldots, m\} \) is at least \( ||A||^2|B|+1, \) we get that for some \( i, j \) \( (i \leq i < j \leq m) \) \( \tilde{p}_i(\tilde{p}_j) \in P. \) It is a contradiction.
Therefore we may assume that \( n_i > 0 \) and the existence of an index \( i \in [n_i] \) such that there are trees \( u' \in \mathcal{F}_{G,1}, v' \in \mathcal{T}_G \) with \( q' = u'(v') \). path \((u') = \text{path}_{i_1}(\tilde{q}_i) \) and \( v' \neq \tilde{q}_i, i_1 \). Obviously, \( n_i > 0 \) holds for each \( i \perp l \). Now let \( i \perp l_{i} \perp i \perp \), \( j \in [n_i] \) be those uniquely determined indices for which \( \text{path}_{i_j}(\tilde{q}_i) \) is a prefix of \( \text{path}_{i_j}(\tilde{q}_i) \). Of course we may assume that \( i_0 = \ldots = i_l = 1 \).

Suppose now that there is no \( x \in \text{path}(\tilde{q}_i) \) such that \( \text{path}_{i_j}(\tilde{q}_i) \) is a prefix of \( \alpha' \) or conversely. In this case let
\[
B_i = \{a_{i,j} \mid \text{path}_{i_j}(\tilde{q}_i) \text{ is a prefix of } \text{path}_{i_j}(\tilde{q}_i)\},
\]
\[
C_i = \{a_{i,j} \mid \text{path}_{i_j}(\tilde{q}_i) \text{ is not a prefix of } \text{path}_{i_j}(\tilde{q}_i)\}
\]
for each \( i \perp i \perp i \). Since the cardinality of the set \( \{l, \ldots, m\} \) is exactly \( 2||A||^2|B| + 1 \) there exist indices \( i_1, i_2, i_3 (l \perp i_1 < i_2 < i_3 \perp m) \) satisfying the following conditions:
\[
(\tilde{p}_i)_B = (\tilde{p}_i)_B = (\tilde{p}_i)_B, \quad B_i = B_i \subseteq B_i, \quad C_i \subseteq C_i \subseteq C_i, \quad A_i \subseteq A_i \subseteq A_i.
\]
By Lemma 6 this yields that at least one of the trees \( \tilde{p}_i(\tilde{p}_i), \tilde{p}_i(\tilde{p}_i), \tilde{p}_i(\tilde{p}_i) \) is in \( P \), which is a contradiction.

We have shown that there exists an \( \alpha' \in \text{path}(\tilde{q}_i) \) such that \( \text{path}_{i_j}(\tilde{q}_i) \) is a prefix of \( \alpha' \) or conversely. Consequently \( n_i > 0 \) for each \( i \perp i \perp i \) and there exist integers \( i_0, \ldots, i_l \) with the property that \( \text{path}_{i_j}(\tilde{q}_i) \) is a prefix of \( \text{path}_{i_j}(\tilde{q}_i) \) or conversely \( (j = 0, \ldots, l) \). We may also assume that if \( j < l_2 \) then \( \text{path}_{i_j}(\tilde{q}_j) \) is a prefix of \( \text{path}_{i_{j+1}}(\tilde{q}_{j+1}) \), moreover, we may assume that \( i_0 = \ldots = i_l = 1 \). In this way either \( \text{path}_{i_j}(\tilde{q}_j) \) is a prefix of \( \text{path}_{i_j}(\tilde{q}_j) \) \( (j = 0, \ldots, l) \) or conversely.

Now there are two cases. First suppose that \( \text{path}_{i_j}(\tilde{q}_j) \) is a prefix of \( \text{path}_{i_j}(\tilde{q}_j) \). If, within this case, there exists an integer \( i \perp i \perp i \perp i \) such that \( ||\text{path}_{i_j}(\tilde{q}_i)|| - ||\text{path}_{i_j}(\tilde{q}_j)|| \geq ||A||^2|B|/K \) then, by Lemma 4, there is a tree \( r \in \mathcal{T}_F \) satisfying both \( \tilde{p}_i(r) \in P \) and \( \text{rn}(r) < \text{rn}(\tilde{p}_i) \). This is a contradiction because \( \text{rn}(r) < \text{rn}(\tilde{p}_i) \) implies \( \text{rn}(\tilde{p}_i) < \text{rn}(p) \). Thus we have \( ||\text{path}_{i_j}(\tilde{q}_i)|| - ||\text{path}_{i_j}(\tilde{q}_j)|| \leq ||A||^2|B|/K \) for every \( i \perp i \perp i \). But this yields another contradiction. Indeed, the cardinality of the set \( \{0, \ldots, k\} \) is equal to \( ||A||^2|A|/|B|(2L+1)+1 \), thus, there are at least two indices \( i, j \perp i \perp i \perp i \) such that \( \text{path}_{i_j}(\tilde{q}_i) \) is a prefix of \( \text{path}_{i_j}(\tilde{q}_j) \), \( \text{path}_{i_j}(\tilde{q}_j) \) is a prefix of \( \text{path}_{i_j}(\tilde{q}_i) \), \( \text{path}_{i_{j+1}}(\tilde{q}_j) \) is a prefix of \( \text{path}_{i_{j+1}}(\tilde{q}_i) \), moreover, \( (\tilde{p}_i)_B = (\tilde{p}_j)_B, \quad a_{i_1} = a_{i_1}, \quad a_{i_1} = a_{i_1}, \quad B_i \subseteq B_j, \quad B_j \subseteq B_j \) such that \( B_{i_j} = \{a_{i_1}, 2 \leq i_1 < n_{i_1}\}, \quad B_{i_j} = \{a_{i_1}, 2 \leq i_1 < n_{i_1}\}, \quad s = i_1, j \). By an application of Lemma 7 this yields that \( \tilde{p}_i(\tilde{p}_j) \in P \) — contrary to the minimality of \( p \).

We have shown that \( \text{path}_{i_j}(\tilde{q}_j) \) can not be a prefix of \( \text{path}_{i_j}(\tilde{q}_j) \). Therefore \( \text{path}_{i_j}(\tilde{q}_j) \) is a prefix of \( \text{path}_{i_j}(\tilde{q}_j) \). If we prove that \( ||\text{path}_{i_j}(\tilde{q}_j)|| - ||\text{path}_{i_j}(\tilde{q}_j)|| \geq ||A||^2|B|/K \) then also \( ||\text{path}_{i_j}(\tilde{q}_j)|| - ||\text{path}_{i_j}(\tilde{q}_j)|| \geq ||A||^2|B|/K \). Again by Lemma 4, this yields a contradiction. Therefore it is enough to show that \( ||\text{path}_{i_j}(\tilde{q}_j)|| \geq ||A||^2|B|/K \).

Assume that this condition does not hold. The cardinality of the set \( \{k+1, \ldots, l\} \) is exactly \( 2||A||^2|A|/|B| ||A||^2|B|/K + 1 \), therefore, there exist indices \( i_1, i_2, i_3 (k \leq i_1 < i_2 \leq i_3) \) such that \( i_1 - i_2 = 2 ||A||^2|A|/|B| \) and \( \text{path}_{i_1}(\tilde{q}_i) = \ldots = \text{path}_{i_1}(\tilde{q}_i) \), i.e. \( q_{i_1+1,1} = \ldots = q_{i_1+1,1} \). Now let
\[
B_i = \{a_{i,j} \mid 1 \leq i_1 < n_{i_1}, \quad \text{path}_{i_1}(\tilde{q}_i) \text{ is a prefix of } \text{path}_{i_1}(\tilde{q}_i)\},
\]
\[
C_i = \{a_{i,j} \mid 1 \leq i_1 < n_{i_1}, \quad \text{path}_{i_1}(\tilde{q}_i) \text{ is not a prefix of } \text{path}_{i_1}(\tilde{q}_i)\}
\]
for each \( j_1 \perp j_2 \perp j_3 \). Since the cardinality of \( \{i_1, \ldots, i_3\} \) is equal to \( 2||A||^2|A|/|B| + 1 \) there exist indices \( j_1, j_2, j_3 (i_1 \perp j_1 < j_2 < j_3 \perp i_3) \) such that each of the following
conditions is satisfied: \((\bar{p}_1)_B = (\bar{p}_1)_B = (\bar{p}_1)_B\), \(A_{j_1} \subseteq A_{j_2} \subseteq A_{j_3}\), \(B_{j_1} = B_{j_2} \subseteq B_{j_3}\), \(C_{j_1} \subseteq C_{j_2} \subseteq C_{j_3}\), \(a_{j_1,1} = a_{j_2,1} = a_{j_3,1}\), where \(A_{j_1} = \{a_{j_1,s} \mid 2 \leq s \leq n\}\). Thus, applying Lemma 5, we get that one of the trees \(\bar{p}_1, (\bar{p}_1)_B, \bar{p}_1(B_{j_3}), \bar{p}_1(p_{j_3})\) is in \(P\), contradicting to the minimality of \(p\). This ends the proof of Theorem 8.

Observe that, by the decomposition result for top-down tree transducers with regular look-ahead in [6], the above theorem holds for this type of transducers as well. But Theorem 8 has some other important consequences, too.

Take two arbitrary top-down or bottom-up tree transducers \(A = (F, A, G, A_0, \Sigma)\) and \(B = (F, B, G, B_0, \Sigma')\). Assume that \(A\) is functional and \(A\) and \(B\) are disjoint. Then construct the sum of \(A\) and \(B\), i.e. take \(C = (F, A \cup B, G, A_0 \cup B_0, \Sigma \cup \Sigma')\). For \(C\) we have the following equivalence: \(\tau_A = \tau_B\) if and only if \(\text{dom } \tau_A = \text{dom } \tau_B\) and \(C\) is functional. From this and by the fact that the equality of regular forests is decidable we get:

**Theorem 9.** There exists an algorithm to decide for an arbitrary tree transducer \(A\) and a functional transducer \(B\) whether they are equivalent, i.e. such that \(\tau_A = \tau_B\).

**Corollary.** A similar argument shows that Theorem 9 holds even if \(\tau_A = \tau_B\) is replaced by \(\tau_A \subseteq \tau_B\). On the other hand every deterministic transducer is functional. Thus, the equivalence problem for deterministic transducers is decidable.

Another consequence of Theorem 8 concerns with minimization of transducers. For any given tree transducer \(A\) one can compute a bound \(k\) with the following property: \(A\) has a corresponding tree transducer \(B\) which is minimal and satisfies that each tree in the right hand side of a rule of \(B\) has depth not exceeding \(k\). This \(k\) can be obtained as \(2K|A|\) in the top-down case and as \(2K|A|\) in the bottom-up case. (Here \(|A|, |B|\) and \(K\) are determined as in the proof of Theorem 8.) Therefore, if we assume that \(A\) is functional and we want to minimize \(A\), it is enough to check only for a finite number of transducers whether they are equivalent to \(A\) or not. This proves

**Theorem 10.** The minimization of functional tree transducers is effectively solvable.

**Corollary.** As every deterministic tree transducer is functional the same statement holds for deterministic transducers.

This corollary as well as the positive decidability result concerning the equivalence problem for deterministic bottom-up transducers and a restricted class of deterministic top-down transducers was independently achieved by Z. Zachar in [12] too.

### 3. Minimization of deterministic transducers

Let \(\mathcal{H}\) be a class of tree transducers. A transducer \(A \in \mathcal{H}\) is said to be **minimal** in \(\mathcal{H}\) if there is no transducer \(B \in \mathcal{H}\) which is equivalent to \(A\) and has fewer states than \(A\). In the preceding section we have shown that if \(\mathcal{H}\) is the class of all functional top-down or all bottom-up transducers, or if \(\mathcal{H}\) is the class of all deterministic top-down or all bottom-up transducers, then, for every given \(A \in \mathcal{H}\), one can effectively find a minimal equivalent transducer \(B \in \mathcal{H}\). However, these minimal realiza-
tions are not uniquely determined. In this section we investigate conditions assuring the uniqueness (up to isomorphism) of minimal realizations. Similar results are already known for Mealy-type automata (cf. [9]) and tree automata [1, 3, 10]. We point out that the minimizing process of Mealy-type automata can be generalized in a natural way for certain classes of deterministic tree transducers. For the sake of simplicity we shall consider completely defined deterministic tree transducers only. Therefore, from now on, by a tree transducer we shall always mean a completely defined deterministic transducer. For the sake of simplicity we shall consider completely defined deterministic tree transducers only. Therefore, from now on, by a tree transducer we shall always mean a completely defined deterministic transducer. Furthermore, all transducers will be taken with a fixed input type $F$ and output type $G$. Since the case $F=F_0$ is trivial we assume that $F^F\neq F_0$.

First we treat top-down transducers. Let $A=(F, A, G, \{a_0\}, \Sigma)$ be a top-down transducer. It is completely defined, i.e. for any $a\in A$ and $f\in F$ there is a rule in $\Sigma$ with left side $af$. Let $B=(F, B, G, \{b_0\}, \Sigma')$ be another top-down transducer and take a mapping $\varphi: A \rightarrow B$. If the following two conditions are satisfied for arbitrary $n, m \geq 0, f \in F_n, p \in T_{G,m}, a, a_1, \ldots, a_m \in A$ and $i_1, \ldots, i_m \in [n]$ then $\varphi$ is called a homomorphism of $A$ into $B$:  

(i) if $af \rightarrow p(a_1 x_{i_1}, \ldots, a_m x_{i_m}) \in \Sigma$ then $bf \rightarrow p(b_1 x_{i_1}, \ldots, b_m x_{i_m}) \in \Sigma'$ where $b=\varphi(a), b_1 = \varphi(a_1) (j \in [m])$,

(ii) $\varphi(a_0) = b_0$.

If, moreover, $\varphi$ is surjective then $B$ is a homomorphic image of $A$. If $\varphi$ is bijective then we speak about isomorphism, written $A \cong B$. If $B \subseteq A$ and $\varphi$ is the natural embedding of $B$ into $A$ then $B$ is a subtransducer of $A$. If $A$ has not proper subtransducers then it is called connected.

The next statement is obvious:

Statement 11. If there is a homomorphism from $A$ into $B$ then $\tau_A = \tau_B$.

As in case of universal algebras there is a bijective correspondence between homomorphic images and congruence relations. Let $A=(F, A, G, \{a_0\}, \Sigma)$ be an arbitrary top-down transducer and take an equivalence relation $\theta$ on $A$. It is called a congruence relation if for any two rules $af \rightarrow p(a_1 x_{i_1}, \ldots, a_m x_{i_m}), bf \rightarrow q(b_1 x_{j_1}, \ldots, b_l x_{j_l}) \in \Sigma (n, m, l \geq 0, f \in F_n, p \in T_{G,m}, q \in T_{G,l}, i_1, \ldots, i_m, j_1, \ldots, j_l \in [n], a_1, \ldots, a_m, b_1, \ldots, b_l, a, b \in A)$ $\theta$ implies $m=l, p=q, i_j=j_l$ and $a\theta b, (t=1, \ldots, m)$. Here for any nonnegative integer $n$ the notation $T_{G,n}$ is used to denote the set $T_{G,n} = \{p \in T_{G,n} | \text{fr}(p) = x_{i_1} \ldots x_{i_l}\}$.

Assume that $\theta$ is a congruence relation of $A$. Then we can define the quotient of $A$ induced by $\theta$. This is the top-down transducer $A/\theta = (F, A, G, \{\theta(a_0)\}, \Sigma')$ where for every $n, m \geq 0, f \in F_n, p \in T_{G,m}, a, a_1, \ldots, a_m \in A$

$\theta(a) f \rightarrow p(\theta(a_1) x_{i_1}, \ldots, \theta(a_m) x_{i_m}) \in \Sigma'$

if and only if

$af \rightarrow p(a_1 x_{i_1}, \ldots, a_m x_{i_m}) \in \Sigma$.

Statement 12. $A/\theta$ is a homomorphic image of $A$. If $B$ is a homomorphic image of $A$ then there is a congruence relation $\theta$ of $A$ such that $A/\theta \cong B$.

Take again the top-down tree transducer $A=(F, A, G, \{a_0\}, \Sigma)$. Let us define an equivalence relation $\theta_A$ on $A$: $a \theta_A b$ if and only if $\tau_{A(a)} = \tau_{A(b)}$. Unfortunately, this will not always be a congruence relation. We need certain additional requirements on $A$.  

Let $\varrho$ be any mapping of the set of nonnegative integers into itself, i.e. \( \varrho: \omega \rightarrow \omega \).
Then let \( \mathcal{K}(\varrho) \) denote the class of all top-down tree transducers \( A=(F, A, G, \{a_0\}, \Sigma) \) which satisfy the condition \( |\text{path}_t(p)| = \varrho(i) \) for every \( n, m \geq 0 \), \( f \in F_n, p \in \mathcal{T}_{G,m}, a, a_1, \ldots, a_m \in A, x_1, \ldots, x_m \in \Sigma, j \in [m] \) and \( af \rightarrow p(a_1x_1, \ldots, a_mx_m) \in \Sigma \), as well as the condition \( |\tau_{A(o)}(T_F)| \geq 1 \) for arbitrary state \( a \) appearing in the right side of a rule in \( \Sigma \).

**Statement 13.** If \( A \in \mathcal{K}(\varrho) \) then \( \theta_A \) is a congruence relation.

**Proof.** Let \( A=(F, A, G, \{a_0\}, \Sigma) \) and assume that \( af \rightarrow p(a_1x_1, \ldots, a_mx_m) \) and \( bf \rightarrow q(b_1x_j, \ldots, b_jx_j) \) are rules in \( \Sigma \) where \( a, b \in A, a \theta_A b, n, m, l \geq 0, f \in F_n, p \in \mathcal{T}_{G,m}, q \in \mathcal{T}_{G,j}, a_1, \ldots, a_m, b_1, \ldots, b_j \in A, i_1, \ldots, i_m, j_1, \ldots, j_l \in [n] \). Assume that there is an integer \( i \in [m] \) such that none of the strings in \( \bigcup \{ \text{path}_t(p) | i_t = j, s \in [l] \} \) is a prefix of \( \text{path}_t(p) \) or conversely. Then, by \( |\tau_{A(o)}(T_F)| = 1 \), it is easy to show the existence of a tree \( r \in T_F \) with \( \tau_{A(o)}(r) = \tau_{B(o)}(r) \). On the other hand if \( i_t = j_s \) holds for some \( i \in [m] \) and \( s \in [l] \) then the equality \( |\text{path}_t(p)| = |\text{path}_t(q)| \) is also valid. This proves that \( m = l, i_t = j_s, \text{path}_t(p) = \text{path}_t(q) \) for \( t = 1, \ldots, m \). But \( \tau_{A(o)} = \tau_{B(o)} \), hence from this we get \( p = q, a \theta_A b \). Consequently, \( A \in \mathcal{K}(\varrho) \) if and only if \( a \theta_A b \).

Another class of top-down transducers in which \( \theta_A \) is always a congruence relation is the class \( \mathcal{K}_d \), where \( d \) denotes an arbitrary nonnegative integer. A top-down transducer \( A=(F, A, G, \{a_0\}, \Sigma) \) is in \( \mathcal{K}_d \) if and only if for every \( a \in A, f \in F_0 \) and \( p \in \mathcal{T}_G \), \( af \rightarrow p \Sigma \) holds for every \( n, m \geq 0, f \in F_n, p \in \mathcal{T}_{G,m}, q \in \mathcal{T}_{G,j}, a_1, \ldots, a_m, b_1, \ldots, b_j \in A, i_1, \ldots, i_m, j_1, \ldots, j_l \in [n] \). Assume that \( a \in A, f \in F_0 \) and \( q \in \mathcal{T}_{G,j} \) for the rules \( af \rightarrow p(a_1x_1, \ldots, a_mx_m) \) and \( bf \rightarrow q(b_1x_j, \ldots, b_jx_j) \).

**Statement 14.** If \( A \in \mathcal{K}_d \) then \( \theta_A \) is a congruence relation.

**Proof.** The proof of this statement is similar to that of Statement 13. Only use the conditions defining \( \mathcal{K}_d \) to establish the bijective correspondence between the sets \( \bigcup \{ \text{path}_t(p) | i_t = j, s \in [l] \} \) and \( \bigcup \{ \text{path}_t(q) | s \in [l] \} \) for the rules \( af \rightarrow p(a_1x_1, \ldots, a_mx_m) \) and \( bf \rightarrow q(b_1x_j, \ldots, b_jx_j) \).

Note that for \( A \in \mathcal{K}(\varrho) \) or \( A \in \mathcal{K}_d \) the definition of \( \theta_A \) can be reformulated as follows. Let \( a, b \in A \). Then \( a \theta_A b \) if and only if for every \( n, m \geq 0, f \in F_n, p \in \mathcal{T}_{G,m} \) and \( i_1, \ldots, i_m \in [n] \) the following equivalence holds:

\[ \exists a_1, \ldots, a_m \in A \quad \forall p \quad q(a_1x_1, \ldots, a_mx_m) \]

if and only if

\[ \exists b_1, \ldots, b_m \in A \quad \forall p \quad q(b_1x_1, \ldots, b_mx_m) \]

This is an easy consequence of statements 13, 14. Observe that this new definition of \( \theta_A \) makes \( \theta_A \) a congruence relation without requiring \( A \in \mathcal{K}(\varrho) \) or \( A \in \mathcal{K}_d \).

A transducer \( A \in \mathcal{K}(\varrho) \) or \( A \in \mathcal{K}_d \) is called reduced if \( \theta_A \) is the equality relation. As both \( \mathcal{K}(\varrho) \) and \( \mathcal{K}_d \) are closed under homomorphic images the transducer \( A/\theta_A \) is reduced for any \( A \in \mathcal{K}(\varrho) \) or \( A \in \mathcal{K}_d \). The following statement is the basic step to show that minimal transducers in \( \mathcal{K}(\varrho) \) and \( \mathcal{K}_d \) are exactly the connected and reduced transducers.

**Theorem 15.** Let \( A, B \in \mathcal{K}(\varrho) \) be connected top-down transducers. Then \( A \) and \( B \) are equivalent if and only if \( A/\theta_A \cong B/\theta_B \). The same holds for \( \mathcal{K}_d \).

**Proof.** Sufficiency follows by statements 11—14. In order to prove necessity first observe that if \( A=(F, A, G, \{a_0\}, \Sigma) \) and \( B=(F, B, G, \{b_0\}, \Sigma') \), moreover,
$a_0 p \Rightarrow A q(a_1 x_{i_1}, \ldots, a_m x_{i_m})$ — where $p \in T_{F,n}$, $n \geq 0$, $q \in T_{G,m}$, $m \geq 0$, $a_1, \ldots, a_m \in A$, $i_1, \ldots, i_m \in [n]$ — then there exist states $b_1, \ldots, b_m \in B$ with $b_0 p \Rightarrow_B q(b_1 x_{i_1}, \ldots, b_m x_{i_m})$. Furthermore, for these states $b_i (i = 1, \ldots, m)$ we have $\tau_A(b_0) = \tau_B(b_0)$. This is a consequence of the assumption $\tau_A = \tau_B$ and the definitions of $\mathcal{X}(q)$ and $\mathcal{X}_d$. Using the above mentioned facts it is easy to prove that the correspondence $\varphi: A/\theta_A \rightarrow B/\theta_B$ defined by $\varphi(\theta_A(a)) = \theta_B(b)$ if and only if there exist $p \in T_{F,1}$, $q \in T_{G,m+1}$ ($m \geq 0$), $a_1, \ldots, a_m \in A$, $b_1, \ldots, b_m \in B$ such that $a_0 p \Rightarrow_A q(a_1 x_{1}, a_1 x_{1}, \ldots, a_m x_{1})$ and $b_0 p \Rightarrow_B q(b_1 x_{1}, b_1 x_{1}, \ldots, b_m x_{1})$ forms an isomorphism of $A/\theta_A$ into $B/\theta_B$.

The next theorem is an immediate consequence of Theorem 15 and the fact that $\mathcal{X}(q)$ and $\mathcal{X}_d$ are closed under the formation of subtransducers and homomorphic images.

**Theorem 16.** A transducer is minimal in $\mathcal{X}(q)$ if and only if it is connected and reduced. If both $A$ and $B$ are minimal in $\mathcal{X}(q)$ and they are equivalent then $A \cong B$, i.e. the minimal realization of a transducer in $\mathcal{X}(q)$ is unique up to isomorphism. The same holds for the class $\mathcal{X}_d$.

Of course Theorem 16 holds for every class $\mathcal{X} \subseteq \mathcal{X}(q)$ or $\mathcal{X} \subseteq \mathcal{X}_d$ provided $\mathcal{X}$ is closed under the formation of subtransducers and homomorphic images. The most important example for a class of this type is the class of all top-down relabelings (cf. [5]).

It is natural to raise the question whether the minimal transducers in $\mathcal{X}(q)$ or $\mathcal{X}_d$ are minimal in the class of all top-down transducers. The following examples prove that the answer is negative in general. In these examples the adjectives “linear”, “nondeleting” are used in the sense of [5]. Furthermore, a top-down tree transducer $A = (F, \Gamma, G, \{a_0\}, \Sigma)$ will be called uniform if each rule $af \rightarrow p$ ($a \in A$, $f \in F_n$ ($n \geq 0$), $p \in T_{G,A \times X_n}$) can be written as $af \rightarrow q(a_1 x, \ldots, a_n x)$ for a tree $q \in T_{G,n}$ and states $a_1, \ldots, a_n \in A$.

**Example 17.** This example shows that there is a linear nondeleting top-down tree transducer $A \in \mathcal{X}_1 \cap \mathcal{X}(q)$ which is connected and reduced — i.e. minimal in both $\mathcal{X}$ and $\mathcal{X}(q)$ — but which is not minimal in the class of all linear nondeleting top-down tree transducers. Here $q: \omega \rightarrow \omega$ is the mapping defined by $q(n) = 1$ ($n \geq 0$). Indeed, let $A = (F, [5], F, \{1\}, \Sigma)$ where $F$ is the type determined by the conditions $F_0 = \{\#\}$, $F_1 = \{f, g\}$, $F_n = \emptyset$ if $n > 1$ and $\Sigma$ consists of the rules (1)—(5) listed below:

1. $1 \rightarrow f(\#)$, $1 f(x_1) \rightarrow f(2x_1)$, $1 g(x_1) \rightarrow g(3x_1)$,
2. $2 \rightarrow f(\#)$, $2 f(x_1) \rightarrow f(4x_1)$, $2 g(x_1) \rightarrow f(4x_1)$,
3. $3 \rightarrow g(\#)$, $3 f(x_1) \rightarrow g(4x_1)$, $3 g(x_1) \rightarrow g(4x_1)$,
4. $4 \rightarrow f(\#)$, $4 f(x_1) \rightarrow f(5x_1)$, $4 g(x_1) \rightarrow g(5x_1)$,
5. $5 \rightarrow f(\#)$, $5 f(x_1) \rightarrow f(1x_1)$, $5 g(x_1) \rightarrow g(1x_1)$.

However, $A$ is equivalent to $A' = (F, [4], F, \{1\}, \Sigma')$ where $\Sigma'$ contains the following rules (1)–(4):

1. $1 \rightarrow f(\#)$, $1 f(x_1) \rightarrow f(2x_1)$, $1 g(x_1) \rightarrow g(2x_1)$,
Example 18. This example proves that there is a top-down tree transducer $A \in \mathcal{K}_0$ which is minimal in $\mathcal{K}_0$ but not minimal in the class of all top-down transducers.

Let us define the types $F$ and $G$ by $F_0 = \{\#\}$, $F_1 = \{f\}$, $G_0 = \{\#\}$, $G_1 = \{f\}$, $G_2 = \{g\}$, $G_3 = \{g\}$, respectively. Then put $A = (F, [4], G, [1], \Sigma)$ where $\Sigma$ consists of the following rules:

1. $1 \# \rightarrow \#$, $1f(x_1) \rightarrow g(2x_1, 3x_1)$,
2. $2 \# \rightarrow \#$, $2f(x_1) \rightarrow f(4x_1)$,
3. $3 \# \rightarrow g(\#, \#)$, $3f(x_1) \rightarrow f(3x_1)$,
4. $4 \# \rightarrow \#$, $4f(x_1) \rightarrow f(1x_1)$.

It is easy to check that $A$ is minimal in $\mathcal{K}_0$. On the other hand $A$ is equivalent to $A' = (F, [3], G, [1], \Sigma')$ with $\Sigma'$ containing the following rules:

1. $1 \# \rightarrow \#$, $1f(x_1) \rightarrow g(2x_1, 3x_1)$,
2. $2 \# \rightarrow g(\#, \#)$, $2f(x_1) \rightarrow g(3x_1)$,
3. $3 \# \rightarrow \#$, $3f(x_1) \rightarrow f(3x_1)$,
4. $4 \# \rightarrow \#$, $4f(x_1) \rightarrow f(1x_1)$.

Observe that $A$ was not uniform.

In spite of Example 18 we have

Theorem 19. If a uniform transducer is minimal in $\mathcal{K}_0$ then it is minimal in the class of all top-down tree transducers.

Proof. Let $A = (F, A, G, \{a_0\}, \Sigma) \in \mathcal{K}_0$ be uniform and minimal in $\mathcal{K}_0$. Assume that the top-down tree transducer $B = (F, B, G, \{b_0\}, \Sigma')$ is equivalent to $A$ and has fewer states than $A$, i.e. $|B| < |A|$. Then take an arbitrary state $a \in A$. We shall correspond to this state a state $\phi(a) \in B$ as follows. Let us choose the trees $p \in T_{F,1}$ and $q \in T_{G,n}$ such that we have $a \phi(a) \rightarrow A \in A$.

Let $r \in T_{G,m}$ be determined by $b \in B$ and $m \in B$. As $|r_{A(c)}(T_F)| > 1$ is satisfied for each $c \in A$ occurring in the right side of a rule in $\Sigma$ we must have $m > 0$. Or even, there must be an index $j \in [m]$ for each $i \in [n]$ with the property that either $\text{path}_{j_i}(r)$ is a prefix of $\text{path}_i(q)$ or conversely. But, by the definition of $\mathcal{K}_0$, it is impossible that $\text{path}_i(q)$ is a proper prefix of $\text{path}_{j_i}(r)$. Therefore $J_i$ is uniquely determined for each $i \in [n]$ and $\text{path}_i(r)$ is a prefix of $\text{path}_i(r)$. As $A$ and $B$ are equivalent this implies that there exist trees $r_{11}, \ldots, r_{1m} \in T_{G,1}$ with $r(r_{11}, \ldots, r_{1m}) = q$. Let $\phi(a) = b_1$ and $r_a = r_{j_1}$. We must have $r_a(\tau_{A(a)}(t)) = \tau_{B(\phi(a)}(t)$ for each $t \in T_F$, i.e. $r_a(\tau_{A(a)}(t)) = \tau_{B(\phi(a)}(t)$.

As $|B| < |A|$ there exist states $a_1 \neq a_2 \in A$ with $\phi(a_1) = \phi(a_2)$. Consequently, $r_a(\tau_{A(a_1)}(t)) = r_a(\tau_{A(a_1)}(t))$. But, again by the definition of $\mathcal{K}_0$, this is possible only if $r_a = r_{a_2}$ and $\tau_{A(a_1)} = \tau_{A(a_2)}$ yielding a contradiction.
We will now turn our attention to the bottom-up case. A deterministic bottom-up tree transducer \( A=(F, A, G, A_0, \Sigma) \) is called completely defined if there is a rule in \( \Sigma \) with left hand side \( f(a_1x_1, \ldots, a_nx_n) \) for every \( n \geq 0 \), \( f \in F_n \) and \( a_1, \ldots, a_n \in A \). First of all we have to define homomorphisms, congruence relations etc.

Let \( A=(F, A, G, A_0, \Sigma) \) and \( B=(F, B, G, B_0, \Sigma') \) be bottom-up transducers. By a homomorphism of \( A \) into \( B \) we mean a mapping \( \varphi: A \rightarrow B \) which satisfies the following two conditions:

(i) \( f(b_1x_1, \ldots, b_nx_n) \rightarrow bp \in \Sigma' \) if \( f(a_1x_1, \ldots, a_nx_n) \rightarrow ap \in \Sigma, \quad b_1 = \varphi(a_1) \)

\((i=1, \ldots, n), \quad b = \varphi(a) \text{ (} n \equiv 0, \quad f \in F_n, \quad a_1, \ldots, a_n, a \in A, \quad p \in T_{G,n},\)

(ii) \( \varphi(A_0) \subseteq B_0, \quad \varphi^{-1}(B_0) \subseteq A_0 \).

Again, if \( \varphi \) is surjective then \( B \) is a homomorphic image of \( A \) and bijective homomorphisms are called isomorphisms. If \( B \subseteq A \) and \( \varphi \) is the natural embedding of \( B \) into \( A \) then \( B \) is a substransducer of \( A \).

We now define congruence relations. A congruence relation of \( A \) is an equivalence relation \( \theta \) on \( A \) with the following property: for any \( n \geq 0 \), \( f \in F_n \), \( a_1, b_1 \in A \) \((i=1, \ldots, n)\), \( a, b \in A \) and \( p, q \in T_{G,n} \) if both \( f(a_1x_1, \ldots, a_nx_n) \rightarrow ap \) and \( f(b_1x_1, \ldots, b_nx_n) \rightarrow bq \) are in \( \Sigma \) and \( a_1\theta b_1 \) \((i=1, \ldots, n)\) are satisfied then \( p=q \) and \( ab\theta b \) hold too. Furthermore, \( A_0 \) is required to be equal to the union of certain blocks of the partition induced by \( \theta \): \( A_0 = \bigcup (\theta(a) | a \in A_0) \). The quotient transducer determined by \( \theta \) is the transducer \( A/\theta=(F, A/\theta, G, A_0/\theta, \Sigma') \) where

\[ \Sigma' = \{f(\theta(a_1)x_1, \ldots, \theta(a_n)x_n) \rightarrow \theta(a)p | f(a_1x_1, \ldots, a_nx_n) \rightarrow ap \in \Sigma\} \]

With the above definitions in mind one can easily prove the analogues of statements 11 and 12.

For a bottom-up transducer \( A=(F, A, G, A_0, \Sigma) \) the relation \( \theta_A \) is defined as follows. Let \( a, b \in A \). Then \( a\theta_A b \) if and only if the equivalence \( \exists a_0 \in A_0 \quad p(a_1x_1, \ldots, a_{i-1}x_{i-1}, ax_i, a_{i+1}x_{i+1}, \ldots, a_nx_n) \rightarrow a_0q \Rightarrow \exists b_0 \in A_0 \quad p(a_1x_1, \ldots, a_{i-1}x_{i-1}, bx_i, a_{i+1}x_{i+1}, \ldots, a_nx_n) \rightarrow b_0q \) holds for all \( n \geq 0, \quad i \in [n], \quad a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \in A, \quad p \in T_{F,n} \) (or equivalently \( p \in T_{F,n} \) or \( p \in T_{G,n} \)) and \( q \in T_{G,n} \).

Likewise in the top-down case, \( \theta_A \) will not always be a congruence relation, but it will be a congruence relation if we require \( A \) to be in \( \mathcal{K}(\varrho) \) for a mapping \( \varrho \) of the set of nonnegative integers into itself. A bottom-up transducer \( A=(F, A, G, A_0, \Sigma) \) belongs to \( \mathcal{K}(\varrho) \) provided it satisfies the following three conditions:

(i) if \( f(a_1x_1, \ldots, a_nx_n) \rightarrow ap \in \Sigma \) \((n \geq 0, \quad f \in F_n, \quad a_1, \ldots, a_n \in A, \quad p \in T_{G,n} \) then \( |w| = \varrho(i) \) holds for each \( i \in [n] \) and \( w \in \text{path}_i(p) \),

(ii) \( A \) is nondeleting, i.e. for all \( n \geq 0, \quad f \in F_n, \quad a, a_1, \ldots, a_n \in A \) and \( p \in T_{G,n} \) if \( f(a_1x_1, \ldots, a_nx_n) \rightarrow ap \in \Sigma \) then each of the variables \( x_1, \ldots, x_n \) occurs in \( \text{fr}(p) \),

(iii) for any \( a \in A \) there exist \( p \in T_{F,n+1}, \quad q \in T_{G,n+1} \) \((n \geq 0)\), \( a_0 \in A_0, \quad a_1, \ldots, a_n \in A \) such that \( p(ax_1, a_1x_2, \ldots, a_nx_{n+1}) \rightarrow a_0q \).

Statement 20. If \( A \in \mathcal{K}(\varrho) \) then \( \theta_A \) is a congruence relation.
Proof. Let \( A = (F, A, G, A_0, \Sigma) \), \( a, b \in A \). Assume that \( a \theta_A b \) and let
\[
\begin{align*}
f(a_1 x_1, \ldots, a_{i-1} x_{i-1}, a x_i, a_{i+1} x_{i+1}, \ldots, a_n x_n) & \rightarrow cp, \\
f(a_1 x_1, \ldots, a_{i-1} x_{i-1}, b x_i, a_{i+1} x_{i+1}, \ldots, a_n x_n) & \rightarrow dq
\end{align*}
\]
be arbitrary rules in \( \Sigma \). Here \( n \geq 0 \), \( i \in [n] \), \( f \in F_n \), \( a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n, c, d \in A \), \( p, q \in T_{G,n} \). We have to show that \( p = q \) and \( c \theta_A d \).

As \( A \in \mathcal{H}(q) \), there exist \( m \geq 0 \), \( c_1, \ldots, c_m \in A \), \( a_0 \in A_0 \), \( r \in T_{F,m+1} \) and \( s \in T_{G,m+1} \) such that
\[
r(c x_1, c_1 x_2, \ldots, c_m x_{m+1}) \Rightarrow a_0 s.
\]

Let \( r_1 = r(f(x_1, \ldots, x_n), x_{n+1}, \ldots, x_{n+m}) \), \( s_1 = s(p, x_{n+1}, \ldots, x_{n+m}) \). Of course we have
\[
r_1(a_1 x_1, \ldots, a_{i-1} x_{i-1}, a x_i, a_{i+1} x_{i+1}, \ldots, a_n x_n, c_1 x_{n+1}, \ldots, c_m x_{n+m}) \Rightarrow a_0 s_1.
\]

Since \( a \theta_A b \), this implies
\[
r_1(a_1 x_1, \ldots, a_{i-1} x_{i-1}, b x_i, a_{i+1} x_{i+1}, \ldots, a_n x_n, c_1 x_{n+1}, \ldots, c_m x_{n+m}) \Rightarrow b_0 s_1
\]
for a state \( b_0 \in A_0 \). But this is possible only if \( s_1 \) is of form \( s_1 = t(q, x_{n+1}, \ldots, x_{n+m}) \) where \( t \in T_{G,m+1} \) and \( r(dx_1, c_1 x_2, \ldots, c_m x_{m+1}) \Rightarrow b_0 t \).

We know that \( s(p, x_{n+1}, \ldots, x_{n+m}) = t(q, x_{n+1}, \ldots, x_{n+m}) \). By (i) and (ii) in the definition of \( \mathcal{H}(q) \) this results that \( s = t \) and \( p = q \). Essentially the same argument shows that \( c \theta_A d \).

Observe that for a bottom-up transducer \( A = (F, A, G, A_0, \Sigma) \in \mathcal{H}(q) \) the relation \( \theta_A \) can be redefined as follows. Let \( a, b \in A \). Then \( a \theta_A b \) if and only if the following two equivalences are satisfied for arbitrary \( p \in T_{F,n}, q \in T_{G,n} \), \( n \geq 0 \), \( a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \in A \) and \( i \in [n] \):
\[
\begin{align*}
(i) & \quad \exists a_0 \in A. p(a_1 x_1, \ldots, a_{i-1} x_{i-1}, a x_i, a_{i+1} x_{i+1}, \ldots, a_n x_n) \Rightarrow a_0 q \\
(ii) & \quad \text{for } a_0 \text{ and } b_0 \text{ of (i) it holds that } a_0 \in A_0 \text{ if and only if } b_0 \in A_0.
\end{align*}
\]

A transducer \( A \in \mathcal{H}(q) \) is called reduced if \( \theta_A \) is the equality relation on \( A \). A transducer \( A \subseteq \mathcal{H}(q) \) is always reduced.

In contrast with the top-down case there are nonisomorphic but equivalent minimal transducers in \( \mathcal{H}(q) \). However, if a bottom-up transducer is minimal in \( \mathcal{H}(q) \) then it is both reduced and connected (i.e. it has not proper subtransducers). The converse is not true in general.

According to the above discussion we need some further restrictions to guarantee the uniqueness of minimal realizations. For this purpose we introduce the subclass \( \mathcal{H}'(q) \) of \( \mathcal{H}(q) \). A bottom-up transducer \( A = (F, A, G, A_0, \Sigma) \in \mathcal{H}'(q) \) belongs to \( \mathcal{H}'(q) \) if and only if it satisfies the condition:
\[
\text{if } f(a_1 x_1, \ldots, a_n x_n) \rightarrow ap \in \Sigma \text{ where } n \geq 0, f \in F_n, \quad a_1, \ldots, a_n, \quad a \in A \text{ and } p \in T_{G,n}
\]
then \( p \in T_{G,n} \) and none of the operational symbols in \( G_0 \) occurs in \( p \).

Now we are able to state an analogue of Theorem 15 for bottom-up transducers.

**Theorem 21.** Let \( A, B \in \mathcal{H}'(q) \) be connected. Then they are equivalent if and only if \( A/\theta_A \equiv B/\theta_B \).
Proof. The sufficiency follows in the same way as in Theorem 14. In order to prove the necessity of our statement, first observe that if $A=(F, A, G, A_0, \Sigma)$ and $B=(F, B, G, B_0, \Sigma')$, moreover, $\tau_{A(o)}(p)=q$ where $p\in T_F$, $q\in T_G$ and $a\in A$, then there is a state $b\in B$ with $\tau_{B(o)}(p)=q$. In fact, if $a_i\in A$, $b_i\in B$ ($i=1, \ldots, n$, $n>0$) are such that $\text{dom } \tau_{A(o)} \cap \text{dom } \tau_{B(o)} \neq \emptyset$ ($i=1, \ldots, n$) and $p(a_1x_1, \ldots, a_nx_n) \Rightarrow_A aq$ where $p\in T_{F,n}$, $q\in T_{G,n}$ and $a\in A$ then there is a state $b\in B$ satisfying $p(b_1x_1, \ldots, b_nx_n) \Rightarrow_B bq$. The same assertions holds if we change the role of $A$ and $B$. By these observations it is easy to verify that the correspondence $\phi>\theta_A\theta_B$ defined by

$$\phi(\theta_A(a))=\theta_B(b)$$

if and only if $\text{dom } \tau_{A(o)} \cap \text{dom } \tau_{B(o)} \neq \emptyset$ is an isomorphism of $A/\theta_A$ into $B/\theta_B$.

Theorem 22. A bottom-up transducer is minimal in $\mathcal{X}(\phi)$ if and only if it is both reduced and connected. The minimal realization of a bottom-up transducer in $\mathcal{X}(\phi)$ is unique up to isomorphism.

Proof. Immediate by Theorem 21.

Observe that Theorem 22 holds for every class $\mathcal{X} \subseteq \mathcal{X}(\phi)$ provided it is closed under the formation of subtransducers and homomorphic images. An example of a class of this sort is the class of all bottom-up relabelings satisfying condition (iii) in the definition of $\mathcal{X}(\phi)$. A tree transducer $A=(F, A, G, A_0, \Sigma)$ is called a bottom-up relabeling if each rule in $\Sigma$ is of form

$$f(a_1x_1, \ldots, a_nx_n) \rightarrow ag(x_1, \ldots, x_n)$$

where $n\geq 0$, $f\in F_n$, $g\in G_n$, $a_1, \ldots, a_n, a\in A$.

The following example shows that there is a transducer which is minimal in $\mathcal{X}(\phi)$ but which is not minimal in the class of all bottom-up transducers. Let $F_0=\{\#\}$, $F_1=\{f, g\}$ and $F_i=\emptyset$ if $i>1$. Take the bottom-up transducer $A=(F, [5], F, [1], \Sigma)$ where $\Sigma$ consists of the following rules:

(1) $\# \rightarrow 1\#$,
(2) $f(1x_1) \rightarrow 2f(x_1)$, $g(1x_1) \rightarrow 3g(x_1)$,
(3) $f(2x_1) \rightarrow 4f(x_1)$, $g(2x_1) \rightarrow 4f(x_1)$,
(4) $f(3x_1) \rightarrow 4g(x_1)$, $g(3x_1) \rightarrow 4g(x_1)$,
(5) $f(4x_1) \rightarrow 5f(x_1)$, $g(4x_1) \rightarrow 4g(x_1)$,
(6) $f(5x_1) \rightarrow 1f(x_1)$, $g(5x_1) \rightarrow 1g(x_1)$.

It is easy to see that $A$ is minimal in $\mathcal{X}(\phi)$ where $\phi$ is a constant mapping: $\phi(n)=1$ for all $n\geq 0$. On the other hand $\tau_A$ can be induced by a four state transducer $B=(F, [4], F, [1], \Sigma')$ where $\Sigma'$ consists of the rules (1)—(5) listed below:

(1) $\# \rightarrow 1\#$,
(2) $f(1x_1) \rightarrow 2f(f(x_1))$, $g(1x_1) \rightarrow 2g(g(x_1))$,
(3) $f(2x_1) \rightarrow 3x_1$, $g(2x_1) \rightarrow 3x_1$,
(4) $f(3x_1) \rightarrow 4f(x_1)$, $g(3x_1) \rightarrow 4g(x_1)$,
(5) $f(4x_1) \rightarrow 1f(x_1)$, $g(4x_1) \rightarrow 1g(x_1)$.
Decidability results concerning tree transducers I.

Theorem 23. Let \( A=(F, A, G, A_0, \Sigma) \) be minimal in \( \mathcal{K}(q) \). Assume that \( A=A_0 \). Then \( A \) is minimal in the class of all bottom-up transducers.

Proof. Let us correspond to each \( a \in A \) a tree \( p_a \in \text{dom} \tau_{A(a)} \). This can be done because \( A \) is connected. Assume that \( B=(F, B, G, B_0, \Sigma') \) is equivalent to \( A \) and has fewer states than \( A \), i.e. \( |B|<|A| \). Of course \( B=B_0 \). Define the mapping \( \varphi : A \rightarrow B \) by \( \varphi(a)=b \) if and only if \( p_a \in \text{dom} \tau_{B(b)} \). Since \( |B|<|A| \) there are distinct states \( a_1, a_2 \in A \) with \( \varphi(a_1)\neq\varphi(a_2) \). Denote this state \( \varphi(a) \) by \( b \). As \( A \) is reduced, there exist \( p \in T_{F,G}, q_1 \neq q_2 \in T_{G,n} (n \geq 0) \) and \( i_0 \in \{ n \} \), as well as states \( c_1, \ldots, c_{i_0-1}, \ldots, c_n, d_1, d_2 \in \mathcal{A} \) such that

\[
\begin{align*}
p(c_1 x_1, \ldots, c_{i_0-1} x_{i_0-1}, a_1 x_{i_0}, c_{i_0+1} x_{i_0+1}, \ldots, c_n x_n) & \rightarrow A d_1 q_1, \\
p(c_1 x_1, \ldots, c_{i_0-1} x_{i_0-1}, a_2 x_{i_0}, c_{i_0+1} x_{i_0+1}, \ldots, c_n x_n) & \rightarrow A d_2 q_2.
\end{align*}
\]

Of course \( q_1, q_2 \in T_{G,n} \).

As \( A \in \mathcal{K}(q) \) we may assume that \( p=f(x_1, \ldots, x_n) \) for an operational symbol \( f \in F_n \). It can be seen, by \( q_1 \neq q_2 \) and \( A \in \mathcal{K}(q) \), that \( q_1 \) and \( q_2 \) are of form \( q_1=q_0(r_1, \ldots, r_m) \) and \( q_2=q_0(r'_1, \ldots, r'_m) \), respectively, where \( q_0 \in T_{G,m} (m \geq 0), r_j, r'_j \in T_{G,n} \). More exactly, we may choose \( q_0 \) in such a way that \( r_j = b_1(s_1) \) and \( r'_j = b_2(s_2) \) hold for some vectors \( s_1, s_2 \) and different operational symbols \( g_1, g_2 \in G \). This implies that

\[
\tau_A(f(p_{c_1}, \ldots, p_{c_{i_0-1}}, p_{c_{i_0}}, p_{c_{i_0+1}}, \ldots, p_{c_n})) \neq \tau_A(f(p_{c_1}, \ldots, p_{c_{i_0-1}}, p_{c_{i_0}}, p_{c_{i_0+1}}, \ldots, p_{c_n})).
\]

Now let \( b_1=\varphi(c_1) \), \( i=1, \ldots, n, i \neq i_0 \). There is a state \( e \in E \) and a tree \( q \in T_{G,n} \) with \( f(b_1 x_1, \ldots, b_{i_0-1} x_{i_0-1}, b x_{i_0}, b_{i_0+1} x_{i_0+1}, \ldots, b_n x_n) \rightarrow e g \in \Sigma' \). Since \( A \) and \( B \) are equivalent we have \( \tau_A(p_n)=\tau_B(p_n) \) (\( i=1, \ldots, n, i \neq i_0 \)), \( \tau_A(p_{c_{i_0}})=\tau_B(p_{c_{i_0}}) \) (\( i=1,2 \)), \( q_1(\tau_A(p_{c_{i_0}})), \ldots, \tau_A(p_{c_{i_0}}), \ldots, \tau_A(p_{c_{i_0+1}}), \ldots, \tau_A(p_{c_{n}}))=q(\tau_B(p_{c_{i_0}}), \ldots, \tau_B(p_{c_{i_0+1}}), \ldots, \tau_B(p_{c_n})) \).

But \( \tau_A(f(p_{c_1}, \ldots, p_{c_{i_0-1}}, p_{c_{i_0}}, p_{c_{i_0+1}}, \ldots, p_{c_n})) \neq \tau_B(f(p_{c_1}, \ldots, p_{c_{i_0-1}}, p_{c_{i_0}}, p_{c_{i_0+1}}, \ldots, \ldots, p_{c_n})) \). Thus \( \tau_B(p_{c_{i_0}}) \neq \tau_B(p_{c_{i_0}}) \) and path\( _i(q) \neq \emptyset \). Even more, by \( r_j \neq r'_j \), there is a string \( w \in \text{path}_{i_0}(q) \) which is a prefix of path\( _j(q_0) \). Now there are two cases.

First suppose that path\( _j(q_0) \) is a prefix of path\( _i(q_1) \) and let \( p_i = f(p_{c_1}, \ldots, p_{c_{i_0-1}}, p_{c_{i_0}}, p_{c_{i_0+1}}, \ldots, p_{c_n}) \). Then \( \tau_A(p_{c_1})=u(\tau_A(p_{c_{i_0}})) \) and \( \tau_B(p_{c_1})=u' \tau_B(p_{c_{i_0}}) \) where \( u, u' \in T_{F} \) satisfy \( \text{path}(u) = \text{path}_{i_0}(q_1) \) and \( \text{path}(u') = w \), respectively. As \( w \) is a proper prefix of path\( _i(q_1) \) and \( \tau_A(p_{c_1})=\tau_B(p_{c_{i_0}}) \) this results that \( \tau_A(p_{c_1}) \neq \tau_B(p_{c_1}) \), contrary to our assumption \( \tau_A=\tau_B \). A similar argument yields a contradiction if path\( _j(q_0) \) is assumed to be a prefix of path\( _i(q_2) \).

Thus none of the strings path\( _i(q_1) \) and path\( _i(q_2) \) is a postfix of path\( _j(q_0) \). This implies that \( \tau_A(p_{c_1})=u(v), \tau_A(p_{c_2})=u'(v), \tau_B(p_{c_1})=u(v) \) and \( \tau_B(p_{c_2})=u'(v) \) where \( u, u' \in T_{F} \), \( v, v' \in T_{G} \) satisfy the conditions \( \text{path}(u) = \text{path}(u') = w \) and \( v \neq v' \). Indeed, \( v = \tau_B(p_{c_{i_0}}) \) and \( v' = \tau_B(p_{c_{i_0}}) \). It is again a contradiction.
References


(Received May 8, 1980)