The invertibility of tree transducers

IMRE NEUMÜLLER

Every finite automaton can be considered to be a finite algebra equipped with unary operations. In this setting, automata process unary polynomial symbols. This observation led to introducing tree automata by dropping the unary requirement. Basically, tree automata are finite universal algebras, and tree automata process polynomial symbols, i.e. trees. Similar generalization when applied to sequential machines leads to the concept of tree transducers. In the first section of the present paper we recall some basic definitions on tree transducers and prove a few simple propositions. The invertibility of frontier-to-root tree transducers is discussed in the second section. Namely, we give a necessary and sufficient condition describing frontier-to-root tree transducers posessing an inverse. In addition, an algorithm is given for constructing inverse transducers. Similar results are formulated in the last section for root-to-frontier tree transducers.

1. Notions and notations

In this section we recall concepts and results in connection with trees and forests.

Definition 1.1. An operation domain F is a disjoint union of sets F_n indexed by nonnegative integers. F_n is the set of n-ary operational symbols. A finite operation domain is called ranked alphabet.

Definition 1.2. Let X_n be a set of n variables. An F-polynomial symbol over X_n is called an F-tree over X_n. A set $T \subseteq T_F(X_n)$ is called an F-forest over X_n.

Let X be a finite set of variables and $p \in T_F(X)$.

Definition 1.3. The set of all subtrees of p, denoted sub (p), is defined as follows:

(1) if $p \in F_0 \cup X$, then $\mathrm{sub}(p) = \{p\}$,

(2) if $p = f(p_1, p_2, ..., p_m)$ $(f \in F_m, m > 0)$, then

$$\sup (p) = \{p\} \cup (\mathrm{sub}(p_i); \quad i = 1, ..., m).$$

Definition 1.4. The root of p, denoted root (p), is given by the following two conditions:

(1) if $p \in F_0 \cup X$, then $\text{root}(p) = p$;

(2) if $p = f(p_1, \ldots, p_m)$ $(f \in F_m, m > 0)$, then $\text{root}(p) = f$.

Definition 1.5. The height $h(p)$ of p is defined by

(1) if $p \in F_0 \cup X$, then $h(p) = 0$,

(2) if $p = f(p_1, \ldots, p_m)$ $(f \in F_0, m > 0)$, then

$$h(p) = \max \{h(p_i);\ i = 1, \ldots, m\} + 1.$$

Definition 1.6. Let F be a ranked alphabet, and X a finite set of variables. The system $\mathbf{A} = (\mathfrak{A}, X, \mathcal{L}, A')$ is called an n-ary F-automaton, where

(1) $\mathfrak{A} = (A, F)$ is a finite F-algebra,
(2) $\mathcal{L}: X \to A$ is the initial assignment,
(3) $A' \subseteqq A$ is the set of final states.

Let $\hat{\mathcal{L}}: T_F(X) \to \mathfrak{A}$ denote the homomorphic extension of $\mathcal{L}$, where $T_F(X)$ is now considered to be the absolutely free F-algebra generated by X. The forest recognized by $\mathbf{A}$ is defined by:

$$T(\mathbf{A}) = \{p \in T_F(X);\ p\hat{\mathcal{L}} \in A'\}.$$

A forest T is called recognizable (regular) if there is an F-automaton $\mathbf{A}$ with $T(\mathbf{A}) = T$.

In defining tree transducers we shall make use of a set $Z = \{z_1, z_2, \ldots\}$ of auxiliary variables. We set $Z_n = \{z_1, \ldots, z_n\}$ $(n > 0)$. Z is supposed to be disjoint with every other set.

Definition 1.7. A system $\mathbf{A} = (T_F(X_n), A, T_G(Y_m), A', P)$ is called a frontier-to-root tree transducer (F-transducer), where

(1) F and G are ranked alphabets,
(2) A is a ranked set containing only unary operational symbols, the state set of $\mathbf{A}$. (It is assumed that A is disjoint with all other sets in the definition of $\mathbf{A}$, expect A'.)
(3) $A' \subseteqq A$ is the set of final states,
(4) P is a finite set of rewriting rules of the following two types:

(i) $x_i \to a(q)$ $(x_i \in X_n,\ a \in A,\ q \in T_G(Y_m))$ and

(ii) $f(a_1(z_1), \ldots, a_k(z_k)) \to a(q(z_1, \ldots, z_k))$ $(f \in F_k;\ k \geqq 0;\ a_1, \ldots, a_k,\ a \in A;\ z_1, \ldots, z_k \in Z_k;\ q(z_1, \ldots, z_k) \in T_G(Y_m \cup Z_k))$.

In what follows, if $a \in A$ and t is a tree, instead of $a(t)$ we shall use the notation at. Accordingly, we write AT for the set $AT = \{at \mid a \in A;\ t \in T\}$ if T is any forest.

Let $\mathbf{A}$ be the above defined F-transducer. It is said that $\mathbf{A}$ is

— linear, if every $z_i \in Z$ occurs at most once on the right side of a rewriting rule,

— nondeleting, if in every rewriting rule, all the variables z_i occurring on the left side occur on the right side, too,

— completely defined, if for every $i(\leqq n)$, $l(\geqq 0)$, $f(\in F_l)$, and $a_1, \ldots, a_l(\in A)$ there are a rewriting rule with left side x_i and a rewriting rule with left the side $f(a_1 z_1, \ldots, a_l z_l)$.

We are now going to define the tree transformation induced by an F-transducer **A**. Let $p, q \in T_F(X_n \cup AT_G(Y_m \cup Z))$ be arbitrary trees, and **A** the tree transducer given in Definition 1.7. We say that p directly derives q in **A**, if q can be obtained from p

(i) by substituting $a\bar{q}$ for an occurrence of x_i in p provided that $x_i \to a\bar{q}$ is a rewriting rule in P,

(ii) or by substituting $a\bar{q}(p_1, \ldots, p_k)$ for an occurrence of a subtree of the form $f(a_1 p_1, \ldots, a_k p_k)$, provided that $f(a_1 z_1, \ldots, a_k z_k) \to a\bar{q}(z_1, \ldots, z_k)$ is a rewriting rule in P. We use the notation $\underset{\mathbf{A}}{\Rightarrow}$ for direct derivation. The reflexive-transitive closure of $\underset{\mathbf{A}}{\Rightarrow}$ is denoted $\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}$. If $p \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} q$ we say that p derives q in **A**.

Definition 1.8. The F-transformation induced by the F-transducer **A** is the following relation $\tau_{\mathbf{A}}$:

$$\tau_{\mathbf{A}} = \{(p, q) | p \in T_F(X_n), \quad q \in T_G(Y_m), \quad p \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} aq, \quad a \in A'\}.$$

Definition 1.9. The system $\mathbf{A} = (T_F(X_n), A, T_G(Y_m), A', P)$ is called a root-to-frontier tree transducer, R-transducer for short, if

(1) F, G and A are as in Definition 1.7,

(2) $A' \subseteq A$ is the set of initial states,

(3) P is a finite set of rewriting rules of one of the following two forms:

(i) $ax_i \to q$ $(x_i \in X_n, \ a \in A, \ q \in T_G(Y_m))$ and

(ii) $af(z_1, \ldots, z_k) \to q$ $(f \in F_k; \ k \geqq 0; \ a \in A; \ z_1, \ldots, z_k \in Z_k; \ q \in T_G(Y_m \cup AZ_k))$.

Linear, nondeleting and completely defined R-transducers are defined in a way analogous to the F-transducer case.

To define the transformation induced by the above R-transducer we define the direct derivation $\underset{\mathbf{A}}{\Rightarrow}$ in **A** for trees $p, q \in T_G(AT_F(X_n \cup Z) \cup Y_m)$ as follows: $p \underset{\mathbf{A}}{\Rightarrow} q$ if and only if either

(i) q is obtained from p by substituting $\bar{q}$ for an occurrence of a subtree ax_i in p provided that $ax_i \to \bar{q}$ is a rewriting rule in P, or

(ii) q is obtained from p by substituting $\bar{q}(p_1, \ldots, p_k)$ for an occurrence of a subtree $af(p_1, \ldots, p_k) \in \text{sub}(p)$ provided that $af(z_1, \ldots, z_k) \to \bar{q}$ is a rewriting rule in P.

The reflexive transitive closure of $\underset{\mathbf{A}}{\Rightarrow}$ is again denoted $\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}$. If $p \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} q$, we say that p derives q in **A**.

Definition 1.10. The root-to-frontier tree transformation (R-transformation) induced by **A** is the binary relation:

$$\tau_{\mathbf{A}} = \{(p, q) | p \in T_F(X_n); \ q \in T_G(Y_m); \ ap \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} q; \ a \in A'\}.$$

Definition 1.11. Let **A** and **B** be R-transducers (F-transducers). It is said that **A** is equivalent to **B** if $\tau_{\mathbf{A}}=\tau_{\mathbf{B}}$.

Definition 1.12. An F-transducer **A** (R-transducer) is called bounded, if $\tau_{\mathbf{A}}^{-1}(q)$ is a finite set for every $(p, q)\in\tau_{\mathbf{A}}$.

Let **A** be an arbitrary F-transducer. A state $a\in A$ is accessible if there is a tree $p\in T_F(X_n)$ with $p\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}aq$ for some $q\in T_G(Y_m)$. In this case we also say p leads to a. Similarly, we say that a state $a'\in A$ is accessible from a state $a\in A$ if there are $p\in T_F(X_n\cup Z_1)$ and $q\in T_G(Y_m\cup Z_1)$ with $p(az_1)\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}a'q$.

Definition 1.13. An F-transducer is called biaccessible if every state a is accessible, and for every state a there is a final state a' such that a' is accessible from a. (In other words, this means that every state occurs in a derivation $p\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}aq$ where $p\in T_F(X_n)$, $q\in T_G(Y_m)$ and $a\in A'$.)

It is easily seen that for every F-transducer **A** there is an equivalent biaccessible F-transducer provided that $\tau_{\mathbf{A}}\neq\emptyset$.

Let **A** be an arbitrary R-transducer. A state $a\in A$ is called essential if there are $p\in T_F(X_n)$ and $q\in T_G(Y_m)$ with $ap\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}q$. A rewriting rule

$$af(z_1, \ldots, z_k) \to r \tag{1}$$

is called essential if there are $p_1, \ldots, p_k\in T_F(X_n)$ and $q\in T_G(Y_m)$ with $af(p_1, \ldots, p_k)\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}q$ such that in the course of the derivation the first rule applied is (1).

Definition 1.14. An R-transducer **A** is called biaccessible if its states and rewriting rules are essential, further, every state occurs in a derivation $ap\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}q$ where $p\in T_F(X_n)$, $q\in T_G(Y_m)$ and $a\in A'$.

Again, it is straightforward to prove that for every R-transducer **A** there is an equivalent biaccessible R-transducer provided that $\tau_{\mathbf{A}}\neq\emptyset$.

Definition 1.15. Let **A** be an F-transducer. A state $a\in A$ is said to be of k-type for $k=0, \ldots, \infty$, if there are exactly k distinct trees leading to a.

Lemma 1.16. Let **A** be a biaccessible F-transducer. There exists a biaccessible F-transducer **B** which is equivalent to **A** and such that every state of **B** is either of 1-type or of ∞-type.

Proof. Since **A** is biaccessible **A** does not have 0-type states. If **A** has only 1-type or ∞-type states set $\mathbf{B}=\mathbf{A}$. Assume that $a\in A$ is of k-type with $1<k<\infty$. There are $p_1, \ldots, p_k$ $(p_i\in T_F(X_n);\quad i=1, \ldots, k)$ and $q_{j1}, \ldots, q_{jt_j}$ $(q_{jl}\in T_G(Y_m)\quad j=1, \ldots, k;$ $l=1, \ldots, t_j)$ with $p_j\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}aq_{jl}$. (The trees p_j and q_{jt} can be determined in an effective way.) Let $A_i=\{p_i, p_i^1, \ldots, p_i^{s_i}\}$ $(i=1, \ldots, k)$ denote the set of subtrees of p_i. In what follows, $\bar{p}_i$ and $\bar{p}_i^j$ will denote states. Take the following sets of rewriting rules P_i $(i=1, \ldots, k)$: (Let r be an arbitrary tree in $T_G(Y_m)$.)

$$\text{if} \quad A_i = \{p_i\} \quad \text{then} \quad v \to \bar{p}_i q_{im} \in P_i \Leftrightarrow v = p_i; \quad v \in X_n \cup F_0$$
$$m = 1, \ldots, t_i,$$
$$\text{if} \quad A_i \neq \{p_i\} \quad \text{then} \quad v \to \bar{p}_i^j r \in P_i \Leftrightarrow v = p_i^j; \quad v \in X_n \cup F_0;$$
$$f_h(\bar{p}_i^{j_1} z_1, \ldots, \bar{p}_i^{j_h} z_h) \to \bar{p}_i^u r \in P_i \quad \text{if and only if} \quad f_h(p_i^{j_1}, \ldots, p_i^{j_h}) = p_i^u$$
$$f_h(\bar{p}_i^{j_1} z_1, \ldots, \bar{p}_i^{j_h} z_h) \to \bar{p}_i q_{im} \in P_i \quad \text{if and only if} \quad f_h(p_i^{j_1} \ldots, p_i^{j_h}) = p_i$$
$$h > 0; \; m = 1, \ldots, t_i; \; u, j_1, \ldots, j_h \in \{1, \ldots, s_i\}.$$

Let P^* consist of all those rules of P not containing an occurrence of the state a as well as the rules formed in the following way: If a rule in P contains an occurrence of a then substitute $p_i (i=1, \ldots, k)$ for a in every possible way. Take the F-transducer $\mathbf{C}=(T_F(X_n), C, T_G(Y_m), C', P')$, where

$$C = A \cup A_1 \cup \ldots \cup A_k - \{a\}$$
$$C' = \begin{cases} A', & \text{if} \quad a \notin A' \\ A' \cup \{\bar{p}_1, \ldots, \bar{p}_k\} - \{a\}, & \text{if} \quad a \in A' \end{cases}$$
$$P' = P_1 \cup \ldots \cup P_k \cup P^*.$$

It is easy to see that **A** is equivalent to **C** and **C** has fewer states of type k, $1 < k < \infty$, than **A**. In a finite number of steps we arrive at the transducer **B** with the required property.

We continue by introducing a few concepts to be used later. Let **A** be an arbitrary F-transducer. A rewriting rule is called jumping provided that it is of the form

$$f(a_1 z_1, \ldots, a_k z_k) \to a z_i \quad (0 < i \leqq k).$$

A state $a \in A$ is called

— deleting state, if there is a rule containing az_i on the left side but z_i does not occure on the right side,

— multiplying state, if there is a rule containing az_i on the left side and z_i occurs at least twice on the right side,

— jumping state, if there is a jumping rule containing az_i on the left side, and the right side is of the form bz_i for some b, i.e., the right side contains the variable corresponding to a on the left side of the rule.

A chain $a_1, a_2, \ldots, a_k$ $(k>0)$ of states is called a jumping cycle if for every $i=1, \ldots, k$ there is a jumping rule containing a_i on the left side and such that its right side is $a_{i+1} z_j$ where z_j is the variable corresponding to a_i on the left side of the rule. If $i=k$, $a_{i+1}=a_1$. For the sake of simplifying the treatment, if the left side of a rule contains a state a of k-type, then the auxiliary variable corresponding to a is called of k-type, too.

Let **A** be an arbitrary R-transducer. A rewriting rule is called a jumping rule provided that it is of the form $af(z_1, \ldots, z_k) \to a' z_i$. A state a is said to be

— jumping state, if there is a jumping rule whose left side contains a. A chain $a_1, \ldots, a_k$ is a jumping cycle if for every i there is a jumping rule with left side containing a_i and right side containing a_{i+1}. Again, $a_{k+1}=a_1$.

Later we shall use the following notation: $\mathbf{A}(a_i)$ is the R-transducer obtained from $\mathbf{A}$ by letting a_i to be the unique initial state.

The proof of the next result can be found in [3].

Theorem 1.17. For every linear nondeleting F-transducer there is an equivalent linear nondeleting R-transducer and conversely.

Definition 1.18. Let $\mathbf{A}$ be an arbitrary F-transducer (R-transducer). An F-transducer (R-transducer) $\mathbf{B}$ is called an inverse of $\mathbf{A}$, if $\tau_{\mathbf{A}}^{-1}=\tau_{\mathbf{B}}$.

2. The invertibility of F-transducers

In what follows we shall always assume that the F-transducers to be considered are biaccessible with states 1-type or ∞-type. By the previous section this assumption does not restrict the generality of the treatment except for the induced transformation is the empty relation — however, in this case the inverse is obviously inducable.

First let us discuss some necessary conditions of invertibility.

Theorem 2.1. Let $\mathbf{A}$ be an arbitrary F-transducer. Then the domain of $\tau_{\mathbf{A}}$ is regular and $\tau_{\mathbf{A}}^{-1}$ preserves regularity.

The proof of the above result can be found in [3].

Lemma 2.2. Let $\mathbf{A}$ be a biaccessible F-transducer. If $\mathbf{A}$ is invertible then $\tau_{\mathbf{A}}$ preserves regularity.

Proof. The statement is obvious by Theorem 2.1.

Lemma 2.3. Let $\mathbf{A}$ be a biaccessible F-transducer. If $\mathbf{A}$ is invertible then $\mathbf{A}$ is bounded.

Proof. If $\mathbf{A}$ is not bounded then there are an infinite number of trees mapped to the same tree q under $\tau_{\mathbf{A}}$. Thus, q has an infinite number of images under $\tau_{\mathbf{A}}^{-1}$. This contradicts the invertibility of $\mathbf{A}$.

Lemma 2.4. Let $\mathbf{A}$ be an arbitrary biaccessible F-transducer. $\mathbf{A}$ is bounded if and only if

(1) $\mathbf{A}$ has no jumping cycle of states and
(2) $\mathbf{A}$ has no deleting state of ∞-type.

Proof. Let $\mathbf{A}=(T_F(X_n), A, T_G(Y_m), A', P)$. Assume that $a\in A$ is a deleting state of ∞-type. Let $r_1, r_2, \ldots$ be distinct trees in $T_F(X_n)$ with $r_i \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} aq_i\ (i=1, 2, \ldots)$, $q_i\in T_G(Y_m)$. As $\mathbf{A}$ is biaccessible, there are $p\in T_F(X_n)$, $q\in T_G(Y_m)$ such that $(p, q)\in\tau_{\mathbf{A}}$, and in the derivation of q from p we go trough the state a at a stage where the subtree r belonging to a is deleted. Let us replace the subtree r in p by $r_1, r_2, \ldots$; respectively. For the trees $p_1, p_2, \ldots$ obtained in this way we have $(p_i, q)\in\tau_{\mathbf{A}}$.

Suppose now that $a_1, \ldots, a_k\ (k>0)$ is a jumping cycle of states, and let $s_1, \ldots, s_k$ be the corresponding jumping rules. Put $r_1=z$, where $z\in Z$ is an auxiliary variable. Having defined $r_l\ (l=1, \ldots, k)$, form r_{l+1} in the following way. Let s_l be the rule

$f(a^1z_1, \ldots, a_lz, \ldots, a^uz_u) \to a_{l+1}z$. (Of course, $a_{k+1}=a_1$) Take a tree $t^i \in T_F(X_n)$ with $t^i \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} a^iq^i$ where $q^i \in T_G(Y_m)$ for every $i=1, \ldots, u$. Put

$$r_{l+1} = f(t^1, \ldots, r_l, \ldots, t^u) \quad l = 1, \ldots, k.$$

Denote by p^* the tree r_{k+1}. Obviously, $h(p^*)>0$. Since **A** is biaccessible there are $p \in T_F(X_n)$, $q \in T_G(Y_m)$ with $(p, q) \in \tau_{\mathbf{A}}$, and such that a derivation of q from p goes through a_1. Denote by $\bar{p}$ that subtree of p leading to a_1. Put

$$\bar{p}_0 = \bar{p},$$

$$\bar{p}_{i+1} = p^* \cdot_z \bar{p}_i,$$

where $\cdot_z$ denotes the z-product of trees. Substitute $\bar{p}_i$ for $\bar{p}$ in p, and let p_i $(i=0, 1, \ldots)$ be the resulting tree. Obviously $(p_i, q) \in \tau_{\mathbf{A}}$, ending the proof of the necessity.

Conversely, if both (1) and (2) are satisfied by **A**, then for every $(p, q) \in \tau_{\mathbf{A}}$ it holds that $h(q) > (h(p)-m)/(k+1)$, where $m = \max \{h(r);\ r \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} aq, q \in T_G(Y_m),\ a \in A$ is a deleting state$\}$, and k is the cardinality of the state set. From this, **A** is easily seen to be bounded.

Next we try to construct an inverse transducer by "inverting the rules", if it is possible.

Definition 2.5.

— The inverse of a rule $x_i \to aq$ or $f_0 \to aq\,(f_0 \in F_0)$ is a finite set of rules ensuring $q \overset{*}{\Rightarrow} ax_i$ or $q \overset{*}{\Rightarrow} af_0$.

— The inverse of a rule $f(a_1z_1, \ldots, a_kz_k) \to aq$ $k>0$, $(q \in T_G(Y_m \cup Z_k)$ $q \neq z_i$ $(i=1, \ldots, k)$ is a finite set of linear rules in which none of the auxiliary variables occure (the auxiliary variables are denoted e.g. by $v_1, v_2, \ldots$), and such that these rules realize a derivation $q(a_1z_1, \ldots, a_kz_k) \overset{*}{\Rightarrow} af(r_1, \ldots, r_k)$ where the variables z_i occure with the same multiplicity and with the same states as in the original rules and

$$r_i = \begin{cases} z_i & \text{if} \quad a_i \quad \text{is of } \infty\text{-type} \\ p_i & \text{if} \quad a_i \quad \text{is of 1-type and } p_i \text{ leads to } a_i. \end{cases}$$

— Further, we say that some occurrences of the variables $z_1, \ldots, z_i$ meet at the same vertex in a tree $q \in T_G(Y_m \cup Z)$ if q has a vertex such that there is a path in q from that vertex to every given occurrence of the variables $z_j\,(j=1, \ldots, i)$ and there is no edge in q belonging to two different such paths.

Lemma 2.6. Suppose that the states occuring in the rules are of 1-type or of ∞-type and none of them is an ∞-type deleting state. Then

(1) all the rules $x_i \to aq$, $f_0 \to aq$ $(f_0 \in F_0;\ q \in T_G(Y_m))$ are invertible.

(2) a linear rule $f(a_1z_1, \ldots, a_kz_k) \to aq\,(k>0;\ q \in T_G(Y_m \cup Z_k);\ q \neq z_i;\ i=1, \ldots, k)$ is invertible if and only if all ∞-type auxiliary variables in q meet at same vertex,

(3) a nonlinear rule $f(a_1z_1, \ldots, a_kz_k) \to aq$ $(k>0; q \in T_G(Y_m \cup Z_k))$ is invertible if and only if each ∞-type auxiliary variable in q has an occurrence such that these occurrences meet at the same vertex.

Proof. Suppose that a rule is invertible. There is a uniquely determined vertex in q such that we get back the given occurrence of the symbol f during the derivation process. Of the arguments of f, all ∞-type auxiliary variables occur. Since none of the auxiliary variables z_i occurs in the rules realizing the inverse of the rule, the vertex in question has an outgoing path to an occurrence of every ∞-type auxiliary variable and no edge belongs to more than one such path. We have proved the necessity in case of conditions (2) and (3).

(1) In the same way as in the proof of Lemma 1.16 we can construct an F-transducer transforming a given tree to a given tree. Let us replace in this transducer every occurrence of the final state by a, the resulting set of rules is the inverse of the rule given in (1).

(2) Further on the symbols $z_i (i=1, ..., k)$ will not be considered to be auxiliary variables. The auxiliary variables will be denoted by v. Denote by $S=\{s_1, ..., s_h\}$ the set of all those vertices of $q(z_1, ..., z_k)$ from which there are paths leading to ∞-type z_i's. Let $R=\{r_1, ... r_l\}$ be the set of those subtrees of $q(z_1, ..., z_k)$ whose roots are directly attached to vertices in S and which does not contain vertices of S. (Different occurrences are treated separately.) Let $t\in T_F(X_n)$ be an arbitrary tree. For every $j (j=1, ..., l)$ there is an F-transducer $\mathbf{B}_j$ with $\tau_{\mathbf{B}_j}=\{(r_j, t)\}$. Let P_j be the set of rules of $\mathbf{B}_j$. If z_i occurs in a rule then replace all occurrences of the state appearing on the right side with a_i. Delete those rules containing an occurrence of one of the symbols z_i. Denote by P' the union of the sets of rules obtained in this way, and let $c_1, ..., c_l$ be the final states. If S is empty then $R=\{q\}$. In this case put $t=f(p_1, ..., p_h)$ where p_i leads to a_i, and replace c_i with a everywhere — the proof is done. Otherwise, let e denote the vertex where the ∞-type z_i's meet, if there is only one such z_i let e be the root. Let us assign to each s_i a state d_i $(i=1, ..., n)$ as follows: If s_i is the root then $d_i=a$, otherwise let d_i be a new state. Let $g(q_1, ... q_\alpha)$ be the subtree corresponding to the vertex s_i. We construct a rule to every vertex s_i in S. If the vertex is different from e then the rule is:

$$g(b_1 v_1, ..., b_\alpha v_\alpha) \to d_i v_u \quad \text{where}$$

$$b_j = \begin{cases} c_\beta, & \text{if} \quad q_j = r_\beta \\ d_k, & \text{if} \quad q_j \text{ has root } s_k \end{cases}$$

$(j=1, ..., \alpha)$ and v_u is the auxiliary variable corresponding to the unique d occurring on the left side. If the vertex coincides with e then

$$g(b_1 v_1, ..., b_\alpha v_\alpha) \to d_i f(\bar{r}_1, ..., \bar{r}_k) \quad \text{where}$$

$$b_j = \begin{cases} c_\beta, & \text{if} \quad q_j = r_\beta \\ d_u, & \text{if} \quad q_u \text{ has root } s_u \end{cases}$$

$$(j = 1, ..., \alpha)$$

$$\bar{r}_j = \begin{cases} p_j, & \text{if} \quad a_j \text{ is of 1-type and } p_j \text{ leads to } a_j \\ v_u, & \text{if} \quad z_j \text{ is of } \infty\text{-type and occurs in } q_u \end{cases}$$

$$(j = 1, ..., k).$$

Take the union of all the sets of rules costructed. Obviously, this set is an inverse of the original rule.

(3) Choose an occurrence of every ∞-type z_i in such a way that these occurrences meet at the same vertex, and consider these occurrences as ∞-type z_i's. The proof is finished as in (2).

The proof of the next two theorems can be found in [4].

Theorem 2.7. Let **A** be an arbitrary F-transducer. $\tau_{\mathbf{A}}$ preserves regularity if and only if **A** is equivalent to a linear F-transducer.

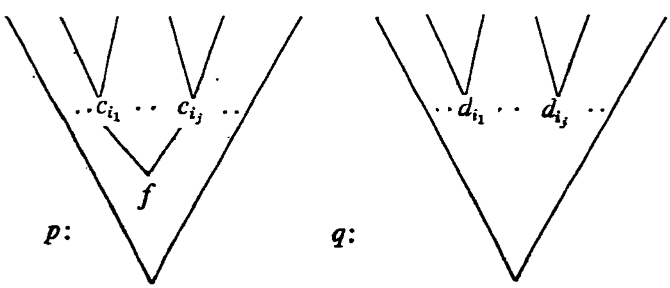
Theorem 2.8. Let **A** be a biaccessible F-transducer. Then **A** is equivalent to a linear F-transducer if and only if **A** is linear or its multiplying states are of finite type (i.e. of k-type with $k<\infty$).

Lemma 2.9. Let **A** be a biaccessible F-transducer. If **A** is invertible then its non-jumping rules are invertible.

Proof. Assume to the contrary that **A** is invertible and has a non-jumping rule which is not invertible. On the basis of the previous results we may assume that **A** is biacessible, linear, and its states are of 1-type or of ∞-type. Denote by **B** an inverse of **A**. Let

$$f(a_1 z_1, \ldots, a_k z_k) \rightarrow a\bar{q}(z_1, \ldots, z_k) \tag{1}$$

be a non-invertible rule. By Lemma 2.6, the ∞-type auxiliary variables do not meet at a single vertex. Because of biaccessibility, there are $p\in T_F(X_n)$ and $q\in T_G(Y_m)$ such that $(p, q)\in\tau_{\mathbf{A}}$ and the rule is used in the course of deriving q from p. Let e denote that vertex of p where the rule (1) is used. Let $z_{i_1}, \ldots, z_{i_j}$ be all the ∞-type auxiliary variables in q with corresponding ∞-type states $a_{i_1}, \ldots, a_{i_j}$. Obviously, $j>2$. Let $c_{i_1}, \ldots c_{i_j}$ denote those verteces in p which are direct descendants of e (i.e. there are edges from e to them) and such that in the course of the derivation we obtain ∞-type states after processing the subtrees belonging to them. The auxiliary variables $z_{i_1}, \ldots, \ldots, z_{i_j}$ correspond to these vertices $c_{i_1}, \ldots, c_{i_j}$. Let $d_{i_t} (t=1, \ldots, j)$ denote the root vertex of the image of the subtree belonging to c_{i_t}.



Since $a_{i_1}, \ldots, a_{i_j}$ are of ∞-type, for every t there are distinct trees p_t^u $(u=1, 2, \ldots; t=1, \ldots, j)$ leading to a_{i_t}. Replacing the subtrees belonging to the vertices $c_{i_1}, \ldots, c_{i_j}$ by $p_1^{u_1}, \ldots, p_j^{u_j}$, respectively, the trees p_n $(n=1, 2, \ldots)$ can be transformed to trees q_n "similar" to q: q_n is obtained from q by replacing the subtrees at $d_{i_1}, \ldots, d_{i_j}$ with images of the trees replacing the subtrees at $c_{i_1}, \ldots, c_{i_j}$ in p. By Lemma 2.3, the set of these trees q_n is infinite, and $(q_n, p_n) \in \tau_B$ by assumption. Classify the vertices of the trees q_n as follows. The vertices of the subtrees replacing the subtrees at $d_{i_1}, \ldots, d_{i_j}$ in q belong to $O_1, \ldots, O_j$, resp. All other vertices form a singleton class. It is plain to see that a "similar" classification is obtained for every n. Since $(q_n, p_n) \in \tau_B$ for all n, therefore every tree q_n has a unique vertex whose translation in **B** gives back that occurrence of the symbol f which is the label of e in p. The above classification is finite, therefore, there is a class containing the vertex in question for infinitely many q_n. With this we have designated an infinite subset of the trees p_n, as well. It is easily seen that there is a class such that the corresponding trees p_n satisfy the following: for each c_{i_t} there is an infinite number of subtrees at c_{i_t}. In what follows we restrict ourselves to this class. Suppose it is one of the singleton classes, i.e. a concrete vertex. If this vertex is not on the paths from the root to one of the d_{i_t}'s then the same tree belongs to this vertex in every q_n. Consequenly, **B** should translate an infinite number of trees from this very same tree, which is impossible. If the vertex is located on a path to one of the vertices $d_{i_1}, \ldots, d_{i_j}$ then, by assumption, at most $j-1$ edges lead from this vertex in the direction of $d_{i_1}, \ldots, d_{i_j}$. Since the designated trees p_n are such that infinitely many independent trees are attached to each of the vertices $c_{i_1}, \ldots, c_{i_j}$, this case leads to a contradiction, too. If the class in question is one of the classes $O_1, \ldots, O_j$ then **B** is not bounded. This derives from the fact that, if e.g. O_1 is the particular class, then an infinite number of trees may be attached to each of the vertices $d_{i_2}, \ldots, d_{i_j}$ but the subtree belonging to this vertex is already obtained during the translation of the subtree belonging to d_{i_1}. It is also impossible because **B** has inverse, namely **A**.

Lemma 2.10. Let **A** be a biaccessible, bounded F-transducer such that τ_A preserves regularity. If the non-jumping rules of **A** are invertible then **A** is invertible.

Proof. We may assume that every state of **A** is of 1-type or of ∞-type and that **A** is linear. By boundedness, **A** does not contain deleting states of ∞-type and jumping cycles. If P has a jumping rule then form all the state chains $a_1, \ldots, a_k$ $(k > 1)$ satisfying the following condition: $a_1, \ldots, a_{k-1}$ are jumping states, a unique jumping rule leads from a_i to a_{i+1} $(i=1, \ldots, k-1)$. In a similar way as we assigned a tree to jumping cycle in the proof of Lemma 2.4 let us assign a tree to every chain in question, as well. We do this in all possible ways. Of course, it may happen that more then one tree is assigned to the same chain (if there are more jumping rules from a_i to a_{i+1}).

Let $a_1, \ldots, a_k$ be a state chain with corresponding tree r. Choose a non-jumping rule leading to a_1, say it is

$$f(a^1 z_1, \ldots, a^l z_l) \to a_1 q. \tag{1}$$

Of course, it can be of the form $x_i \to a_1 q$, too. Form the following formal transition rule:

$$r \cdot_z f(a^1 z_1, \ldots, a^l z_l) \to a_k q, \tag{2}$$

where q is the tree appearing an the right side of (1). Take all possible formal transition rules (2). The inverse of a formal transition rule is meant a finite set of linear rules not containing the auxiliary variables $z_1, \ldots, z_l$ (the auxiliary variables are denoted by v) and such that they realize the derivation $q(a^1 z_1, \ldots, a^l z_l) \overset{*}{\Rightarrow} r \cdot_z f(s_1, \ldots, s_l)$, where the z_i's $(i=1, ..., l)$ occure with the same multiplicity and with the same states as in (2), and

$$s_i = \begin{cases} z_i & \text{if } a^i \text{ is of } \infty\text{-type,} \\ p_i & \text{if } a^i \text{ is of 1-type and } p_i \text{ leads to } a_i. \end{cases}$$

Obviously, we get a finite set of rules and if (1) is invertible then so is (2).

Let $\mathbf{B}=(T_G(Y_m), B, T_F(X_n), B', P')$ be the F-transducer where $B=A\cup\bar{A}$, $\bar{A}$ is the set of the states which are obtained in the inversion process of non-jumping and formal rules,

$B'=A'$,

P' is the set of the rules which are obtained in the inversion process of non-jumping and formal rules. It suffices to show that

$$p \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} aq \Leftrightarrow q \overset{*}{\underset{\mathbf{B}}{\Rightarrow}} ap \quad a\in A;\ p\in T_F(X_n);\ q\in T_G(Y_m).$$

Proof of $\Rightarrow$. If $h(p)=0$ the implication holds by the invertibility of the rules. We proceed by induction of $h(p)$. Suppose the claim for $h(p)<m$ and let $h(p)=m$. If the rule applied for the last time in the derivation $p \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} aq$ is not a jumping rule then $p=f(p_1, ..., p_k)$ $(k>0)$ and

$$f(a_1z_1, ..., a_kz_k) \to a\bar{q}\in P, \quad q = \bar{q}(q_1, ..., q_k), \quad \bar{q} \neq z_i \quad (i = 1, ..., k)$$

$$\text{and} \quad p_j \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} a_jq_j \quad (j = 1, ..., k)$$

for some states $a_1, ..., a_k\in A$. Thus,

$$p = f(p_1, ..., p_k) \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} f(a_1q_1, ..., a_kq_k) \underset{\mathbf{A}}{\Rightarrow} a\bar{q}(q_1, ..., q_k) = aq.$$

By the induction hypothesis, $q_j \overset{*}{\underset{\mathbf{B}}{\Rightarrow}} a_jp_j$ $(j=1, ..., k)$. Since all the non-jumping rules of $\mathbf{A}$ are invertible,

$$q = \bar{q}(q_1, ..., q_k) \overset{*}{\underset{\mathbf{B}}{\Rightarrow}} \bar{q}(a_1p_1, ..., a_kp_k) \overset{*}{\underset{\mathbf{B}}{\Rightarrow}} af(p_1, ..., p_k) = ap.$$

If the rule applied for the last time is a jumping rule then we have "used" a formal rule at the end of the derivation. In this case the derivation can be written as

$$p = \bar{p}(f(p_1, ..., p_k)) \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} \bar{p}(f(a_1q_1, ..., a_kq_k)) \underset{\mathbf{A}}{\Rightarrow}$$

$$\bar{p}(a'\bar{q}(q_1, ..., q_k)) \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} a\bar{q}(q_1, ..., q_k) = aq,$$

where the rule applied in the second part is non-jumping and all the rules applied in the third part are jumping. In this case there is a formal rule $\bar{p}(f(a_1z_1, ..., a_kz_k)) \to$ $\to a\bar{q}(z_1, ..., z_k)$. Formal rules are invertible, therefore,

$$q = \bar{q}(q_1, ..., q_k) \overset{*}{\underset{\mathbf{B}}{\Rightarrow}} \bar{q}(a_1p_1, ..., a_kp_k) \overset{*}{\underset{\mathbf{B}}{\Rightarrow}} ap.$$

Proof of $\Leftarrow$. The proof is accomplished by induction on $h(q)$. Suppose $h(q)=0$. Then, by the construction of the rules in P', either $h(p)=0$ and $p\to aq\in P$, or $p= =f(p_1, \dots, p_k)$ $(k>0)$ and $f(a_1z_1, \dots, a_kz_k)\to aq\in P$; $p_j\overset{*}{\underset{\mathrm{A}}{\Rightarrow}}a_jq_j$, $(j=1, \dots, k)$ where a_j is 1-type. But then $p\overset{*}{\underset{\mathrm{A}}{\Rightarrow}}aq$. Suppose that the proof is done for $h(q)<m$, and let $h(q)=m$. Take the graph representation of q. Let us mark those verteces in the graph at which the derivation gets into a state belonging to A. Denote it by q^*. Denote by q' the maximal connected subgraph of q^* containing the vertex which corresponds to the root of q but not containing any other marked vertex. Let $\bar{q}$ denote that part of q corresponding to q'. We have

$$q=\bar{q}(q_1, \dots, q_k)\overset{*}{\underset{\mathrm{B}}{\Rightarrow}}\bar{q}(a_1p_1, \dots, a_kp_k)\overset{*}{\underset{\mathrm{B}}{\Rightarrow}}a\bar{p}(p_1, \dots, p_k)=ap \quad a_1, \dots, a_k, a\in A.$$

After the first part we will not get back to a state in A only at the root. The inversion of the rule was done in such a way that we introduced new states at each stage, and with the new rules, we could get to a state belonging to A only at the root of the right side of the original rule. From this it follows that either $\bar{p}(a_1z_1, \dots, a_kz_k)\to \to a\bar{q}(z_1, \dots, z_k)\in P$ or there is a formal rule with left side $\bar{p}(a_1z_1, \dots, a_kz_k)$ and right side $a\bar{q}(z_1, \dots, z_k)$. This ends the proof.

The main results of this section easily follows from Lemmas 2.2, 2.3, 2.9, 2.10.

Theorem 2.11. A biaccessible F-transducer A is invertible if and only if A is bounded, the non-jumping rules of A are invertible and τ_{A} preserves regularity.

Theorem 2.12. The invertibility of F-transducers is decidable. There is an effective procedure for constructing inverse F-transducers.

3. The invertibility of R-transducers

Again, we only treat biaccessible R-transducers. The proof of the following theorem can be found in [3].

Theorem 3.1. Let A be an R-transducer. Then the domain of τ_{A} is regular and τ_{A}^{-1} preserves regularity.

Lemma 3.2. Let A be a biaccessible R-transducer. If A is invertible then τ_{A} preserves regularity.

Proof. The statement is trivial by Theorem 3.1.

Lemma 3.3. Let A be a biaccessible R-transducer. If A is invertible then A is bounded.

Proof. Similar to that of Lemma 2.3.

Lemma 3.4. Let A be an arbitrary biaccessible R-transducer. A is bounded if and only if:

(1) A has no jumping cycle of states,
(2) A is nondeleting.

Proof. The proof is similar to that of Lemma 2.4.

Theorem 3.5. Let **A** be an arbitrary biaccessible R-transducer. **A** is invertible if and only if:

(1) $\tau_{\mathbf{A}}$ preserves regularity,
(2) **A** is bounded,
(3) the rewriting rules are of one of the following three types:

(a) $ap \to q \quad a\in A;\ p\in X_n \cup F_0;\ q\in T_{G_0\cup G_1}(Y_m)$,

(b) $af(z_1) \to q \quad a\in A;\ f\in F_1;\ q\in T_{G_1}(AZ_1)$,

(c) $af(z_1, \ldots, z_k) \to q;\quad a\in A;\ f\in F_k \quad (k>1)$;

$q\in T_G(\{v\})\cdot_v g_k(r_1, \ldots, r_k)$, where $g_k\in G_k;\ r_i\in T_{G_1}(AZ_k) \quad (i=1, \ldots, k)$.

Proof. The necessity of the first two conditions directly comes from Lemmas 3.2 and 3.3. Suppose **B** is an inverse of **A**. Then both **A** and **B** are nondeleting. Let $(p, q)\in\tau_{\mathbf{A}}$ be arbitrary. Let $n(p)$ $(n(q))$ denote the number of those vertices of p (q) whose label is an operation symbol with arity at least 2. Since **A** and **B** are nondeleting and $(p, q)\in\tau_{\mathbf{A}}$, $(q, p)\in\tau_{\mathbf{B}}$, we have $n(p)=n(q)$. From this it follows that the rules of **A** are of one of the three types as indicated.

For the converse, suppose that **A** satisfies all the conditions (1), (2), (3). Observe that this assumption implies that **A** is linear and nondeleting. By Theorem 1.14, there is a linear nondeleting F-transducer **C** equivalent to **A**. By the assumptions and the construction given in the proof of Theorem 1.17 the auxiliary variables meet at the same vertex in right side of the rules of **C**. Therefore, we can consider all the states of **C** and all the auxiliary variables to be of ∞-type. Let $f(c_1z_1, \ldots, c_lz_l)\to q$ $(l>0)$ be a rule in **C**. Then, again by the construction given in the proof of Theorem 1.17 and our assumptions, the frontier of q only consists of auxiliary variables. Thus, **C** is invertible. Let us invert the rules in such a way that every auxiliary variable is taken ∞-type. The inverse obtained is a linear nondeleting F-transducer. By Theorem 1.17, it has an equivalent linear nondeleting R-transducer. This ends the proof.

To obtain a complete solution we would need to describe R-transducers equivalent to linear R-transducers. For our purposes, we may confine ourselves to bounded regularity preserving R-transducers.

Lemma 3.6. Let $\mathbf{A}=(T_F(X_n), A, T_G(Y_m), A', P)$ be a biaccessible R-transducer. Suppose that **A** is nonlinear. Let $af(z_1, \ldots, z_l)\to q(\mathbf{a}_1z_1^{n_1}, \ldots, \mathbf{a}_lz_l^{n_l})\in P;\ \mathbf{a}_i\in A^{n_i}$, $(i=1, \ldots, l)$ be a multiplying rule. Especially, $\mathbf{a}_1=(a_1, \ldots, a_{n_1})$. Let $T= \bigcap\{\operatorname{dom}(\tau_{\mathbf{A}(a_i)})|i=1, \ldots, n_1\}$ and $T_i=\tau_{\mathbf{A}(a_i)}(T)$ $(i=1, \ldots, n_1)$, where $\operatorname{dom}(\tau_{\mathbf{A}(a_i)})$ is the domain of the transformation $\tau_{\mathbf{A}(a_i)}$. Thus, to every multipying rule there are corresponding forests T and T_i. If for every T at most one of the associated T_i's is infinite then $\tau_{\mathbf{A}}$ preserves regularity.

Proof. Let $A=\{a_1, \ldots, a_k\}$ be the state set of **A**. Denote by T' the union of the finite T_i's. T' is finite, say $T'=\{q_1, \ldots, q_t\}$ $q_i\in T_G(Y_m)$, $i=1, \ldots, t$. Denote by T_{ij} $(i=1, \ldots, k; j=1, \ldots, t)$ the set of those trees $p\in T_F(X_n)$ with $a_ip\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}q_j$. The forests T_{ij} are regular, consequently, there are tree automata $\mathbf{A}_{ij}=(\mathfrak{A}_{ij}, \mathscr{L}_{ij}, A'_{ij})$

with $T_{ij}=T(\mathbf{A}_{ij})$. Let V denote the set of all vectors of dimension kt over the set $\{0, 1\}$. Let $H=H_0\cup H_1\cup \dots$ be a new ranked alphabet where $H_0=F_0$; $H_l=F_l\times V^l$ $(l>0)$. Take the F-transducer $\mathbf{B}=(T_F(X_n), B, T_H(X_n), B', P')$ with

$$B = A_{11}\times A_{12}\times \dots \times A_{k1}\times \dots \times A_{kt}, \quad \text{where} \quad A_{ij} \text{ is the state set of } \mathbf{A}_{ij},$$

$$B' = B$$

P':

(i) $$x \to (x\mathcal{L}_{11}, \dots, x\mathcal{L}_{kt})\times \in P', \quad x\in X_n,$$

(ii) $$f \to (f^{\mathfrak{A}_{11}}, \dots, f^{\mathfrak{A}_{kt}})\, f\in P', \quad f\in F_0,$$

(iii) $$f(b_1 z_1, \dots, b_l z_l) \to b(f, v_1, \dots, v_l)(z_1, \dots, z_l)\in P'$$

$$f\in F_l;\ l>0;\ b, b_1, \dots, b_l\in B;\ v_i\in V \quad (i=1, \dots, l);$$

where $b=(f^{\mathfrak{A}_{11}}(b_{1,1}; \dots, b_{l,1}), \dots, f^{\mathfrak{A}_{kt}}(b_{1,kt}, \dots, b_{l,kt}))$, and let $0<\alpha\leqq k$, $0<\beta\leqq t$, $v_{ij}=1$ if and only if $b_{i,j}\in A_{\alpha\beta}$, where $j=(\alpha-1)t+\beta$. $\tau_{\mathbf{B}}$ is a linear nondeleting R-transformation by Theorem 1.17.

Let $\mathbf{C}=(T_H(X_n), A, T_G(Y_m), A', P'')$ where A and A' are the same as in $\mathbf{A}$, and P'' is obtained in the following way.

In case of nonlinear rules:

(1) Let $af(z_1, \dots, z_l)\to r(\mathbf{a}^1\mathbf{z}_1, \dots, \mathbf{a}^l\mathbf{z}_l)\in P$ be nonlinear. Take all possible rules $a(f, v_1, \dots, v_l)(z_1, \dots, z_l)\to\bar r$, where $v_i\in V$ $(i=1, \dots, l)$ and $\bar r$ is obtained from r as follows: Let $0<\alpha\leqq k$, $0<\beta\leqq t$. If $v_{ij}=1$ $(j=(\alpha-1)t+\beta)$ and z_i occurs in r with state a_α $(i=1, \dots, l)$, then substitute q_β for $a_\alpha z_i$ in r. Let P'' contain those rules $a(f, v_1, \dots, v_l)(z_1, \dots, z_l)\to\bar r$ which are linear and such that $\bar r$ contains an auxiliary variable if and only if it occurs in r.

In case of linear rules:

(2) $af(z_1, \dots, z_l)\to r\in P$ if and only if

$$a(f, v_1, \dots, v_l)(z_1, \dots, z_l) \to r\in P''; \quad v_i\in V \quad (i=1, \dots, l).$$

(3) $ap\to r\in P$ if and only if $ap\to r\in P''$; $p\in T_{F_0}(X_n)$. We have $\tau_{\mathbf{A}}=\tau_{\mathbf{B}}\circ\tau_{\mathbf{C}}$, yielding that $\tau_{\mathbf{A}}$ preserves regularity.

Corollary 3.7. If $\mathbf{A}$ is nondeleting then so is $\mathbf{C}$. It is known that the composition of linear nondeleting R-transformations is a linear R-transformation. Thus, $\tau_{\mathbf{A}}$ is a linear R-transformation if $\mathbf{A}$ is nondeleting, further, one can effectively construct an equivalent linear R-transducer.

Lemma 3.8. Let $\mathbf{A}$ be a biaccessible bounded nonlinear R-transducer. Let a nonlinear rule of $\mathbf{A}$ be $af(z_1, \dots, z_l)\to\bar q(\mathbf{a}_1 z_1^u, \dots, \mathbf{a}_l z_l^{n_l})$. Suppose that this rule multiplies z_1, i.e. $u>1$. Put $\mathbf{a}_1=(a_1, \dots, a_u)$. Denote by T the forest $T=\bigcap\{\mathrm{dom}(\tau_{\mathbf{A}(a_i)});\ i=1, \dots, u\}$. If $\tau_{\mathbf{A}}$ preserves regularity then T is finite.

Proof. Since $\mathbf{A}$ is bounded so are $\mathbf{A}(a_i)$ and $\mathbf{A}(a)$ $(i=1, \dots, u)$. If $\tau_{\mathbf{A}}$ preserves regularity then $\tau_{\mathbf{A}(a_i)}$ preserves regularity as well. Therefore, it is enough to deal with the transducer $\mathbf{A}(a)$ instead of $\mathbf{A}$. Suppose that T is infinite although the assumptions are fulfilled. Since T is a finite intersection of domains, T is regular. Let $s\in T_F(X_n\cup Z_1)$ be a tree in which z_1 occurs exactly once. Then

$$s^i = \underbrace{s\cdot_{z_1}s\cdot_{z_1}\cdots\cdot_{z_1}s}_{i\text{-times}}.$$

Since T is infinite and regular, there is an $r\in T$ such that $r=r_1(r_2(r_3))$ where $r_3\in T_F(X_n)$, $r_1, r_2\in T_F(X_n\cup Z_1)$, $h(r_2)>0$ and for every k, $r^{(k)}=r_1(r_2^k(r_3))\in T$. Then $T'=\{r^{(k)}; k=1, 2, \ldots\}$ is an infinite regular subset of T. Since $T'\subseteq T$, $T_i=\tau_{\mathbf{A}(a_i)}(T')\neq\emptyset$ $(i=1, \ldots, u)$, and by the boundedness, T_i is infinite. Since $\tau_{\mathbf{A}}$ preserves regularity by supposition, all the forests T_i are regular. Let $p_2, \ldots, p_l\in T_F(X_n)$ be trees with $a_{ij}p_i\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}q_{ij}$ $i=2, \ldots, l$ and $j=1, \ldots, n_i$. Let $p=f(z_1, p_2, \ldots, p_l)$. It holds that $ap\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}\bar{q}(a_1z_1, \ldots, a_uz_1)$. Set $T''=p\cdot_{z_1}T'$ $(=\{p(r_1(r_2^k(r_3))), k=1, 2, \ldots\})$. T'' is regular, and so is $K=\tau_{\mathbf{A}}(T'')$. Let $\mathbf{B}$ be a tree automaton recognizing K. Since all the forests T_i are regular, there is a $q\in K$, $q=\bar{q}(a_1, s_2, \ldots, s_u)$, with $s_i\in T_i$ $(i=1, \ldots, u)$ such that $h(s_1)$, $h(s_l)$ are greater than the number of states of $\mathbf{B}$. Then s_i $(i=1, 2)$ can be written in the form $s_{i_1}(s_{i_2}(s_{i_3}))$ such that $h(s_{i_2})>0$ and $t_i^{(k)}=s_{i_1}(s_{i_2}^k(s_{i_3}))\in T_i$ for every k. We have $q=\bar{q}(t_1^{(k)}, t_2^{(l)}, s_3, \ldots, s_u)\in K$. Let n_1 denote the number of states of $\mathbf{A}$, and n_2 the maximum of the heights of the trees occurring on the right side of the rules in P. Choose $t_1^{(k)}$ and $t_2^{(l)}$ as follows:

$$h(t_1^{(k)}) > n_2h(p(r))+1,$$

$$h(t_2^{(l)}) > n_2(n_1+1)h(t_1^{(k)})+n_2(n_1+1)h(\bar{q}(z_1, z_2, s_3, \ldots, s_u)).$$

Let $q^*=\bar{q}(t_1^{(k)}, t_2^{(l)}, s_3, \ldots, s_u)$, where $t_1^{(k)}, t_2^{(l)}$ are the trees given above. It is obvious that $q^*\in K$. There is a $p_n\in T''$ $(p_n=p(r_1(r_2^n(r_3))))$ with $ap_n\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}q^*$. Denote by d_1 and d_2 the root of $t_1^{(k)}$ and $t_2^{(l)}$, resp. There are vertices c_1, c_2 in p_n such that d_i $(i=1, 2)$ is obtained when the translation process arrives at c_i. The lengths of the paths from the root to c_1 and c_2 cannot exeed $(n_1+1)h(\bar{q}(z_1; z_2, s_3, \ldots, s_u))$ because $\mathbf{A}$ does not have jumping cycles. Let $\bar{r}_i$ $(i=1, 2)$ denote the subtree belonging to c_i. The subtree r_3 cannot contain c_1 and c_2, or even, c_1 and c_2 are located on the path to the root of r_3. (The reason is the height of the trees $t_1^{(k)}, t_2^{(l)}$.) Thus, either $\bar{r}_1$ is a subtree of $\bar{r}_2$ or conversely.

Suppose first that $\bar{r}_1\in\operatorname{sub}(\bar{r}_2)$. Then

$$h(\bar{r}_2) < h(\bar{r}_1)+(n_1+1)h(\bar{q}(z_1, z_2, s_3, \ldots, s_u))$$

because $\bar{r}_1$ is a subtree of $\bar{r}_2$ and that part of $\bar{r}_2$ not contained by $\bar{r}_1$ cannot be higher than $(n_1+1)h(\bar{q}(z_1, z_2, s_3, \ldots, s_u))$. Since $\bar{r}_1$ is translated to $t_1^{(k)}$ and $\bar{r}_1$ is a subtree of $\bar{r}_2$, an upper bound for the height of the trees that can be translated from $\bar{r}_2$ is

$$n_2(n_1+1)(h(t_1^{(k)})+h(\bar{q}(z_1, z_2, s_3, \ldots, s_u))).$$

However, this is impossible by the choice of the trees $t_1^{(k)}$ and $t_2^{(l)}$.

If $\bar{r}_2\in\operatorname{sub}(\bar{r}_1)$ then $h(\bar{r}_2)<h(\bar{r}_1)$. Thus, a tree translated from $\bar{r}_1$ is at least as high as a tree translated from $\bar{r}_2$. Since $t_2^{(l)}$ is translated from $\bar{r}_2$, $h(\bar{r}_2)>h(t_2^{(l)})/n_2$, and a tree translated from $\bar{r}_1$ is at least $h(\bar{r}_2)/(n_1+1)$ high. It follows that the trees translated from $\bar{r}_1$ are at least $h(t_1'^{(k)})+h(\bar{q}(z_1, z_2, s_3, \ldots, s_u))$ high. This contradicts the choice of $t_1^{(k)}, t_2^{(l)}$.

From our result we immediately obtain.

Theorem 3.9. It is decidable for a bounded R-transducer **A** if **A** preserves regularity.

Theorem 3.10. It is decdiable if an R-transducer is invertible, and the inverse, if exists, can be effectively constructed.

Further results on regularity preserving R-transducers can be found e.g. in [1].

DEPT. OF COMPUTER SCIENCE
A. JÓZSEF UNIVERSITY
ARADI V. TERE 1
SZEGED, HUNGARY
H—6720

References

[1] ÉSIK, Z., Decidability results concerning tree transducers II., Acta Cybernet., 6 (1983), 303—314.
[2] GÉCSEG, F., M. Steinby Algebraic theory of tree automata I., Matematikai lapok 26 (1975), 169—207, in Hungarian.
[3] GÉCSEG, F., M. Steinby, Algebraic theory of tree automata II., Matematikai lapok 27 (1976), 283—336, in Hungarian.
[4] GÉCSEG, F., Tree transformation preserving recognizability, Colloquia Mathematica Societatis János Bolyai 28. Finite Algebra and Multiple-Valued Logic, 251—273.

(Received May 6, 1986)