Acta Cybernetica **12** (1996) 261-277.

On Domain and Range Tree Languages of Superlinear Deterministic Top-down Tree Transformations*

Gábor Dányi†

Abstract

Denote by *sl-DT* the class of superlinear deterministic top-down tree transformations, by *REC* the class of recognizable tree languages, and by *DREC* the class of deterministic recognizable tree languages. In this paper we present the following results. The class dom(*sl-DT*) is exactly the class of tree languages recognized by semi-universal deterministic top-down tree recognizers, which are introduced in this paper. Moreover, for any $L \in DREC$, it is decidable whether $L \in \text{dom}(sl\text{-}DT)$ holds and we also present a decision procedure. Finally, we show that range(*sl-DT*) = *REC*.

1 Introduction

Top-down tree transducers were introduced in [Rou] as formal models of syntax-directed compilers. They are finite devices processing terms over ranked alphabets, which are called trees in this area. A top-down tree transducer induces a binary relation over trees, called a top-down tree transformation. Tree transformations induced by top-down tree transducers serve as abstract models of translations realized by syntax-directed compilers.

Top-down tree transducers and top-down tree transformations were studied in several papers, e.g., in pioneer works [Eng1], [Eng2] and [Bak]. A number of special types have been defined (linear, nondeleting, etc.) and compared to each other, see, for example, [GécSte1] or [FülVág].

A top-down tree transducer translates an input tree by applying so called rules at the nodes, processing the tree from the root to the leaves. Each rule has the form $q(\sigma(x_1, \ldots, x_m)) \to \xi$, where σ is an input symbol of rank m that labels a node of the input tree, q is a state of the tree transducer and ξ is a term consisting of output symbols and terms of the form $p(x_i)$, where $1 \leq i \leq m$, x_i refers to the ith

*Research of the author was supported by the Research Foundation of Hungary under Grant F012852 and by the Hungarian Cultural and Educational Ministry under Grant 223/95.

†Department of Foundations of Computer Science, József Attila University, P.O. Box 652., H-6701 Szeged, Hungary, email: danyi@inf.u-szeged.hu

direct subtree and p is a state of the tree transducer. A top-down tree transducer is said to be deterministic if, for any state q and symbol σ, there is at most one rule of the above form. In this paper we consider only deterministic top-down tree transducers. A top-down tree transducer is called linear if, for every rule of the above form and $1 \leq i \leq m$, each term of the form $p(x_i)$ appears at most once in ξ. As a result of this condition, the translation of any direct subtree t_i of a tree $\sigma(t_1, \ldots, t_m)$ appears at most once in the translation of the whole tree.

Superlinear deterministic top-down tree transducers were defined first time by Z. Fülöp and H. Vogler during a personal communication in 1992. A characterization of this type and comparisons to other ones can be found in [DánFül1] and [DánFül2].

Informally speaking, a superlinear deterministic top-down tree transducer is a linear deterministic top-down tree transducer, which satisfies the following condition. For any symbol σ of rank m and two different states q and q', if both the rules $q(\sigma(x_1, \ldots, x_m)) \to \xi$ and $q'(\sigma(x_1, \ldots, x_m)) \to \xi'$ exist, then, for every $1 \leq i \leq m$, a term of the form $p(x_i)$ appears in at most one of ξ and ξ'. This implies that the translation of a direct subtree t_i of a tree $\sigma(t_1, \ldots, t_m)$ appears in the translation of the whole tree if and only if the transducer starts the processing of $\sigma(t_1, \ldots, t_m)$ in a distinguished state depending on i.

Investigating a certain type of top-down tree transducers, the questions naturally arises, what kind of trees can be processed by this type, and what kind of trees may occur as result of such a processing. For a top-down tree transducer, the sets of possible input and output trees are called the domain and the range of the induced tree transformation, respectively.

Tree sets are also called tree languages. Similarly to the string languages, for tree languages there are also finite state recognizers. By the help of these devices, we can define the classes of recognizable and deterministic recognizable tree languages, see [GécStel]. It turned out that domain and range tree languages of top-down tree transformations can generally be associated with these two classes, see [GécStel] and [FülVág]. However, it is known from [DánFül1] that the class of domain tree languages of superlinear deterministic top-down tree transformations is a proper subclass of the class of deterministic recognizable tree languages.

In this paper we investigate domain and range tree languages of superlinear deterministic top-down tree transformations. We define a new type of tree recognizers, called semi-universal deterministic tree recognizer. We show that the domain tree languages of superlinear deterministic top-down tree transducers are exactly that ones, which are recognizable by semi-universal deterministic tree recognizers. On the basis of this result we develop a decision algorithm, which decides whether an arbitrary deterministic recognizable tree language can be the domain of a superlinear deterministic top-down tree transformation. Moreover, we prove that the range tree languages of superlinear deterministic top-down tree transducers are exactly the recognizable tree languages. The outline of the paper is the following.

In Section 2 we introduce the notions and notations, which are necessary for understanding the paper, and recall some preliminary results.

In Section 3 we give a characterization of the domain of the class of superlinear deterministic top-down tree transformations. Moreover, we show that, for a deterministic recognizable tree language, it is decidable whether it can be the domain of a superlinear deterministic top-down tree transformation.

In Section 4 we show that the range tree languages of superlinear deterministic top-down tree transformations are exactly the recognizable tree languages.

Finally, in Section 5 we summarize our results and raise some related open questions.

2 Preliminaries

In this section we introduce the notions and notations, which are necessary for understanding the paper. Furthermore, we recall some preliminary results, although some of them are referred to in a modified form.

2.1 Sets and relations

We denote by $\mathrm{pow}(A)$ and $|A|$ the power set and the cardinality of a set A, respectively. A finite nonempty set is also called an alphabet.

Given two sets A and B, an arbitrary subset θ of $A \times B$ is a *relation* from A to B. We also write $a\theta b$ meaning that $(a,b) \in \theta$. A relation from A to A is called a *relation over* A. The *identity relation over* A is $\mathrm{Id}(A) = \{(a,a) \mid a \in A\}$.

Let A be a set and $a,b,c \in A$ arbitrary elements. A relation θ over a set A is said to be *reflexive* if $(a,a) \in \theta$ always holds, *transitive* if $a\theta b$ and $b\theta c$ implies $a\theta c$, *symmetric* if $a\theta b$ implies $b\theta a$, and *equivalence* if it is reflexive, transitive, and symmetric. An equivalence relation $\equiv$ over A defines a partitioning of A, where, for any $a,b \in A$, a and b are the same class if and only if $a \equiv b$. We denote by $[a]_{\equiv}$ the class of an $a \in A$ with respect to $\equiv$.

The *transitive closure* θ^+ of a relation θ over A is also a relation over A. For any $a,b \in A$, $a\theta^+ b$ holds if and only if there exist $a_1,\ldots,a_n \in A$ with $n > 1$ such that $a_1 = a$, $a_n = b$, and $a_i\theta a_{i+1}$, for every $1 \le i < n$. The *transitive-reflexive closure* of θ is $\theta^* = \theta^+ \cup \{(a,a) \mid a \in A\}$.

Let θ be a relation from A to B. The sets $\mathrm{dom}(\theta) = \{a \in A \mid \exists b \in B : a\theta b\}$ and $\mathrm{range}(\theta) = \{b \in B \mid \exists a \in A : a\theta b\}$ are called the *domain* and the *range* of θ, respectively. We say that θ is *total*, if $\mathrm{dom}(\theta) = A$, and *partial* otherwise.

We extend the concepts of domain and range for classes of relations. Let $\mathcal{C}$ be a class of relations, then the domain and the range of $\mathcal{C}$ are defined by $\mathrm{dom}(\mathcal{C}) = \{\mathrm{dom}(\theta) \mid \theta \in \mathcal{C}\}$ and $\mathrm{range}(\mathcal{C}) = \{\mathrm{range}(\theta) \mid \theta \in \mathcal{C}\}$, respectively.

Let A and B be sets. A relation $\nu \subseteq A \times B$ is called a *mapping* from A to B, denoted by $\nu : A \to B$, if, for any $a \in A$, there is exactly one $b \in B$ such that $(a,b) \in \nu$ holds, for which we also write $\nu(a) = b$. A mapping ν from A to B is said to be *injective* if, for each $b \in B$, there is at most one $a \in A$ such that $\nu(a) = b$, *surjective* if $\mathrm{range}(\nu) = B$, and *bijective* if it is injective and surjective. A bijective mapping is also called a *bijection*.

2.2 Trees

A *ranked alphabet* Σ is an alphabet, in which every symbol has a unique rank in the set of nonnegative integers. For each $m \geq 0$, the set of symbols in Σ having rank m is denoted by Σ_m. We write $\Sigma = \{\sigma_1^{(m_1)}, \ldots, \sigma_n^{(m_n)}\}$ meaning that $\Sigma = \{\sigma_1 \ldots, \sigma_n\}$ is a ranked alphabet, where the symbol σ_i has the rank m_i, for each $1 \leq i \leq n$.

We specify an enumerable set $X = \{x_1, x_2, \ldots\}$ of symbols, called *variables*, and we put $X_n = \{x_1, \ldots, x_n\}$, for every $n \geq 0$. We assume that X is disjoint with any ranked alphabet Σ.

Let Σ be a ranked alphabet. For a set H, disjoint with Σ, the *set of trees over* Σ *indexed by* H is denoted by $T_\Sigma(H)$ and defined as the smallest set U, which satisfies the following conditions.

(*i*) $H \cup \Sigma_0 \subseteq U$.

(*ii*) $\sigma(t_1, \ldots, t_m) \in U$, whenever $m > 0$, $\sigma \in \Sigma_m$, and $t_1, \ldots, t_m \in U$.

The set $T_\Sigma(\emptyset)$ of *ground trees* over Σ is also written as T_Σ.

The trees can be represented as expressions with parentheses. For example, if $\Sigma = \{\delta^{(2)}, \sigma^{(1)}, \#^{(0)}\}$ then $\delta(\sigma(\#), \#) \in T_\Sigma$ and $\delta(\delta(x_1, \#), \delta(\sigma(x_2), \sigma(x_1))) \in T_\Sigma$. A chain tree, like $\sigma(\ldots\sigma(\#)\ldots)$, where σ occurs i times, is abbreviated by $\sigma^i(\#)$. For instance, $\sigma^3(\#)$ denotes the tree $\sigma(\sigma(\sigma(\#)))$.

We write $T_{\Sigma,n}$ for $T_\Sigma(X_n)$, and distinguish a subset $\hat{T}_{\Sigma,n}$ of $T_{\Sigma,n}$ as follows. A tree $t \in T_{\Sigma,n}$ is in $\hat{T}_{\Sigma,n}$ if and only if each variable in X_n appears in t exactly once and the order of them from left to right is $x_1, \ldots, x_n$. For instance, if $\Sigma = \{\sigma^{(2)}\}$, then $\sigma(x_1, x_2) \in \hat{T}_{\Sigma,2}$, but $\sigma(x_1, x_1), \sigma(x_2, x_1) \notin \hat{T}_{\Sigma,2}$.

Let $t \in T_{\Sigma,n}$, for some $n \geq 0$. We define the *height of* t and the *set of variables occurring in* t, denoted by $\text{height}(t)$ and $\text{var}(t)$, respectively, as follows.

(*i*) If $t = x_i \in X_n$, then $\text{height}(t) = 0$ and $\text{var}(t) = \{x_i\}$.

(*ii*) If $t = \sigma \in \Sigma_0$, then $\text{height}(t) = 0$ and $\text{var}(t) = \emptyset$.

(*iii*) If $t = \sigma(t_1, \ldots, t_m)$, where $m > 0$, $\sigma \in \Sigma_m$, and $t_1, \ldots, t_m \in T_{\Sigma,n}$, then $\text{height}(t) = 1 + \max\{\text{height}(t_i) \mid 1 \leq i \leq m\}$ and $\text{var}(t) = \bigcup_{1 \leq i \leq m} \text{var}(t_i)$.

We introduce the concept of *tree substitution*. Let $n \geq 0$, $t \in T_{\Sigma,n}$ and let $s_1, \ldots, s_n$ be arbitrary trees. We denote by $t[s_1, \ldots, s_n]$ the tree, which is obtained from t by replacing each occurrence of x_i by s_i, for every $1 \leq i \leq n$.

Let r and s be ground trees over a ranked alphabet Σ. We say that r is a *subtree* of s, if there exists a tree $t \in \hat{T}_{\Sigma,1}$ such that $s = t[r]$.

Let Σ and Δ be ranked alphabets. A *tree language* L over Σ is a subset of T_Σ. A *tree transformation from* T_Σ *to* T_Δ is a relation from T_Σ to T_Δ. Since the tree transformations are relations, the concepts of their domain and range should be clear. Note that if τ is a tree transformation from T_Σ to T_Δ, then $\text{dom}(\tau)$ and $\text{range}(\tau)$ are tree languages over Σ and Δ, respectively.

2.3 Top-down tree transducers

A *top-down tree transducer* is a 5-tuple $T = (Q, \Sigma, \Delta, q_0, R)$, where

- Q is an unary ranked alphabet, i.e. $Q = Q_1$, called the set of *states*, which satisfies $Q \cap (\Sigma \cup \Delta) = \emptyset$,
- Σ and Δ are arbitrary ranked alphabets, called *input* and *output* ranked alphabets, respectively,
- $q_0 \in Q$ is the initial state, and
- R is a finite set of rules of the form

$$q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})],$$

  where $m, n \geq 0$, $\sigma \in \Sigma_m$, $1 \leq i_1, \ldots, i_n \leq m$, $q, q_1, \ldots, q_n \in Q$, and $t \in \hat{T}_{\Delta,n}$.

A rule as above will be referred to as a q-rule for σ, or shortly as a (q, σ)-rule. We say that q is defined on σ in R, if there exists a (q, σ)-rule in R.

A top-down tree transducer T is called *deterministic*, if, for any $q \in Q$ and $\sigma \in \Sigma$, there is at most one (q, σ)-rule in R. For brevity, we write "dt" for "deterministic top-down" in the sequel.

Consider the above (q, σ)-rule. The term $t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ is called the *right-hand side* of the rule and it is denoted by $\mathrm{rhs}(q, \sigma)$. Moreover, for each $1 \leq j \leq m$, we define $\mathrm{rst}(q, \sigma, j) = \{q_k \in Q \mid 1 \leq k \leq n, i_k = j\}$, i.e. the set of states applied to x_j in $\mathrm{rhs}(q, \sigma)$.

For a set S of trees, we put $Q(S) = \{q(s) \mid q \in Q,\ s \in S\}$. The rules in R induce a relation, called *derivation*, denoted by $\Rightarrow_T$, over the set $T_\Delta(Q(T_\Sigma))$. For any trees $r, s \in T_\Delta(Q(T_\Sigma))$, $r \Rightarrow_T s$ holds if and only if there is a rule $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ in R such that, for some $t_1, \ldots, t_m \in T_\Sigma$, s is obtained from r by replacing an occurrence of a subtree $q(\sigma(t_1, \ldots, t_m))$ of r by $t[q_1(t_{i_1}), \ldots, q_n(t_{i_n})]$.

The *tree transformation* τ_T *induced by* T is defined as

$$\tau_T = \{(r, s) \in T_\Sigma \times T_\Delta \mid q_0(r) \Rightarrow_T^* s\}.$$

A tree transformation is called a *dt tree transformation*, if it can be induced by a dt tree transducer. The class of dt tree transformations is denoted by DT.

We note that nondeterministic top-down tree transducers are sometimes defined to have more than one initial states. However, that concept is not essentially different from our one. It is an easy exercise to show that, for each top-down tree transducer having more initial state, a top-down tree transducer with one initial state can be constructed, which induces the same tree transformation.

We introduce some special types of dt tree transducers applying certain restrictions to the form of rules. Moreover, we specify a unique abbreviation for the name of each type. Let $T = (Q, \Sigma, \Delta, q_0, R)$ be a dt tree transducer. We say that it is

- *linear (l)*, if, for every rule $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ in R, each of the variables $x_1, \ldots, x_m$ appears at most once in the right-hand side. Note that in this case $m \geq n$.

- *superlinear (sl)*, if it is linear and, for every $m \geq 0$, $\sigma \in \Sigma_m$, and any two different states $q, p \in Q$, $\text{var}(\text{rhs}(q, \sigma)) \cap \text{var}(\text{rhs}(p, \sigma)) = \emptyset$ holds. In other words, T is sl-dt, if it is linear and, for every $m \geq 0$, $\sigma \in \Sigma_m$, and $1 \leq i \leq m$, there is at most one state $q \in Q$ such that x_i occurs in $\text{rhs}(q, \sigma)$.

- *relabeling (rl)*, if each rule in R is of the form

$$q(\sigma(x_1, \ldots, x_m)) \to \delta(q_1(x_1), \ldots, q_m(x_m)),$$

where $m \geq 0, \sigma \in \Sigma_m, \delta \in \Delta_m$. Roughly speaking, processing a tree, T does not change the skeleton, only relabels the nodes.

Observe that an rl-dt tree transducer is necessarily linear, but generally not superlinear.

These attributes can be combined. For example, by an rl-sl-dt transducer we mean a relabeling and superlinear deterministic top-down tree transducer.

Let x be a combination of some of the modifiers in {l,sl,rl}, such as rl-sl, etc. A dt tree transformation is said to be an x-dt transformation if it can be induced by an x-dt transducer. The class of x-dt tree transformations is denoted by x-DT.

2.4 Top-down tree recognizers

A *top-down tree recognizer (ttr)* is a top-down tree transducer $T = (Q, \Sigma, \Sigma, q_0, R)$, of which the rules are of the form

$$q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)),$$

where $m \geq 0$. If T is deterministic, then it is called a *deterministic top-down tree recognizer (dttr)*. Observe that $\tau_T \subseteq \text{Id}(T_\Sigma)$ holds and T is an rl-dt tree transducer.

Let $T = (Q, \Sigma, \Sigma, q_0, R)$ be a dttr. We say that a state $q \in Q$ is *universal*, if, for all $t \in T_\Sigma$, $q(t) \Rightarrow_T^* t$ holds, i.e. $\{t \in T_\Sigma \mid q(t) \Rightarrow_T^* t\} = T_\Sigma$. Observe that, for any rule $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$, if q is universal, then $q_1, \ldots, q_m$ are necessarily universal, too.

We say that T *recognizes* the tree $t \in T_\Sigma$, if $q_0(t) \Rightarrow_T^* t$. The *tree language recognized by* T is $L(T) = \{t \in T_\Sigma \mid q_0(t) \Rightarrow_T^* t\}$. Observe that $L(T) = \text{dom}(\tau_T)$. A tree language is *recognizable* (resp. *deterministic recognizable*), if it is recognized by a ttr (resp. dttr). We denote by REC (resp. $DREC$) the class of recognizable (resp. deterministic recognizable) tree languages.

Note that the original concept of recognizability concerning tree languages is defined by descending (or bottom-up) tree automata, see in [GécSte1]. However, consulting Chapter II in [GécSte1], one can easily see, that top-down tree recognizers are equivalent to regular tree grammars in normal form and hence to descending tree automata.

Clearly, $DREC \subseteq REC$ holds. Moreover, it is a well-known result (see, e.g., [GécSte1]) that the inclusion is proper, i.e. $DREC \subset REC$.

2.5 Minimal deterministic top-down tree recognizers

Deterministic top-down tree recognizers also have automaton type equivalent, namely *deterministic ascending (or top-down) tree automata (dtta)*. A short reflection will show that there are mainly notational differences between these types of devices, hence we can apply the notions and results in [GécSte2] to dttr's without difficulties.

An n-ary dtta is a 5-tuple $A = (Q, \Sigma, Y_n, q_0, F)$, where $n > 0$,

- Q is the finite nonempty set of states,
- $q_0 \in Q$ is the initial state,
- $F = (Q_1, \ldots, Q_n) \in (\mathrm{pow}(Q))^n$ is the final state vector,
- $Y_n = \{y_1, \ldots, y_n\}$ is the set of automaton variables, and
- Σ is a ranked alphabet, where $\Sigma \cap Y_n = \emptyset$, $\Sigma_0 = \emptyset$, and every $\sigma \in \Sigma_m$ with $m > 0$ is realized as a mapping $\sigma^A : Q \to Q^m$.

We now specify how a dtta A recognizes trees. Define the mapping $\alpha_A : T_\Sigma(Y_n) \to \mathrm{pow}(Q)$ as follows.

(i) $\alpha_A(y_i) = Q_i$, for $1 \leq i \leq n$, and

(ii) $\alpha_A(t) = \{q \in Q \mid \sigma^A(q) \in \alpha_A(t_1) \times \ldots \times \alpha_A(t_m)\}$, if $t = \sigma(t_1, \ldots, t_m)$ with $m > 0$, $\sigma \in \Sigma_m$, and $t_1, \ldots, t_m \in T_\Sigma(Y_n)$.

The tree language recognized by A is $L(A) = \{t \in T_\Sigma(Y_n) \mid q_0 \in \alpha_A(t)\}$.

We show that, for any dttr, an equivalent dtta can be constructed.

Construction 2.1 *Consider an arbitrary dttr $T = (Q, \Sigma, \Sigma, q_0, R)$ and suppose that $\Sigma_0 = \{\delta_1, \ldots, \delta_n\}$ with $n > 0$. Let $p \notin Q$ be a new state. Define the dtta $A = (Q \cup \{p\}, \Sigma - \Sigma_0, \Sigma_0, q_0, F)$, where*

- *$F = (Q_1, \ldots, Q_n)$ with $Q_i = \{q \in Q \mid q(\delta_i) \to \delta_i \in R\}$, for $1 \leq i \leq n$ and,*
- *for all $m > 0$, $\sigma \in \Sigma_m$, and $q \in Q \cup \{p\}$, if $q \in Q$ and $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ is in R, then let $\sigma^A(q) = (q_1, \ldots, q_m)$, otherwise let $\sigma^A(q) = (p, \ldots, p)$.*

It is straightforward to prove $L(A) = L(T)$. Note that the case $n = 0$ is trivial, because then $L(T) = \emptyset$.

Conversely, for any dtta, an equivalent dttr can be constructed.

Construction 2.2 *Let $A = (Q, \Sigma, Y_n, q_0, F)$ be an arbitrary dtta. Assign the rank 0 to each element of Y_n and let $\Delta = \Sigma \cup Y_n$. Define the dttr $T = (Q, \Delta, \Delta, q_0, R)$, where R is constructed as follows:*

(i) for all $1 \leq i \leq n$, $q(y_i) \to y_i \in R$ if and only if $q \in Q_i$ and,

(ii) for all $m > 0$, $\sigma \in \Sigma_m$ and $q, q_1, \ldots, q_m \in Q$, the rule $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ is in R if and only if $\sigma^A(q) = (q_1, \ldots, q_m)$.

It is easy to show $L(T) = L(A)$.

We now recall some definitions and results from [GécSte2] using the dttr notation. Note that two dttr's are called equivalent, if they recognize the same tree language.

Let $T = (Q, \Sigma, \Sigma, q_0, R)$ be a dttr. A state $q \in Q$ of T is called 0-*state* if the set $\{t \in T_\Sigma \mid q(t) \Rightarrow_T^* t\}$ is empty. A dttr $T = (Q, \Sigma, \Sigma, q_0, R)$ is said to be *normalized*, if either it has no 0-state, or the only 0-state is q_0 and in this case $Q = \{q_0\}$ and $R = \emptyset$ hold.

We note that this concept of normalization differs from the original one in [GécSte2] on page 40. Namely, if a dttr is normalized by the above definition, then the dtta given by the Construction 2.1 is normalized in the sense of [GécSte2]. However, the converse is not true, that is if a dtta is normalized in the sense of [GécSte2], then the dttr given by the Construction 2.2 is generally not normalized by the above definition. The difference follows from the fact that in the case of dtta's every $\sigma \in \Sigma_m$ with $m > 0$ is realized as a mapping $\sigma^A : Q \to Q^m$, hence σ^A should be defined for each $q \in Q$. That is why the 0-states cannot be discarded completely from the state set of a normalized dtta. On the other hand, the 0-states (except q_0) and the corresponding rules can be deleted without difficulties in the case of dttr's, as it is shown by the following proposition.

Propositon 2.3 *For any dttr $T = (Q, \Sigma, \Sigma, q_0, R)$, an equivalent normalized dttr $T_{nor} = (Q', \Sigma, \Sigma, q_0, R')$ can be constructed effectively such that $Q' \subseteq Q$ and $R' \subseteq R$.*

Proof. The set of non 0-states can be computed as follows. Define a sequence $Q^{(0)} \subseteq Q^{(1)} \subseteq \ldots$ of subsets of Q, where

(i) $Q^{(0)} = \{q \in Q \mid \exists \delta \in \Sigma_0 : q(\delta) \to \delta \in R\}$ and,

(ii) for $i \geq 0$, $Q^{(i+1)} = Q^{(i)} \cup \{q \in Q \mid \exists m \geq 1, \sigma \in \Sigma_m : q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$ and $q_1, \ldots, q_m \in Q^{(i)}\}$.

Obviously, there is a $k \geq 0$ such that $Q^{(k)} = Q^{(k+j)}$, for every $j \geq 1$.

If $q_0 \notin Q^{(k)}$, then let $Q' = \{q_0\}$ and $R' = \emptyset$. Clearly, in this case $L(T) = L(T_{nor}) = \emptyset$ holds.

Finally, if $q_0 \in Q^{(k)}$, then let $Q' = Q^{(k)}$ and $R' = \{q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R \mid q, q_1, \ldots, q_m \in Q^{(k)}\}$. Observe that, for any $t \in L(T)$, during the derivation $q_0(t) \Rightarrow_T^* t$ only such rules are applied, which do not contain a 0-state, hence $q_0(t) \Rightarrow_{T_{nor}}^* t$, too. Therefore $L(T_{nor}) = L(T)$. □

We define the binary relation $\mapsto_T$ over Q as follows. Let $q, p \in Q$, then $q \mapsto_T p$ if and only if there exists a $\sigma \in \Sigma_m$ with $m > 0$ such that p appears in $\text{rhs}(q, \sigma)$. We say that p is *accessible* from q if $q \mapsto_T^* p$ holds. The dttr T is called *connected* if every state in Q is accessible from q_0.

Note that the above concept of accessibility is derived from the concept of reachability of states of dtta's defined in [GécSte2] on pages 41-42.

Propositon 2.4 *For any dttr $T = (Q, \Sigma, \Sigma, q_0, R)$, an equivalent connected dttr $T_{con} = (Q', \Sigma, \Delta, q_0, R')$ can be constructed effectively such that $Q' \subseteq Q$ and $R' \subseteq R$. Moreover, if T is normalized, then T_{con} is also normalized.*

Proof. The set of accessible states can be determined in the following way. Define a sequence $Q^{(0)} \subseteq Q^{(1)} \subseteq \dots$ of subsets of Q, where

(i) $Q^{(0)} = \{q_0\}$ and,

(ii) for $i \geq 0$, $Q^{(i+1)} = Q^{(i)} \cup \{q \in Q \mid \exists m \geq 1, \sigma \in \Sigma_m, p \in Q^{(i)} : q \text{ occurs in } \text{rhs}(p, \sigma)\}$.

Clearly, there is a $k \geq 0$ such that $Q^{(k)} = Q^{(k+j)}$, for every $j \geq 1$.

Let $Q' = Q^{(k)}$ and $R' = \{q(\sigma(x_1, \dots, x_m)) \to \sigma(q_1(x_1), \dots, q_m(x_m)) \in R \mid q, q_1, \dots, q_m \in Q^{(k)}\}$. It is easy to show that $L(T_{con}) = L(T)$.

Note that the construction of Q' is derived from the procedure computing H_k's in [GécSte2] on page 43.

It is not hard to show that if T is normalized, then T_{con} is also normalized. □

By Proposition 2.4, if T is a normalized dttr, then T_{con} is also normalized. However, the converse is not true, i.e. if T is a connected dttr, then T_{nor} is not necessarily connected. To see this, consider the following example. Let $T = (\{q_0, q_1, q_2\}, \Sigma, \Sigma, q_0, R)$, where $\Sigma = \{\sigma^{(2)}, \#^{(0)}\}$ and $R = \{q_0(\sigma(x_1, x_2)) \to \sigma(q_1(x_1), q_2(x_2)),\ q_0(\#) \to \#,\ q_1(\#) \to \#\}$. Clearly, T is a connected dttr, but $T_{nor} = (\{q_0, q_1\}, \Sigma, \Sigma, q_0, \{q_0(\#) \to \#,\ q_1(\#) \to \#\})$ is not connected, because q_1 is not an accessible state in T_{nor}.

If we refer to the construction of the normalized and connected equivalent dttr $T_{nor,con}$ of a dttr T in the sequel, then we always mean that T_{nor} should be determined first from T as defined in the proof of Proposition 2.3 and $T_{nor,con}$ should be computed from T_{nor} as specified in the proof of Proposition 2.4.

A dttr T is said to be *minimal* if, for every dttr T' such that T' is equivalent to T, $|Q| \leq |Q'|$ holds, where Q and Q' are the sets of states of T and T', respectively (cf. minimal dtta on page 38 in [GécSte2]).

Let $T = (Q, \Sigma, \Sigma, q_0, R)$ and $T' = (Q', \Sigma, \Sigma, q_0', R')$ be dttr's. We say that T and T' are *isomorphic* if there exists a bijection $\nu : Q \to Q'$ such that $\nu(q_0) = q_0'$ holds and, for any $m \geq 0$, $\sigma \in \Sigma_m$, and states $q, q_1, \dots, q_m \in Q$, the rule $q(\sigma(x_1, \dots, x_m)) \to \sigma(q_1(x_1), \dots, q_m(x_m))$ is in R if and only if the rule $\nu(q)(\sigma(x_1, \dots, x_m)) \to \sigma(\nu(q_1)(x_1), \dots, \nu(q_m)(x_m))$ is in R'. In this case ν is also called a *dttr isomorphism* (cf. dtta isomorphism on page 39 in [GécSte2]). Note that if T and T' are isomorphic, then clearly $|Q| = |Q'|$ and they are equivalent.

We say that a minimal dttr T is *unique up to the isomorphism* if, for each minimal dttr T' equivalent to T, T' and T are isomorphic.

The following result is derived from Theorem 8 in [GécSte2].

Propositon 2.5 *For any dttr T, there exists an equivalent minimal dttr T_{min}. Moreover, it is unique up to isomorphism and can effectively be constructed.*

We also present the construction of the minimal dttr. We note that the following construction is derived from the construction of reduced dtta presented in [GécSte2] on page 43.

Construction 2.6 *Let $T = (Q, \Sigma, \Sigma, q_0, R)$ be a dttr. By Propositions 2.3 and 2.4, we can assume without loss of generality that T is normalized and connected (if it is not, then consider $T_{nor,con}$ instead of T). We define a sequence $\equiv_0 \supseteq \equiv_1 \supseteq \ldots$ of equivalence relations over Q, where*

(i) $q \equiv_0 p$ if and only if, for every $\sigma \in \Sigma_0$, $q(\sigma) \to \sigma \in R$ holds if and only if $p(\sigma) \to \sigma \in R$ and,

(ii) for $i \geq 0$, $q \equiv_{i+1} p$ if and only if $q \equiv_i p$ and, for every $\sigma \in \Sigma_m$ with $m > 0$, either both q and p are not defined on σ, or both $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ and $p(\sigma(x_1, \ldots, x_m)) \to \sigma(p_1(x_1), \ldots, p_m(x_m))$ are in R and then $q_j \equiv_i p_j$ holds, for each $1 \leq j \leq m$.

Clearly, there is a $k \geq 0$ such that $\equiv_k$ and $\equiv_{k+j}$ are the same, for every $j \geq 1$. The minimal dttr equivalent to T is defined as $T_{min} = (Q', \Sigma, \Sigma, [q_0]_{\equiv_k}, R')$, where

- *$Q' = \{[q]_{\equiv_k} \mid q \in Q\}$ and*
- *$[q]_{\equiv_k}(\sigma(x_1, \ldots, x_m)) \to \sigma([q_1]_{\equiv_k}(x_1), \ldots, [q_m]_{\equiv_k}(x_m)) \in R'$ if and only if $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$, for any $m \geq 0$, $\sigma \in \Sigma_m$, and $q, q_1, \ldots, q_m \in Q$.*

The proof of Proposition 2.5 is rather long, technical and needs new concepts to introduce (e.g. dttr congruence, quotient dttr, etc.). However, it is an easy exercise to present it if one follows the proof of Theorem 8 in [GécSte2] step by step. Hence we omit the proof here.

However, we note that, proving Proposition 2.5 on the basis of [GécSte2], it should be considered that, in contrast with a dtta, a state of a dttr is not necessarily defined for all input symbols (cf. definitions of $\equiv_{i+1}$ in (ii) of Construction 2.6 and ρ_{k+1} in (ii) in [GécSte2] on page 43).

2.6 Recognizability of domain and range tree languages of deterministic top-down tree transformations

Recall that a dttr is also an rl-dt tree transducer, hence $DREC \subseteq \text{dom}(rl\text{-}DT)$ holds. On the other hand, the following statement shows that $\text{dom}(DT) \subseteq DREC$. The original statement can be found as Lemma 5 in [FülVág] (this result also appears in [Eng2]), although it is slightly modified here for our purposes.

Propositon 2.7 *For any dt tree transducer $T = (Q, \Sigma, \Delta, q_0, R)$, there exists a connected dttr $T' = (Q', \Sigma, \Sigma, \{q_0\}, R')$ such that $L(T') = dom(\tau_T)$.*

Proof. We define a sequence $Q^{(0)} \subseteq Q^{(1)} \subseteq \ldots$ of subsets of pow(Q) and a sequence $R^{(0)} \subseteq R^{(1)} \subseteq \ldots$ of finite sets of rules of the form $P(\sigma(x_1, \ldots, x_m)) \to \sigma(P_1(x_1), \ldots, P_m(x_m))$, where $m \geq 0$, $\sigma \in \Sigma_m$, $P, P_1, \ldots, P_m \subseteq Q$.

(i) Put $P_0 = \{q_0\}$. Let $Q^{(0)} = \{P_0\} \cup \{\mathrm{rst}(q_0, \sigma, j) \mid m \geq 1, \sigma \in \Sigma_m, 1 \leq j \leq m,\ q_0 \text{ is defined on } \sigma \text{ in } R\}$. Moreover, let $R^{(0)} = \{P_0(\sigma(x_1, \ldots, x_m)) \to \sigma(\mathrm{rst}(q_0, \sigma, 1)(x_1), \ldots, \mathrm{rst}(q_0, \sigma, m)(x_m)) \mid m \geq 0, \sigma \in \Sigma_m,\ q_0 \text{ is defined on } \sigma \text{ in } R\}$.

(ii) Let $k \geq 0$. For any $P \in Q^{(k)}$ and $\sigma \in \Sigma$, we say that P is defined on σ if, for each $q \in P$, q is defined on σ in R and in this case if $\sigma \in \Sigma_m$ with $m \geq 1$, we put $S_{P,\sigma,j} = \bigcup_{q \in P} \mathrm{rst}(q, \sigma, j)$, for every $1 \leq j \leq m$. Specially, $P = \emptyset$ is defined for all $\sigma \in \Sigma$ and if $\sigma \in \Sigma_m$ with $m \geq 1$, then $S_{\emptyset,\sigma,j} = \emptyset$, for each $1 \leq j \leq m$.
Now let $Q^{(k+1)} = Q^{(k)} \cup \{S_{P,\sigma,j} \mid m \geq 1, \sigma \in \Sigma_m, 1 \leq j \leq m, P \in Q^{(k)},\ P \text{ is defined on } \sigma\}$. Moreover, let $R^{(k+1)} = R^{(k)} \cup \{P(\sigma(x_1, \ldots, x_m)) \to \sigma(S_{P,\sigma,1}(x_1), \ldots, S_{P,\sigma,m}(x_m)) \mid m \geq 0, \sigma \in \Sigma_m, P \in Q^{(k)},\ P \text{ is defined on } \sigma\}$. Specially, if $\emptyset \in Q^{(k)}$, then $\emptyset(\sigma(x_1, \ldots, x_m)) \to \sigma(\emptyset(x_1), \ldots, \emptyset(x_m)) \in R^{(k+1)}$, for every $m \geq 0$ and $\sigma \in \Sigma_m$.

Clearly, there exists a $k \geq 0$ such that $Q^{(k+1)} = Q^{(k)}$, and then $Q^{(k+1)} = Q^{(k+2)} = \ldots$ and $R^{(k+1)} = R^{(k+2)} = \ldots$ hold. Let $Q' = Q^{(k+1)}$ and $R' = R^{(k+1)}$. It is an exercise to show that T' is exactly the connected version of the dttr defined in the proof of Lemma 5 in [FülVág]. □

Observe that if T is linear, then Q' in the proof of Proposition 2.7 consists of sets containing at most one element. Moreover, if $\emptyset \in Q'$, then it is a universal state of T'.

We have dom$(rl\text{-}DT)$ = dom$(l\text{-}DT)$ = dom(DT) = $DREC$. On the other hand, by (2) of Theorem 4.3 in [DánFül1], dom$(sl\text{-}DT) \subset DREC$ holds.

As for the ranges of various types of dt tree transformation classes, we recall the following results. It is well-known and easy to prove that range$(DT) \not\subseteq REC$. On the other hand, by Corollary 6.6 of Chapter IV in [GécSte1], range$(l\text{-}DT) \subseteq REC$ holds. We note that even the equality can be proved, although this result has not been published yet, to our best knowledge. The equality is shown in Section 4.

3 On domain of sl-DT

In this section we give a characterization of the class dom$(sl\text{-}DT)$. Moreover, we show that, for any $L \in DREC$, it is decidable whether $L \in$ dom$(sl\text{-}DT)$ holds and we present a decision procedure.

Let $T = (Q, \Sigma, \Sigma, q_0, R)$ be a dttr. We say that T is a *semi-universal deterministic top-down tree recognizer (su-dttr)*, if the following condition holds. For any $m \geq 1$, $\sigma \in \Sigma_m$, and two different states $q, p \in Q$, if $q(\sigma(x_1, \ldots, x_m)) \to$

$\sigma(q_1(x_1),\ldots,q_m(x_m))$ and $p(\sigma(x_1,\ldots,x_m)) \to \sigma(p_1(x_1),\ldots,p_m(x_m))$ are in R, then, for each $1 \le i \le m$, at least one of q_i and p_i is universal. We denote by *su-DREC* the class of tree languages recognized by su-dttr's.

First we show that dom(*sl-DT*) and *su-DREC* are equal classes.

Lemma 3.1 *For any sl-dt tree transducer $T = (Q,\Sigma,\Delta,q_0,R)$, dom($\tau_T$) is recognized by an su-dttr.*

Proof. Let the dttr $T' = (Q',\Sigma,\Sigma,\{q_0\},R')$ be constructed from T as defined in the proof of Proposition 2.7, then $L(T') = \text{dom}(\tau_T)$. We show that T' is su-dttr.

Since T is linear, the sets in Q' contain at most one element. Observe that, for any $m \ge 1$, $\sigma \in \Sigma_m$, and $q \in Q$, if $\{q\}(\sigma(x_1,\ldots,x_m)) \to \sigma(P_1(x_1),\ldots,P_m(x_m))$ is in R', then $q(\sigma(x_1,\ldots,x_m)) \to t[q_1(x_{i_1}),\ldots,q_n(x_{i_n})] \in R$, for some $0 \le n \le m$ and $t \in \hat{T}_{\Delta,n}$. Moreover, for each $1 \le j \le m$, if $j = i_k$, for some $1 \le k \le n$, then $P_j = \{q_k\}$, and $P_j = \emptyset$ otherwise. Note that, since T is linear, the i_ks are different.

Now suppose that, for some $m \ge 1$, $\sigma \in \Sigma_m$ and two different states $q,p \in Q$, $\{q\}(\sigma(x_1,\ldots,x_m)) \to \sigma(P_{q,1}(x_1),\ldots,P_{q,m}(x_m))$ and $\{p\}(\sigma(x_1,\ldots,x_m)) \to \sigma(P_{p,1}(x_1),\ldots,P_{p,m}(x_m))$ are in R'. Then, by the above observations, there exist rules $q(\sigma(x_1,\ldots,x_m)) \to t[q_1(x_{i_1}),\ldots,q_n(x_{i_n})]$ and $p(\sigma(x_1,\ldots,x_m)) \to s[p_1(x_{i'_1}),\ldots,p_{n'}(x_{i'_{n'}})]$ in R, for some $0 \le n,n' \le m$, $t \in \hat{T}_{\Delta,n}$, and $s \in \hat{T}_{\Delta,n'}$, where, for each $1 \le j \le m$, if $j = i_k$, for some $1 \le k \le n$, then $P_{q,j} = \{q_k\}$, else $P_{q,j} = \emptyset$. Moreover, if $j = i'_k$, for some $1 \le k \le n'$, then $P_{p,j} = \{p_k\}$, and $P_{p,j} = \emptyset$ otherwise.

Since T is superlinear, $\{i_1,\ldots,i_n\} \cap \{i'_1,\ldots,i'_{n'}\} = \emptyset$, hence we have that, for each $1 \le j \le m$, at least one of $P_{q,j}$ and $P_{p,j}$ is $\emptyset$. We saw that if $\emptyset \in Q'$, then it is necessarily a universal state, hence T' is su-dttr. □

Lemma 3.2 *For any $L \in$ su-DREC, there exists an sl-dt tree transducer T' such that dom($\tau_{T'}$) $= L$.*

Proof. If $L \in$ *su-DREC*, then it is recognized by an su-dttr $T = (Q,\Sigma,\Sigma,q_0,R)$. For every $m \ge 0$, $\sigma \in \Sigma_m$, $q \in Q$, and $q(\sigma(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m)) \in R$, consider the set $\{i_1,\ldots,i_n\} \subseteq \{1,\ldots,m\}$ of indices, where $i_1 < \ldots < i_n$ and, for any $1 \le j \le m$, $j \in \{i_1,\ldots,i_n\}$ holds if and only if q_j is not a universal state. Then let σ_q be a new symbol having the rank n and define the rule $r_{q,\sigma} : q(\sigma(x_1,\ldots,x_m)) \to \sigma_q(q_{i_1}(x_{i_1}),\ldots,q_{i_n}(x_{i_n}))$.

Let $T' = (Q,\Sigma,\Delta,q_0,R')$ be a dt transducer, where

- $\Delta = \{\sigma_q \mid q$ is defined on σ in $R\}$ and
- $R' = \{r_{q,\sigma} \mid q$ is defined on σ in $R\}$.

We show that T' is superlinear. Obviously, it is linear. Let $q,p \in Q$ be two different states. Suppose that $q(\sigma(x_1,\ldots,x_m)) \to \sigma_q(q_1(x_{i_1}),\ldots,q_k(x_{i_k}))$ and $p(\sigma(x_1,\ldots,x_m)) \to \sigma_p(p_1(x_{j_1}),\ldots,p_l(x_{j_l}))$ are in R, for some $m \ge 0$, $0 \le k,l \le m$ and $\sigma \in \Sigma_m$. Then, by the construction of T', $\{q_{i_1},\ldots,q_{i_k}\}$ and $\{q_{j_1},\ldots,q_{j_l}\}$ are the sets of non-universal states of rhs(q,σ) and rhs(p,σ) in T, respectively. Since T is su-dttr, $\{i_1,\ldots,i_k\} \cap \{j_1,\ldots,j_l\} = \emptyset$ holds. Therefore T' is superlinear.

Finally, we show that, for any tree $t \in T_\Sigma$ and state $q \in Q$, $q(t) \Rightarrow_T^* t$ holds if and only if $q(t) \Rightarrow_{T'}^* t'$, for some $t' \in T_\Delta$. This implies dom$(\tau_T) = L$ immediately. We prove the statement by induction on height(t).

Basis. Suppose that height$(t) = 0$, then $t = \delta$, for some $\delta \in \Sigma_0$. By the definition of T', $q(\delta) \to \delta_q \in R'$ if and only if $q(\delta) \to \delta \in R$, hence the statement holds by $t' = \delta_q$.

Induction step. Suppose that height$(t) = n+1$ with $n \geq 0$, then $t = \sigma(t_1,\ldots,t_m)$, for some $m \geq 1$, $\sigma \in \Sigma_m$, and $t_1,\ldots,t_m \in T_\Sigma$, where height$(t_i) \leq n$, for each $1 \leq i \leq m$. Recall that $q(\sigma(x_1,\ldots,x_m)) \to \sigma_q(q_{i_1}(x_{i_1}),\ldots,q_{i_n}(x_{i_n})) \in R'$ if and only if $q(\sigma(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m)) \in R$, where $q_{i_1},\ldots,q_{i_n}$ are exactly the non-universal states of rhs(q,σ) in T. Furthermore, by the induction hypothesis, for each $j \in \{i_1,\ldots,i_n\}$, $q_j(t_j) \Rightarrow_T^* t_j$ holds if and only if $q_j(t_j) \Rightarrow_{T'}^* t'_j$, for some $t'_j \in T_\Delta$. Hence $q(t) \Rightarrow_T \sigma(q_1(t_1),\ldots,q_m(t_m)) \Rightarrow_T^* t$ if and only if $q(t) \Rightarrow_{T'} \sigma_q(q_{i_1}(t_{i_1}),\ldots,q_{i_n}(t_{i_n})) \Rightarrow_{T'}^* t'$, where $t' = \sigma_q(t'_{i_1},\ldots,t'_{i_n})$. □

Summarizing the results of the above two lemmas, we have that the domain tree languages of sl-dt tree transformations are exactly those tree languages, which are recognized by su-dttr's.

Theorem 3.3 *dom(sl-DT) = su-DREC*

In the rest of the section we show that, for any $L \in DREC$ given by a dttr recognizing L, it is decidable whether $L \in$ dom(*sl-DT*) holds. Moreover, we present a decision procedure.

Recall that, for a dttr T, T_{nor} and T_{con} denote the normalized and connected equivalents of T, according to Propositions 2.3 and 2.4, respectively. Moreover, if T is normalized and connected, then T_{min} is the minimal equivalent of T, according to Proposition 2.5.

Lemma 3.4 *Let $T = (Q,\Sigma,\Sigma,q_0,R)$ be an su-dttr, then T_{nor} and T_{con} are su-dttr's, too. Moreover, if T is normalized and connected, then T_{min} is also an su-dttr.*

Proof. By Proposition 2.3, $T_{nor} = (Q',\Sigma,\Sigma,q_0,R')$, where $Q' \subseteq Q$ and $R' \subseteq R$ hold. Hence it should be clear that if T is su-dttr, then T_{nor} is also su-dttr.

Similarly, by Proposition 2.4, $T_{con} = (Q'',\Sigma,\Sigma,q_0,R'')$, where $Q'' \subseteq Q$ and $R'' \subseteq R$. Therefore if T is su-dttr, then T_{con} is necessarily su-dttr, too.

Now suppose that T is a normalized and connected su-dttr. Denote by $\equiv$ the equivalence relation, by which T_{min} is constructed from T (see Construction 2.6). Recall that $T_{min} = (Q''',\Sigma,\Sigma,[q_0]_\equiv,R''')$, where $R''' = \{[q]_\equiv(\sigma(x_1,\ldots,x_m)) \to \sigma([q_1]_\equiv(x_1),\ldots,[q_m]_\equiv(x_m)) \mid q(\sigma(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m)) \in R\}$ and $Q''' = \{[q]_\equiv \mid q \in Q\}$.

It can be easily shown that, if T has universal states, then they constitute exactly one class of Q with respect to $\equiv$. Moreover, if this class exists, then it is the only universal state in T_{min}. By the construction of $\equiv$, the proofs of these statements are straightforward.

Suppose that the states $q, p \in Q$ are in different classes with respect to $\equiv$, that is $[q]_\equiv \neq [p]_\equiv$. If, for some $\sigma \in \Sigma_m$ with $m \geq 1$, both $[q]_\equiv$ and $[p]_\equiv$ are defined on σ in R''', then the $([q]_\equiv, \sigma)$-rule and the $([p]_\equiv, \sigma)$-rule of R''' can be written of the form $[q]_\equiv(\sigma(x_1, \ldots, x_m)) \to \sigma([q_1]_\equiv(x_1), \ldots, [q_m]_\equiv(x_m))$ and $[p]_\equiv(\sigma(x_1, \ldots, x_m)) \to \sigma([p_1]_\equiv(x_1), \ldots, [p_m]_\equiv(x_m))$, respectively, where $q_1, \ldots, q_m$, $p_1, \ldots, p_m \in Q$ and the rules $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m))$ and $p(\sigma(x_1, \ldots, x_m)) \to \sigma(p_1(x_1), \ldots, p_m(x_m))$ are in R. Since T is su-dttr, then, for any $1 \leq i \leq m$, at least one of q_i and p_i is universal in T. Thus, by the observations of the previous paragraph, at least one of $[q_i]_\equiv$ and $[p_i]_\equiv$ is universal in T_{min}. Therefore T_{min} is su-dttr, too. □

We recall from Proposition 2.5 that, for any $L \in DREC$, the minimal dttr recognizing L is unique up to isomorphism. Denote this dttr by T_L. The following theorem establishes our decidability result.

Lemma 3.5 *For any tree language $L \in DREC$, $L \in$ su-$DREC$ if and only if T_L is su-dttr.*

Proof. If T_L is su-dttr, then $L \in$ *su-DREC* by definition. Conversely, suppose $L \in$ *su-DREC*, then there exists an su-dttr T such that $L(T) = L$. By Propositions 2.3, 2.4 and 2.5, T_L can be computed from T and, by Lemma 3.4, it is su-dttr, too. □

Theorem 3.6 *For any tree language $L \in DREC$ given by a dttr T recognizing L, it is decidable whether $L \in dom(sl\text{-}DT)$ holds.*

Proof. By Propositions 2.3, 2.4 and 2.5, T_L can be constructed effectively from T. Moreover, it is obviously decidable whether T_L is su-dttr. Hence, by Lemma 3.5 and Theorem 3.3, the statement of the theorem holds. □

Finally, we present an algorithm, which, for any tree language $L \in DREC$ given by a dttr recognizing L, decides whether $L \in$ dom(*sl-DT*) holds. The method is based on the proof of the Theorem 3.6.

Let L be an arbitrary deterministic recognizable tree language and let $T^{(1)}$ be a dttr, which recognizes L. The algorithm gives the answer *YES* if L can be the domain of a superlinear deterministic top-down tree transformation, otherwise it answers *NO*.

1. Compute $T^{(1)}_{nor}$ as defined in the proof of the Proposition 2.3. Denote $T^{(1)}_{nor}$ by $T^{(2)}$.
2. Compute $T^{(2)}_{con}$ as defined in the proof of the Proposition 2.4. Denote $T^{(2)}_{con}$ by $T^{(3)}$.
3. Construct $T^{(3)}_{min}$ as determined in Construction 2.6. Denote $T^{(3)}_{min}$ by T_L.

4. Decide whether T_L is semi-universal. (It is trivially decidable, e.g., check all rule pairs, which concern the same input symbol.) If it is, then the answer is *YES*, else the answer is *NO*.

4 On range of *sl-DT*

In this section we prove range(*sl-DT*) $= REC$. Furthermore, as a by-product, we get range(*l-DT*) $= REC$, too.

Assume that $L \in REC$, then there exists a ttr $T = (Q, \Sigma, \Sigma, q_0, R)$ satisfying $L(T) = L$. We define the ranked alphabet Δ such that, for each $m \geq 0$, we put

$$\Delta_m = \{\sigma_{q,q_1,\ldots,q_m} \mid \sigma \in \Sigma_m,\ q(\sigma(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m)) \in R\}.$$

Let $T' = (Q, \Delta, \Sigma, q_0, R')$ be a dt tree transducer, where

$$R' = \{q(\sigma_{q,q_1,\ldots,q_m}(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m)) \mid \sigma_{q,q_1,\ldots,q_m} \in \Delta\}.$$

Observe that T' is an rl-sl-dt tree transducer.

Lemma 4.1 *For any tree $t \in T_\Sigma$ and state $q \in Q$, $q(t) \Rightarrow_T^* t$ if and only if $q(t') \Rightarrow_{T'}^* t$ holds, for some $t' \in T_\Delta$.*

Proof. First assume $q(t) \Rightarrow_T^* t$. We show the existence of the above t' by induction on height(t).

Basis. Suppose that height$(t) = 0$, then $t = \delta$, for some $\delta \in \Sigma_0$, and then $q(t) \Rightarrow_T^* t$ implies $q(\delta) \to \delta \in R$. By the construction of T', $\delta_q \in \Delta_0$ and $q(\delta_q) \to \delta \in R'$, hence $q(t') \Rightarrow_{T'}^* t$ holds, for $t' = \delta_q$.

Induction step. Suppose that height$(t) = n + 1$ with $n \geq 0$, then $t = \sigma(t_1,\ldots,t_m)$, for some $m \geq 1$, $\sigma \in \Sigma_m$, and $t_1,\ldots,t_m \in T_\Sigma$, where height$(t_i) \leq n$, for all $1 \leq i \leq m$. Since $q(t) \Rightarrow_T^* t$, there should be a rule $q(\sigma(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m))$ in R, where $q_i(t_i) \Rightarrow_T^* t_i$ holds, for each $1 \leq i \leq m$. By the construction of T', $\sigma_{q,q_1,\ldots,q_m} \in \Delta$ and the rule $q(\sigma_{q,q_1,\ldots,q_m}(x_1,\ldots,x_m)) \to \sigma(q_1(x_1),\ldots,q_m(x_m))$ is in R. Moreover, by the induction hypothesis, there exist trees $t'_1,\ldots,t'_m \in T_\Delta$ such that $q_i(t'_i) \Rightarrow_{T'}^* t_i$, for all $1 \leq i \leq m$. Let $t' = \sigma_{q,q_1,\ldots,q_m}(t'_1,\ldots,t'_m)$, then we have $q(t') \Rightarrow_{T'} \sigma(q_1(t'_1),\ldots,q_m(t'_m)) \Rightarrow_{T'}^* \sigma(t_1,\ldots,t_m) = t$.

Now suppose that there exists a tree $t' \in T_\Delta$ satisfying $q(t') \Rightarrow_{T'}^* t$. We prove $q(t) \Rightarrow_T^* t$ also by induction on height(t). Recall that, since T' is relabeling, height$(t') =$ height(t) necessarily holds.

Basis. Suppose that height$(t) = 0$. Then $t = \delta$, for some $\delta \in \Sigma_0$. By the construction of T', $t' = \delta_q \in \Delta_0$ and $q(\delta_q) \to \delta \in R'$, hence $q(\delta) \to \delta \in R$. Therefore, $q(t) \Rightarrow_T^* t$ holds.

Induction step. Let $n \geq 0$. Suppose that height$(t) = n + 1$, then $t = \sigma(t_1,\ldots,t_m)$, for some $m \geq 1$, $\sigma \in \Sigma_m$, and $t_1,\ldots,t_m \in T_\Sigma$, where height$(t_i) \leq n$ holds for all $1 \leq i \leq m$. By the construction of T', for some $q_1,\ldots,q_m \in Q$ and $t'_1,\ldots,t'_m \in T_\Delta$, $t' = \sigma_{q,q_1,\ldots,q_m}(t'_1,\ldots,t'_m)$ and $q(\sigma_{q,q_1,\ldots,q_m}(x_1,\ldots,x_m)) \to$

$\sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$ hold. Moreover $q_i(t_i') \Rightarrow_{T'}^* t_i$, for each $1 \leq i \leq m$. Hence $q(\sigma(x_1, \ldots, x_m)) \to \sigma(q_1(x_1), \ldots, q_m(x_m)) \in R$ and, by the induction hypothesis, $q_i(t_i) \Rightarrow_T^* t_i$, for all $1 \leq i \leq m$. Therefore, we have $q(t) \Rightarrow_T \sigma(q_1(t_1), \ldots, q_m(t_m)) \Rightarrow_T^* \sigma(t_1, \ldots, t_m) = t$. □

Lemma 4.1 implies that, for any tree $t \in T_\Sigma$, $q_0(t) \Rightarrow_T^* t$ holds if and only if there exists a tree $t' \in T_\Delta$ satisfying $q_0(t') \Rightarrow_{T'}^* t$. Hence $t \in L(T)$ if and only if $t \in$ range$(\tau_{T'})$. Since L was arbitrary and T' is rl-sl-dt tree transducer, it follows that $REC \subseteq$ range(*sl-DT*). On the other hand, range(*sl-DT*) $\subseteq$ range(*l-DT*) obviously holds and, by Corollary 6.6 of Chapter IV in [GécSte1], range(*l-DT*) $\subseteq REC$, thus we have the following result.

Theorem 4.2 *range(sl-DT) = range(l-DT) = REC*

5 Concluding remarks

In this paper we have considered the domain and range tree languages of superlinear deterministic top-down tree transformations. Our main results are as follows.

1. The class dom(*sl-DT*) is exactly *su-DREC*, i.e. the subclass of *DREC* consisting of that tree languages, which are recognized by semi-universal deterministic top-down tree recognizers.

2. For any deterministic recognizable tree language L, it is decidable whether L is in dom(*sl-DT*). Namely, L is in dom(*sl-DT*) if and only if the minimal dttr recognizing L is an su-dttr. Moreover, a decision procedure is given.

3. The class range(*sl-DT*) is exactly *REC*, that is the class of all recognizable tree languages.

Finally, we recall that, by (1) of Theorem 4.3 in [DánFül1], the hierarchy dom(*sl-DT*) $\subset$ dom(*sl-DT*^{2}) $\subset \ldots$ is proper and, for any $n \geq 1$, dom(*sl-DT*^{n}) $\subset$ *DREC* holds, where *sl-DT*^{n} is the n-fold composition of *sl-DT*. Hence, for $n \geq 2$, the characterization of the class dom(*sl-DT*^{n}) and the decidability of $L \in$ dom(*sl-DT*^{n}) with $L \in DREC$ may be a topic of further research.

Acknowledgement. The author is grateful to Zoltán Fülöp for his valuable suggestions regarding the result of Section 4.

References

[Bak] Baker, B. S., Composition of top-down and bottom-up treetransductions, Information and Control, 41 (1979) 186-213.

[DánFül1] Dányi, G. and Fülöp, Z., Superlinear Deterministic Top-down Tree Transducers, Mathematical Systems Theory, 29 (1996) 507-534.

[DánFül2] Dányi, G. and Fülöp, Z., Compositions with Superlinear Deterministic Top-down Tree Transducers, Theoretical Computer Science, accepted for publication

[Eng1] Engelfriet, J., Bottom-up and top-down tree transformations - a comparison, Mathematical Systems Theory, 9 (1975) 198-231.

[Eng2] Engelfriet, J., Top-down tree transducers with regular look-ahead, Mathematical Systems Theory, 10 (1977) 289-303.

[FülVág] Fülöp, Z. and Vágvölgyi, S., Results on compositions of deterministic root-to-frontier tree transformations, Acta Cybernetica, 8 (1987) 49-61.

[GécSte1] Gécseg, F. and Steinby, M., Tree Automata, Akadémiai Kiadó, Budapest, 1984.

[GécSte2] Gécseg, F. and Steinby, M., Minimal ascending tree automata, Acta Cybernetica, 4 (1980) 37-44.

[Rou] Rounds, W. C., Mappings on grammars and trees, Mathematical Systems Theory, 4 (1970) 257-287.

Received July, 1996