Superlinear Deterministic Top-Down Tree Transducers

Abstract of the Ph.D. Thesis

by

Gábor Dányi

József Attila University
Departments of Informatics
Szeged, Hungary

1997

Introduction

In theoretical computer science tree transducers have been studied since the early seventies. They are finite devices processing terms over ranked alphabets. Such terms are called trees in this area. A tree transducer induces a binary relation over tree sets, called a tree transformation.

Several types of tree transducers have been defined. Namely, the concept of the top-down tree transducer was introduced in [Rou] and [Tha1]. Then the notion of the bottom-up tree transducer was defined in [Tha2]. Later on, in order to increase the transformational capacity, more powerful devices were introduced, such as top-down tree transducers with regular look-ahead (see [Eng2]), macro tree transducers (see [EngVog1]), attributed tree transducers (see [Fül1]), high level tree transducers (see [EngVog2]), modular tree transducers (see [EngVog3]), and high level modular tree transducers (see [Vog]).

In this thesis we shall consider only deterministic top-down tree transducers and tree transformations induced by such tree transducers.

The motivation of studying top-down tree transducers is that they serve as formal models of syntax-directed compilers, thus tree transformations induced by top-down tree transducers are abstract models of translations realized by syntax-directed compilers, see [Eng4].

Several restricted subtypes of deterministic top-down tree transducers have been defined and investigated. In this thesis we work with, among others, total, linear, nondeleting, and homomorphism deterministic top-down tree transducers.

In our sense, a tree transformation class is generally a class of tree transformations induced by tree transducers of a certain type. Thus we can distinguish the class of deterministic top-down tree transformations, denoted by DT, and its subclasses of total, linear, nondeleting, and homomorphism deterministic top-down tree transformations, denoted by $t\text{-}DT$, $l\text{-}DT$, $nd\text{-}DT$, and HOM, respectively. Moreover, type properties can be combined resulting more special devices. For instance, we can speak about linear and nondeleting deterministic top-down tree transducers, of which the induced tree transformation class is denoted by $l\text{-}nd\text{-}DT$.

Investigating a certain type of top-down tree transducers, the questions naturally arise, what sets of trees can be processed by top-down tree transducers of that type, and what sets of trees can occur as results of such processings. 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 string languages, for tree languages there also exist finite state recognizers. Using these devices, we can define the classes of recognizable and deterministic recognizable tree languages, see [GécSte4]. It turned out that the domains of deterministic top-down tree transformations are exactly the deterministic recognizable tree languages. Moreover, the class of ranges of linear deterministic top-down tree transformations is exactly the class of recognizable tree languages.

Since tree transformations are binary relations over tree sets, the concept of their composition, denoted by $\circ$, is clear. Moreover, the composition operation can naturally be extended to classes of tree transformations.

Compositions and decompositions of deterministic top-down tree transformation classes are of special interest, because they model consecutive applications of deterministic top-down tree transducers of certain types to tree languages in such a way that the output of a device is the input of its successor. The motivation of studying compositions comes from the fact that applying deterministic top-down tree transducers in succession can yield extra computational power in the sense that the resulting tree transformation cannot be induced in general by a single deterministic top-down tree transducer. Similarly, the investigation of decompositions is motivated by the intention that one would like to know whether a deterministic top-down tree transformation of a certain type could also be induced by the consecutive application of two or more deterministic top-down tree transducers of simpler types.

Top-down tree transducers and top-down tree transformations were studied in a large number of papers.

In pioneer works [Rou], [Tha1], [Eng1], [Eng3], [Bak1], [Bak2] and [Bak3] several restricted types (total, linear, nondeleting, etc.) were defined, the transformational power of different types were compared to each other, and some closure properties of the corresponding tree transformation classes were explored. A good survey of these results can be found in [GécSte4]. Moreover, [FülVág1] also contains important observations concerning closure properties of the class of deterministic top-down tree transformations and its subclasses.

Recognizability of domains and ranges of top-down tree transformations also have been studied very intensively, see [Rou], [Eng2], [GécSte4], [FülVág1], and [FülVág3].

The undecidability of equivalence problem of top-down tree transducers in the general case immediately follows from the result of [Gri] on the undecidability of equivalence problem of GSM's. On the other hand, it turned out that the equivalence is decidable in the deterministic case, see [Ési1] and [Zac]. Moreover, the equivalence problem were studied for some other restricted nondeterministic types in [AndBos]. The decidability of some other properties (injectivity, recognizability of the range tree set, etc.) have also been investigated, see [Ési1], [Ési2], [Fül4], and [FülGye].

Compositions and decompositions of tree transformation classes have been studied very intensively. Almost all papers regarding tree transducers contains such results and hence a huge number of decomposition and composition equations have been obtained. It was desirable to find a uniform way for researching this area. In [FülVág4] and [FülVág6] a general method was proposed for developing an algorithm, which, for an arbitrarily fixed base set of tree transformation classes, can decide the relationship concerning the inclusions between tree transformation classes obtained by composition from base classes. The method has numerous applications using different base sets of tree transformation classes, see [FülVág4], [FülVág5], [Fül2], [SluVág], and [GyeVág].

The subject of this thesis is the characterization of a new subtype of deterministic top-down tree transducers, called *superlinear deterministic top-down tree transducers*. We denote the class of superlinear deterministic top-down tree transformations by $sl\text{-}DT$. Superlinear deterministic top-down tree transducers are specialized linear deterministic top-down tree transducers and it holds that $sl\text{-}DT \subset l\text{-}DT$.

The concept of superlinear deterministic top-down tree transducers was proposed by H. Vogler during a personal communication with Z. Fülöp in 1992. It was motivated by the well known decomposition equation $DT = nd\text{-}HOM \circ l\text{-}DT$, which appeared first time in [Eng1] and [Bak3]. They discussed whether $l\text{-}DT$ in the above equation can be substituted by an even more restricted subclass of DT. It was guessed that a proper subclass of $l\text{-}DT$ would be suitable, namely the class $sl\text{-}DT$ of superlinear deterministic top-down tree transducers.

In this work we investigate properties of superlinear deterministic top-down tree transducers and the corresponding tree transformation class $sl\text{-}DT$. Although the starting point of the research was the decomposition equation $DT = nd\text{-}HOM \circ sl\text{-}DT$, we have explored many other interesting results concerning superlinear deterministic top-down tree transducers and transformations. The problems we have arisen and answered were motivated by the earlier works regarding tree transducers (see, e.g., [Eng1], [Eng3], [Bak3], [FülVág1], [FülVág2]). These are as follows:

- What is the relationship between $sl\text{-}DT$ and the known tree transformation classes, such as DT or $l\text{-}DT$? In other words, how does the superlinear deterministic top-down tree transducers compare with the known types regarding transformational power?
- Is the class $sl\text{-}DT$ closed under the composition?
- What kind of tree languages can be domain and range tree languages of superlinear deterministic top-down tree transformations?
- How can we characterize the compositions of $sl\text{-}DT$ with other known classes?

This thesis is strongly based on the papers [DánFül1], [DánFül2] and [Dán]. All results presented here appear in the above works. We shall refer to the corresponding paper at the beginning of each chapter.

Results and outline of the thesis

Chapter 1: Preliminaries

Here we introduce notions and notations. Moreover, we recall some earlier results used in later chapters.

In this abstract we introduce only that concepts and notations, which are necessary to understand the results presented here.

Let A, B, and C be sets, moreover, let $\theta \subseteq A \times B$ and $\mu \subseteq B \times C$ be relations. The sets $\mathrm{dom}(\theta) = \{a \in A \mid \text{there is a } b \in B \text{ such that } a\theta b\}$ and $\mathrm{range}(\theta) = \{b \in B \mid \text{there is an } a \in A \text{ such that } a\theta b\}$ are called the *domain* and the *range* of θ, respectively. The *composition* $\theta \circ \mu$ of θ and μ is a relation from A to C defined as $\theta \circ \mu = \{(a, c) \mid \text{there is a } b \in B \text{ such that } a\theta b \text{ and } b\mu c\}$.

Let $\mathcal{C}$ and $\mathcal{C}'$ be classes of relations. 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. The composition of $\mathcal{C}$ and $\mathcal{C}'$ is the relation class $\mathcal{C} \circ \mathcal{C}' = \{\theta \circ \mu \mid \theta \in \mathcal{C} \text{ and } \mu \in \mathcal{C}'\}$.

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.

Let Σ be a ranked alphabet. For a set H, the *set of trees over* Σ *indexed by* H is denoted by $T_\Sigma(H)$ and it is defined as the smallest set U satisfying the following two 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 written as T_Σ.

We specify a countable set $X = \{x_1, x_2, \ldots\}$ of symbols, called *variables*, and we put $X_m = \{x_1, \ldots, x_m\}$, for every $m \geq 0$. We assume that X is disjoint to any ranked alphabet. We write $T_{\Sigma,m}$ for $T_\Sigma(X_m)$. We distinguish a subset $\hat{T}_{\Sigma,m}$ of $T_{\Sigma,m}$ as follows. A tree $t \in T_{\Sigma,m}$ is in $\hat{T}_{\Sigma,m}$ if and only if each variable in X_m appears exactly once in t and the order of the variables from left to right in t is exactly $x_1, \ldots, x_m$.

We introduce the concept of *tree substitution*. Let $m \geq 0$, $t \in T_{\Sigma,m}$, and $s_1, \ldots, s_m \in S$ where S is an arbitrary set of trees. We denote by $t[s_1, \ldots, s_m]$

the tree, which is obtained from t by replacing each occurrence of x_i in t by s_i, for every $1 \leq i \leq m$.

A *tree language* L over a ranked alphabet Σ is a subset $L \subseteq T_\Sigma$. Let Σ and Δ be ranked alphabets. A *tree transformation from T_Σ to T_Δ* is a relation from T_Σ to T_Δ. We specify the class $I = \{\mathrm{id}(T_\Sigma) \mid \Sigma \text{ is a ranked alphabet}\}$ of identity tree transformations.

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

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

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

  called (q, σ)-rules, 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 $t \in \hat{T}_{\Delta,n}$. The rule is said to be *reducing* if $t = x_1$ holds, i.e. it is of the form $q(\sigma(x_1, \ldots, x_m)) \to q'(x_i)$, for some $q' \in Q$ and $1 \leq i \leq m$.

The rules in R induce a relation, called *derivation*, denoted by $\Rightarrow_T$, over the set $T_\Delta(Q(T_\Sigma))$, where $Q(T_\Sigma)$ denotes the set $\{q(t) \mid q \in Q, t \in T_\Sigma\}$. 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 s is obtained from r by replacing an occurrence of a subtree $q(\sigma(t_1, \ldots, t_m))$ of r by $t[q_1(t_{i_1}), \ldots, q_n(t_{i_n})]$, where $t_1, \ldots, t_m \in T_\Sigma$. The *tree transformation τ_T induced by T* is defined as $\tau_T = \{(r, s) \in T_\Sigma \times T_\Delta \mid q_0(r) \Rightarrow_T^* s\}$.

We say that T is *deterministic* if, for any $q \in Q$ and $\sigma \in \Sigma$, there is at most one (q, σ)-rule in R. The expression deterministic top-down tree transducer is abbreviated to dt tree transducer in the sequel. A tree transformation τ is called a *dt tree transformation* if a dt tree transducer T exists so that $\tau = \tau_T$. The class of all dt tree transformations is denoted by DT.

Suppose that T is deterministic and consider an arbitrary (q, σ)-rule in R of the above $(*)$ form. Then 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 $\mathrm{rhs}(q, \sigma)$.

Let $T = (Q, \Sigma, \Delta, q_0, R)$ be a dt tree transducer. We say that T is:

- *Total* (t) if, for any $\sigma \in \Sigma$ and $q \in Q$, there is a (q, σ)-rule in R.
- *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.

- *Superlinear* (sl) if it is linear and, for every $\sigma \in \Sigma_m$ with $m \geq 0$ and $1 \leq i \leq m$, there is at most one state $q \in Q$ such that x_i occurs in rhs(q, σ).
- *Nondeleting* (nd) if, for every $q(\sigma(x_1, \ldots, x_m)) \to t[q_1(x_{i_1}), \ldots, q_n(x_{i_n})]$ in R, each of the variables $x_1, \ldots, x_m$ appears at least once in rhs(q, σ).
- *Homomorphism* (hom) if it is total and Q is a singleton set, i.e. $Q = \{q_0\}$.

These attributes can be combined. For instance, by an l-nd-dt tree transducer, we mean a linear and nondeleting dt tree transducer.

A *top-down tree recognizer (ttr)* $T = (Q, \Sigma, \Sigma, q_0, R)$ is a top-down tree transducer, 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 $m \geq 0$, $\sigma \in \Sigma_m$, and $q, q_1, \ldots, q_m \in Q$. If T is deterministic, then it is called a *deterministic top-down tree recognizer (dttr)*.

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. $\{t \in T_\Sigma \mid q(t) \Rightarrow_T^* t\} = T_\Sigma$.

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 $L(T) = \{t \in T_\Sigma \mid q_0(t) \Rightarrow_T^* t\}$. 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.

Chapter 2: Properties of sl-dt tree transducers

In this chapter we investigate some properties of superlinear deterministic top-down tree transducers.

2.1 Basic properties

The main results are as follows.

Theorem 2.1.4 [DánFüll] *sl-DT^2 − $DT \neq \emptyset$*

This implies that the tree transformation class *sl-DT* is not closed under the composition.

Lemma 2.1.5 [DánFüll] *l-$DT \subseteq$ nd-$HOM \circ$ sl-DT*

The following two results can easily be derived from the above lemma:

Theorem 2.1.6 [DánFüll] *DT = nd-$HOM \circ$ sl-DT*

Corollary 2.1.7 [DánFüll] *DT^2 = nd-$HOM \circ$ sl-DT^2*

We note that, by earlier results (see [FülVág1]), any tree transformation defined by composition of dt tree transformations can be substituted by composition of two appropriate dt tree transformations, i.e. DT^2 consists of all tree transformations, which can be realized by any sequence of dt tree transducers.

It is known (see, e.g., [FülVág1]) that the syntactic composition of dt tree transducers preserves the properties t, l, nd, and hom. We have studied this problem for the sl property and have obtained the following:

Lemma 2.1.11 [DánFül1] *The syntactic composition $T'' = T \circ T'$ of two sl-dt tree transducers $T = (Q, \Sigma, \Delta, q_0, R)$ and $T' = (Q', \Delta, \Omega, q_0', R')$ is an sl-dt tree transducer if and only if T has no reducing rule or Q' is a singleton set.*

Theorem 2.1.4 shows that there are two sl-dt tree transformations such that their composition cannot be induced by a dt and hence by an l-dt tree transducer. This suggests that the consecutive application of a sequence of sl-dt tree transducers can have big transformation power. However, the next theorem shows that this is not the case. Namely, we show that generally even the total l-dt tree transformations cannot be induced by sequences of sl-dt tree transducers.

Theorem 2.1.13 [DánFül1] $t\text{-}l\text{-}DT - sl\text{-}DT^+ \neq \emptyset$

2.2 Domain tree languages

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.

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 $1 \leq i \leq m$, at least one of q_i and p_i is universal. We denote by $su\text{-}DREC$ the class of tree languages recognized by su-dttr's.

Theorem 2.2.3 [Dán] $dom(sl\text{-}DT) = su\text{-}DREC$

Theorem 2.2.6 [Dán] *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.*

2.3 Range tree languages

We have obtained the following characterization of range tree languages of sl-dt tree transformations:

Theorem 2.3.2 [Dán] $range(sl\text{-}DT) = range(l\text{-}DT) = REC$

Chapter 3: Hierarchy theorems of sl-dt tree transformations

It turned out in the previous chapter that, similarly to the classes DT and $l\text{-}DT$, the class $sl\text{-}DT$ is not closed under the composition.

However, in Section 3.1 we show that, in contrast with the classes DT and $l\text{-}DT$, the hierarchy $\{sl\text{-}DT^n \mid n \geq 0\}$ never collapses. More exactly, we prove a stronger statement, namely that $\{\mathrm{dom}(sl\text{-}DT^n) \mid n \geq 1\}$ is a proper hierarchy. Moreover, we prove in Section 3.2 that even the hierarchy $\{t\text{-}sl\text{-}DT^n \mid n \geq 0\}$ is proper.

3.1 The hierarchies dom($sl\text{-}DT^n$) and $sl\text{-}DT^n$

Theorem 3.1.3 [DánFüll] *For any integer $n \geq 1$, the following inclusions hold:*

(1) $dom(sl\text{-}DT^n) \subset dom(sl\text{-}DT^{n+1})$.

(2) $dom(sl\text{-}DT^n) \subset DREC$.

(3) $sl\text{-}DT^n \subset sl\text{-}DT^{n+1}$.

Moreover, we present the inclusion diagram (see Figure 3.2) of the classes DT^2, DT, $l\text{-}DT^2$, $l\text{-}DT$, and $sl\text{-}DT^n$ with $n \geq 1$.

3.2 The hierarchy $t\text{-}sl\text{-}DT^n$

Theorem 3.2.3 [DánFüll] *For any integer $n \geq 1$, $t\text{-}sl\text{-}DT^n \subset t\text{-}sl\text{-}DT^{n+1}$ holds.*

Moreover, we present the inclusion diagram (see Figure 3.3) of the tree transformation classes $t\text{-}DT$, $t\text{-}l\text{-}DT$, and $t\text{-}sl\text{-}DT^n$ with $n \geq 1$.

Chapter 4: Compositions with sl-dt tree transformations

In this chapter we explore how the class $sl\text{-}DT$ behaves when composing with other known tree transformation classes. These other classes are HOM, $l\text{-}DT$, $nd\text{-}DT$, and DT.

On behalf of this, we fix the set $M = \{HOM, sl\text{-}DT, l\text{-}DT, nd\text{-}DT, DT\}$ of tree transformation classes. Then we consider the monoid $[M]$ of all tree transformation classes of the form $X_1 \circ \ldots \circ X_m$, where $m \geq 0$ and the X_i's are elements of M. For arbitrary composition classes $\mathcal{C}_1$ and $\mathcal{C}_2$ of the above form, we want to know whether some inclusion, equality, or incomparability holds between them. Clearly, it is enough if we can decide whether $\mathcal{C}_1 \subseteq \mathcal{C}_2$ holds.

As the main result of the chapter, we give an effective description of the monoid $[M]$ with respect to the inclusion. This means that we present an algorithm, which can decide, given arbitrary two elements of the monoid, whether some inclusion, equality, or incomparability holds between them.

The main steps of the development of this algorithm are as follows:

(1) We consider the free monoid $(M^*; \cdot, e)$ of strings generated by M. Then a unique homomorphism $|\ | : M^* \to [M]$ exists such that, for any $X_1, \ldots, X_n \in M$, $|X_1 \cdot \ldots \cdot X_n| = X_1 \circ \ldots \circ X_n$ (see [BurSan]). We denote the kernel of $|\ |$ by θ, that is, for any strings $u, v \in M^*$, $u \theta v$ if and only if $|u| = |v|$.

(2) We present a confluent and terminating rewriting system $R \subseteq M^* \times M^*$ (see Figure 4.1) such that $\Leftrightarrow_R^* = \theta$, where $\Leftrightarrow_R^*$ is the reflexive, symmetric, and transitive closure of the reduction relation $\Rightarrow_R$ over M^*.

Moreover, we show that the set of R-normal forms is exactly the following:

$$\begin{aligned} NF(R) = \ & \{l\text{-}DT^2, l\text{-}DT \cdot HOM, l\text{-}DT^2 \cdot nd\text{-}DT, DT^2\} \cup \\ & \{sl\text{-}DT^n \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot HOM \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot l\text{-}DT \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot nd\text{-}DT \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot l\text{-}DT \circ nd\text{-}DT \mid n \geq 0\} \cup \\ & \{sl\text{-}DT^n \cdot DT \mid n \geq 0\} \end{aligned}$$

(3) We present the inclusion diagram (see Figure 4.2), i.e. the Hasse diagram with respect to the inclusion of the set $|NF(R)| = \{|u| \mid u \in NF(R)\}$.

The main result of this chapter sounds as follows:

Theorem 4.2.8 [DánFül2] *For any two tree transformation classes $X_1 \circ X_2 \circ \ldots \circ X_m$ and $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ in $[M]$, it is decidable whether the inclusion $X_1 \circ X_2 \circ \ldots \circ X_m \subseteq Y_1 \circ Y_2 \circ \ldots \circ Y_n$ holds.*

The deciding algorithm works as follows. Given two arbitrary composition classes $\mathcal{C} = X_1 \circ \ldots \circ X_n$ and $\mathcal{D} = Y_1 \circ \ldots \circ Y_m$, we form the strings $x = X_1 \cdot \ldots \cdot X_n$ and $y = Y_1 \cdot \ldots \cdot Y_m$, and compute the corresponding R-normal forms $x \Rightarrow_R^* u$ and $y \Rightarrow_R^* v$. Since R is terminating and confluent, u and v exist and unique. Moreover, $|x| = |u|$ and $|y| = |v|$ hold by $\Leftrightarrow_R^* = \theta$. Hence $\mathcal{C} \subseteq \mathcal{D}$ if and only if $|u| \subseteq |v|$. However, this latter can be decided by direct inspection of the inclusion diagram of $NF(R)$ (see Figure 4.2).

We note that the results of this chapter can be found in [DánFül2].

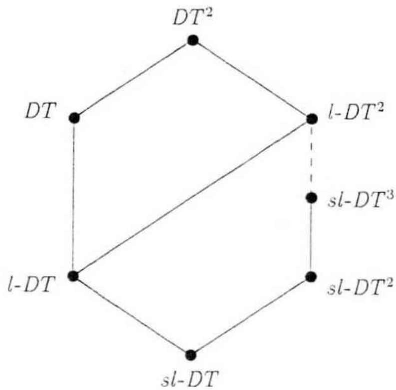

Figure 3.2: The hierarchy of $sl\text{-}DT^n$

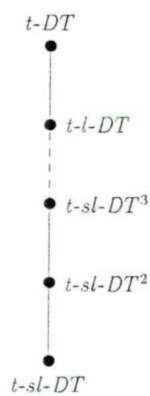

Figure 3.3: The hierarchy of $t\text{-}sl\text{-}DT^n$

$$
\begin{array}{rlcl}
(1) & l\text{-}DT^2 \cdot HOM & \rightarrow & l\text{-}DT \cdot HOM \\
(2) & HOM \cdot HOM & \rightarrow & HOM \\
(3) & DT \cdot HOM & \rightarrow & DT^2 \\
(4) & sl\text{-}DT \cdot l\text{-}DT \cdot HOM & \rightarrow & l\text{-}DT \cdot HOM \\
(5) & l\text{-}DT^3 & \rightarrow & l\text{-}DT^2 \\
(6) & l\text{-}DT \cdot sl\text{-}DT & \rightarrow & l\text{-}DT^2 \\
(7) & l\text{-}DT \cdot DT & \rightarrow & DT^2 \\
(8) & HOM \cdot l\text{-}DT & \rightarrow & DT \\
(9) & HOM \cdot sl\text{-}DT & \rightarrow & DT \\
(10) & HOM \cdot DT & \rightarrow & DT \\
(11) & DT \cdot l\text{-}DT & \rightarrow & DT^2 \\
(12) & DT \cdot sl\text{-}DT & \rightarrow & DT^2 \\
(13) & DT^3 & \rightarrow & DT^2 \\
(14) & sl\text{-}DT \cdot l\text{-}DT^2 & \rightarrow & l\text{-}DT^2 \\
(15) & sl\text{-}DT \cdot DT^2 & \rightarrow & DT^2 \\
(16) & nd\text{-}DT \cdot HOM & \rightarrow & DT^2 \\
(17) & nd\text{-}DT \cdot sl\text{-}DT & \rightarrow & DT^2 \\
(18) & nd\text{-}DT \cdot l\text{-}DT & \rightarrow & DT^2 \\
(19) & nd\text{-}DT \cdot nd\text{-}DT & \rightarrow & nd\text{-}DT \\
(20) & nd\text{-}DT \cdot DT & \rightarrow & DT^2 \\
(21) & l\text{-}DT \cdot HOM \cdot nd\text{-}DT & \rightarrow & l\text{-}DT^2 \cdot nd\text{-}DT \\
(22) & DT \cdot nd\text{-}DT & \rightarrow & DT
\end{array}
$$

Figure 4.1: Rewriting rules of R

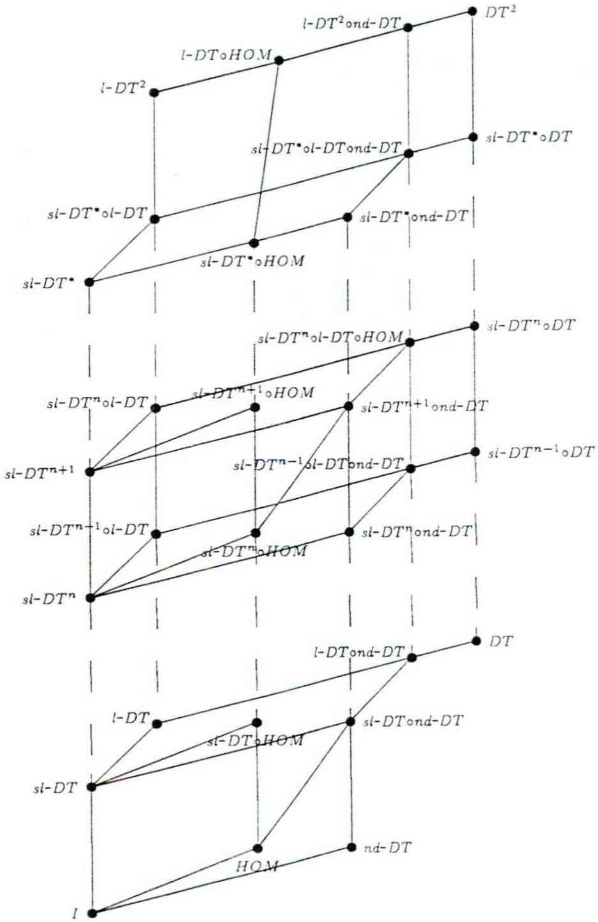

Figure 4.2: The inclusion diagram of normal forms

Conclusions

In this thesis we have considered superlinear deterministic top-down tree transducers and the class *sl-DT* of superlinear deterministic top-down tree transformations. Our main results are as follows:

- The classes *sl-DT* and *t-sl-DT* are not closed under the composition.
- $t\text{-}l\text{-}DT - sl\text{-}DT^+ \neq \emptyset$, where $sl\text{-}DT^+$ is the transitive closure of the class *sl-DT* under the composition. Roughly speaking, even the consecutive application of arbitrary many sl-dt tree transducers has no enough transformational power to generate all l-dt tree transformations.
- $DT = nd\text{-}HOM \circ sl\text{-}DT$, that is sl-dt tree transducers have enough computational capacity to generate all dt tree transformations with the help of nondeleting homomorphism top-down tree transducers.
- The class dom(*sl-DT*) is exactly *su-DREC*, i.e. the subclass of *DREC* consisting of those tree languages which are recognized by semi-universal deterministic top-down tree recognizers.
- For any deterministic recognizable tree language L, it is decidable whether L is in dom(*sl-DT*).
- The class range(*sl-DT*) is exactly *REC*, that is the class of all recognizable tree languages.
- The hierarchies
  $\{\mathrm{dom}(sl\text{-}DT^n) \mid n \geq 0\}$,
  $\{sl\text{-}DT^n \mid n \geq 0\}$, and
  $\{t\text{-}sl\text{-}DT^n \mid n \geq 0\}$
  are proper.
- We have considered a monoid $[M]$ generated by the tree transformation classes *HOM*, *sl-DT*, *l-DT*, *nd-DT*, and *DT* with the operation composition.
  Using string rewriting techniques, we have developed an algorithm which,

given any two elements $X_1 \circ X_2 \circ \ldots \circ X_m$ and $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ of $[M]$, can decide whether the inclusion $X_1 \circ X_2 \circ \ldots \circ X_m \subseteq Y_1 \circ Y_2 \circ \ldots \circ Y_n$ holds. We have represented elements of $[M]$ by strings, and have presented a terminating and confluent string rewriting system R as well as the inclusion diagram of the normal forms with respect to R.

The inclusion between two elements of $[M]$ can be decided in the following way. We reduce the strings representing the tree transformation classes $X_1 \circ X_2 \circ \ldots \circ X_m$ and $Y_1 \circ Y_2 \circ \ldots \circ Y_n$ to normal forms with respect to R. The string rewriting system R is constructed in such a way that $\subseteq$ (resp. $\supseteq$, $=$, incomparability) holds between the two tree transformation classes if and only if the same relation holds between the tree transformation classes represented by the corresponding normal forms. However, this latter can be read from the inclusion diagram depicted in Figure 4.2.

Bibliography

[AndBos] Andre, Y. and Bossut, F., Decidability of equivalence for linear letter-to-letter top-down tree transducers, Proceedings of FCT '93 (Z. Ésik ed.), Lecture Notes in Computer Science 710 (1993) 142-151.

[Bak1] Baker, B. S., Tree transducers and families of tree languages, 5th Ann. ACM STC (1973), 200-206.

[Bak2] Baker, B. S., Tree transducers and tree languages, Information and Control 37 (1978), 241-266.

[Bak3] Baker, B. S., Composition of top-down and bottom-up tree transductions, Information and Control 41 (1979) 186-213.

[Boo] Book, R. V., Thue systems and the Church-Rosser Property: replacement systems, specification of formal languages and presentation of monoids, Progress in combinatorics on words (L. Cummings ed.), 1-38, Academic Press, 1983, New York

[BooOtt] Book, R. V. and Otto, F., String-rewriting systems, Springer-Verlag, 1993.

[BurSan] Burris, S. and Sankappanavar, H. P., A course in universal algebra, Springer-Verlag, 1981.

[Dán] Dányi, G., On domain and range tree languages of superlinear deterministic top-down tree transformations, Acta Cybernetica, 12 (1996) 216-277.

[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.

[Eng3] Engelfriet, J., Three hierarchies of transducers, Mathematical Systems Theory 15 (1982) 95-125.

[Eng4] Engelfriet, J., Tree transducers and syntax-directed semantics, Proceedings of the 7th CAAP, Lille, 1982, 95-125.

[EngVog1] Engelfriet, J. and Vogler, H., Macro tree transducers, Journal of Computational System Sciences 31 (1985) 71-146.

[EngVog2] Engelfriet, J. and Vogler, H., High level tree transducers and iterated push-down tree transducers, Acta Informatica 26 (1988) 131-192.

[EngVog3] Engelfriet, J. and Vogler, H., Modular tree transducers, Theoretical Computer Science 78 (1991) 267-303.

[Ési1] Ésik, Z., Decidability results concerning tree transducers I., Acta Cybernetica 5 (1980) 1-20.

[Ési2] Ésik, Z., Decidability results concerning tree transducers II., Acta Cybernetica 6 (1983) 303-314.

[Fül1] Fülöp, Z., On attributed tree transducers, Acta Cybernetica 5 (1981) 261-279.

[Fül2] Fülöp, Z., A complete description for a monoid of deterministic bottom-up tree transformation classes, Theoretical Compututer Science 88 (1991) 253-268.

[Fül3] Fülöp Z., Eldönthetőségi kérdések fatranszformáció osztályok által generált monoidokban, Alkalmazott Matematikai Lapok 15 (1990-91), 219-266.

[Fül4] Fülöp, Z., Undecidable properties of deterministic top-down tree transducers, Theoretical Computer Science 134 (1994) 311-328.

[FülGye] Fülöp, Z. and Gyenizse, P., On injectivity of deterministic top-down tree transducers, Information Processing Letters 48 (1993), 183-188.

[FülVág1] Fülöp, Z. and Vágvölgyi, S., Results on compositions of deterministic root-to-frontier tree transformations, Acta Cybernetica 8 (1987) 49-61.

[FülVág2] Fülöp, Z. and Vágvölgyi, S. An infinite hierarchy of tree transformations in the class NDR, Acta Cybernetica 8 (1987) 153-168.

[FülVág3] Fülöp, Z. and Vágvölgyi, S., On domains of tree transducers, Bulletin of the EATCS 34 (1988) 55-61.

[FülVág4] Fülöp, Z. and Vágvölgyi, S., A finite presentation for a monoid of tree transformation classes, Proceedings of 2nd Conference on Automata, Languages, and Programming Systems (F. Gécseg and I. Peák Eds.), Salgótarján, 1988.

[FülVág5] Fülöp, Z. and Vágvölgyi, S., A complete classification of deterministic root-to-frontier tree transformation classes, Theoretical Compututer Science 81 (1991) 1-15.

[FülVág6] Fülöp, Z. and Vágvölgyi, S., Decidability of the inclusion in monoids generated by tree transformation classes, Tree Automata and Languages (M. Nivat and A. Podelski eds.), Elsevier Science Publishers B. V., Amsterdam, 1992, 381-408.

[GécSte1] Gécseg, F. and Steinby, M., A faautomaták algebrai elmélete I., Matematikai Lapok 26 (1978), 169-207.

[GécSte2] Gécseg, F. and Steinby, M., A faautomaták algebrai elmélete II., Matematikai Lapok 27 (1979), 283-366.

[GécSte3] Gécseg, F. and Steinby, M., Minimal ascending tree automata, Acta Cybernetica 4 (1980) 37-44.

[GécSte4] Gécseg, F. and Steinby, M., Tree automata, Akadémiai Kiadó, Budapest, 1984.

[Gri] Griffiths, T. V., The unsolvability of the equivalence problem for Λ-free nondeterministic generalized machines, Journal of the ACM 15 (1968) 409-413.

[GyeVág] Gyenizse, P. and Vágvölgyi, S., Compositions of deterministic bottom-Up, top-down, and regular look-ahead tree transformations, Theoretical Compututer Science 156 (1996) 71-97.

[Huet] Huet, G., Confluent Reductions: Abstract properties and applications to term rewriting systems, Journal of the Association for Computing Machinery 4 (1980) 797-821.

[Jan] Jantzen, M., Confluent string rewriting, Springer-Verlag, 1988.

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

[SluVág] Slutzki, G. and Vágvölgyi, S., A hierarchy of deterministic top-down tree transformations, Proceedings of FCT '93 (Z. Ésik ed.), Lecture Notes in Computer Science 710 (1993) 440-451.

[Tha1] Thatcher, J. W., Generalized sequential machine maps, Journal of Computational System Sciences 4 (1970) 339-367.

[Tha2] Thatcher, J. W., Tree auttomata: an informal survey, Currents in the Theory of Computing (A. V. Aho ed.) Prenttice-Hall, 1973, 143-172.

[Vog] Vogler, H., High level modular tree transducers, 1978-1988, Ten years IIG, Institutes for Information Processing (H. J. Pongratz and W. Schinnerl eds.), Report Nr. 260, Institutes for Information Processing, Graz University of Technology

[Zac] Zachar, Z., The solvability of the equivalence problem for deterministic frontier-to-root tree transducers, Acta Cybernetica 4 (1979), 167-177.