A Hierarchy of Deterministic Top-down Tree Transformations

TR 93-10a

Giora Slutzki and Sandor Vagvolgyi

April 13, 1993

Iowa State University of Science and Technology
Department of Computer Science
226 Atanasoff
Ames, IA 50011

A Hierarchy of Deterministic Top-down Tree Transformations

Giora Slutzki and Sándor Vágvölgyi*
Department of Computer Science
Iowa State University
Ames, Iowa 50011, USA

Abstract

The class DTT^{DR} (respectively, DTT) is the family of all deterministic top-down tree transductions with deterministic top-down look-ahead (respectively, no look-ahead). In this paper we prove that the two hierarchies : $(DTT^{DR})^n$ and $(DTT^{DR})^n \circ DTT$ are proper and that they "shuffle perfectly" in the sense that $(DTT^{DR})^n \circ DTT$ is properly contained in $(DTT^{DR})^{n+1}$, for all $n \geq 0$. Using these results we show that the problem of determining the correct inclusion relationship between two arbitrary compositions of tree transformation classes from the set $M = \{ DTA, DTT, DTT^{DR}, DTT^R \}$ can be decided in linear time.

1 Introduction

There is a considerable interest in finding inclusions and equalities that hold for compositions of tree transformation classes. Such results are, for example, the following six ones.

(a) $DTA \circ DTA = DTA$	(b) $DTT^R \circ DTT^R = DTT^R$
(c) $DTT \subset DTT \circ DTT$	(d) $DTA \circ DTT = DTT^2 = DTT^3$
(e) $DTT^{DR} \subset DTT^{DR} \circ DTT^{DR}$	(f) $DTT \circ DTT \subset DTT^{DR}$

where DTA (respectively, DTT) stands for the class of tree transformations induced by deterministic top-down tree automata (respectively, transducers) and the superscript R (respectively, DR) stands for regular (respectively, deterministic top-down) look-ahead (see [5],[15],[9],[10]). Results (a), (b) mean that DTA, DTT^R are closed under composition, the results (c), (e) states that DTT, DTT^{DR} are not closed under composition, and (d) means

*On leave from Research Group on Theory of Automata, Hungarian Academy of Sciences, supported by a grant from the Soros Foundation.

that the composition of three deterministic top-down tree transformations can be computed by the composition of two, and moreover, the first one can be a deterministic top-down tree automaton. It is easy to see that from the already verified equalities and inclusions we obtain new ones by substituting either side of a valid equation for an occurrence of the other side. For example, (c) and (d) imply :

$$\text{(g)}\ \ DTT \subset DTA \circ DTT,$$

and (a), (d), (f), and (e) imply :

$$\text{(h)}\ \ DTA \circ DTA \circ DTT \circ DTT \circ DTT \subset DTT^{DR} \circ DTT^{DR}.$$

One may naturally raise the question whether (a)-(f) can be completed with finitely many other inclusions and equations such that by applying substitutions we can derive every inclusion and equation which holds among the compositions of DTA, DTT, DTT^{DTR}, DTT^{R}. In general, one may be interested in generating all valid equalities and inclusions between compositions of tree transformation classes that are taken from a given finite reservoir of such classes. We formalize these questions in the following way. Let M be a finite set of tree transformation classes. We consider two monoids defined in terms of M: the *free monoid* M^* (with the operation of concatenation denoted by "$\cdot$") and $[M]$, the monoid finitely generated by M (with the operation of composition denoted by "$\circ$"). Strings over M represent transformation classes in $[M]$ by means of a homomorphism $||\ || : M^* \to [M]$ defined by

$$||Y_1 \cdot Y_2 \cdot \ldots \cdot Y_m|| \;=\; Y_1 \circ Y_2 \circ \ldots \circ Y_m \quad .$$

We denote by $I \in [M]$ the tree transformation class consisting of all identity tree transformations, i.e., $I = ||\lambda||$. Let θ be the *kernel* of $||\ ||$, i.e., the *congruence* relation induced by the homomorphism $||\ ||$:

$$\theta = ker(||\ ||) = \{(v, w) \in M^* \times M^* \mid\ ||v|| \;=\; ||w||\,\} \quad .$$

Fülöp and Vágvölgyi [12] raised the following problem. Give an algorithm which, given $Y_1, \ldots, Y_m, Z_1, \ldots, Z_n \in M$, decides which one of the following four conditions holds:

$$\begin{array}{ll} \text{(i)}\ Y_1 \circ \ldots \circ Y_m \;=\; Z_1 \circ \ldots \circ Z_n\ , & \text{(ii)}\ Y_1 \circ \ldots \circ Y_m \;\subset\; Z_1 \circ \ldots \circ Z_n\ , \\ \text{(iii)}\ Z_1 \circ \ldots \circ Z_n \;\subset\; Y_1 \circ \ldots \circ Y_m\ , & \text{(iv)}\ Y_1 \circ \ldots \circ Y_m \;\bowtie\; Z_1 \circ \ldots \circ Z_n\ , \end{array}$$

where "$\bowtie$" stands for the incomparability relationship. They suggested an approach by which such an algorithm can be constructed provided M is not "too general". Our aim is to apply this approach for the monoid $M = \{\, DTA, DTT, DTT^{DR},\ DTT^{R}\,\}$. The choice of M was motivated by equations and inclusions (a)-(f) and by the interesting hierarchy results we obtain for $[M]$. Specifically, we show that $(DTT^{DR})^n$ and $(DTT^{DR})^n \circ DTT$

form two proper hierarchies and that the second hierarchy fits perfectly and properly "in between" the consecutive levels of the first hierarchy, i.e. for all $n \geq 0$:

$$(DTT^{DR})^n \circ DTT \subset (DTT^{DR})^{n+1} \ .$$

The paper is organized as follows. In Section 2 we introduce and recall the notation and basic concepts to be used. In Section 3 we outline the method of the paper. In Section 4 we give a Thue system $T_M \subseteq M^* \times M^*$ which is our candidate for the set of generators of θ. T_M contains the previously cited composition results (a), (b), and the "first half" of (d). Then we prove a "soundness" result for T_M, i.e., that for every $(u,v) \in T_M$, $||u|| = ||v||$. In this way elements of T_M represent equalities over $[M]$. In Section 5 we give a subset N of M^* which is a candidate for a set of representatives for the congruence classes of θ. Then we give the inclusion diagram of the set $\{ ||u|| \mid u \in N \}$, which is, in fact, the set of tree transformation classes represented by the elements of N. In Section 6, we show that the linear time algorithm of [3] can be applied such that, given $w \in M^*$, it computes a representative $u \in N$ of the congruence class of w. In Section 7, we summarize our results.

2 Preliminaries

2.1 Tree Transducers

A *ranked alphabet* Σ is an alphabet in which every symbol has a unique *rank* (*arity*) in the set of nonnegative integers. For any $m \geq 0$, we denote by Σ_m the set of symbols in Σ which have rank m. For a ranked alphabet Σ and a set H, *the set of trees* (or *terms*) *over* Σ *indexed by* H, denoted by $T_\Sigma(H)$, is 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 of *trees over* Σ is $T_\Sigma(\emptyset)$, and we simply write T_Σ for $T_\Sigma(\emptyset)$. We specify a countable set $X = \{ x_1, x_2, ... \}$ of variables and set $X_m = \{ x_1, \ldots, x_m \}$ for every $m \geq 0$. We distinguish a subset $\bar{T}_\Sigma(X_m)$ of $T_\Sigma(X_m)$ as follows: a tree $t \in T_\Sigma(X_m)$ is in $\bar{T}_\Sigma(X_m)$ if and only if each variable in X_m appears exactly once in t and the order of the variables in t is $x_1, \ldots, x_m$. For example, if $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{ a \}$ and $\Sigma_2 = \{ \sigma \}$, then $\sigma(x_1, \sigma(a, x_1)) \in T_\Sigma(X_1)$ but $\sigma(x_1, \sigma(a, x_1)) \notin \bar{T}_\Sigma(X_1)$. On the other hand, $\sigma(x_1, \sigma(a, x_2)) \in \bar{T}_\Sigma(X_2)$. The notion of *tree substitution* is defined as follows. Let $m \geq 0$, $t \in T_\Sigma(X_m)$ and $h_1, \ldots, h_m \in H$, where H is an arbitrary set. We denote by $t[h_1, \ldots, h_m]$ the tree which is obtained from t by replacing each occurrence of x_i in t by h_i for every $1 \leq i \leq m$. Let Σ and Δ be two ranked alphabets. Then any subset of $T_\Sigma \times T_\Delta$ is a *tree transformation* from T_Σ to T_Δ. For a tree language L, the partial identity $\{ (t,t) \mid t \in L \}$ is denoted by $ID(L)$.

Definiton 2.1 A *top-down tree transducer* (tt for short) is a system $\mathcal{A} =< \Sigma, \Delta, A, A_0, P >$, where

(1) Σ is a ranked *input* alphabet;
(2) Δ is a ranked *output* alphabet;
(3) A is a ranked *state* alphabet, it is a *unary* alphabet, i.e., $A = A_1$; also, $A \cap (\Sigma \cup \Delta \cup X) = \emptyset$;
(4) A_0 is a subset of A, the set of *initial* states;
(5) P is a finite set of *rules* of the form

$$a(\sigma(x_1, \ldots, x_m)) \to t$$

where $m \geq 0$, $\sigma \in \Sigma_m$, $a \in A$, and $t \in T_\Delta(A(X_m))$. (Here and in what follows, for a unary ranked alphabet A and a set L of terms, $A(L)$ denotes the set $\{ a(t) \mid a \in A \text{ and } t \in L \}$.)

Computation of tt's is formalized as follows. Define the binary relation $\Rightarrow_\mathcal{A}$ on the set $T_\Delta(A(T_\Sigma))$ so that for any $t, s \in T_\Delta(A(T_\Sigma))$, $t \Rightarrow_\mathcal{A} s$ if and only if the following two conditions hold:

(a) there is a rule $a(\sigma(x_1, \ldots, x_m)) \to r$ in P,
(b) s can be obtained from t by replacing an occurrence of a subtree $a(\sigma(t_1, \ldots, t_m))$ of t by $r[t_1, \ldots, t_m]$, where $t_1, \ldots, t_m \in T_\Sigma$.

Clearly, the relation $\Rightarrow_\mathcal{A}$ is interpreted as a method of rewriting terms into terms. The reflexive, transitive closure of $\Rightarrow_\mathcal{A}$, denoted by $\Rightarrow^*_\mathcal{A}$, is interpreted as the computation relation of $\mathcal{A}$. The tree transformation computed by $\mathcal{A}$ is the relation

$$\tau_\mathcal{A} = \{ (t, s) \in T_\Sigma \times T_\Delta \mid a(t) \underset{\mathcal{A}}{\overset{*}{\Rightarrow}} s \text{ for some } a \in A_0 \}.$$

We now introduce some special types of tt's. Let $\mathcal{A} = < \Sigma, \Delta, A, A_0, P >$ be a tt. We say that $\mathcal{A}$ is

(a) a *deterministic top-down tree transducer* (dtt) if A_0 is a singleton and there are no two different rules in P with the same left-hand side;
(b) a *top-down tree automaton* (ta) if $\Sigma = \Delta$ and each rule in P is of the form $a(\sigma(x_1, \ldots, x_m)) \to \sigma(a_1(x_1), \ldots, a_m(x_m))$ where $a, a_1, \ldots, a_m \in A$; in that case, the tree transformation $\tau_\mathcal{A}$ is a partial identity on T_Σ;
(c) a *deterministic top-down tree automaton* (dta) if $\mathcal{A}$ is a ta and a dtt.

The class of all tt's (respectively, dtt's, ta's, and dta's) is denoted by TT (respectively, DTT, TA, and DTA). The *tree language recognized* by a ta $\mathcal{A}$ is $L(\mathcal{A}) = dom(\tau_\mathcal{A})$. The classes of tree languages recognized by ta's and dta's are

$$R = dom(TA), \qquad \text{and} \qquad DR = dom(DTA).$$

Here R is the well-known class of *recognizable tree languages*, equal to the class of all tree languages definable by bottom-up tree automata. It is well known that $DR \subset R$ or equivalently $DTA \subset TA$; a proof can be found in [4] or [13].

Top-down tree transducers with look-ahead, one of the main topics of this paper, were defined in [5]. It transpired that they have a number of nice properties, especially in the deterministic case. For example, the class of deterministic top-down tree tree transformations with regular look-ahead is closed under composition. The concept of look-ahead also proved useful in other contexts [6], [7], [8]. Following [5], Fülöp and Vágvölgyi [9], [10] defined and studied top-down tree transducers and deterministic top-down tree automata with deterministic top-down look-ahead capacity.

Let $C \subset R$ be a class of tree languages. A *top-down tree transducer with C look-ahead* (tt^C) is a system $\mathcal{A} =< \Sigma, \Delta, A, A_0, P >$, where the components are defined exactly as in Definition 2.1, except that the rules in P are of the form

$$< a(\sigma(x_1, \ldots, x_m)) \to t;\ L_1, \ldots, L_m >$$

where

$$a(\sigma(x_1, \ldots, x_m)) \to t$$

is an ordinary tt-rule, as in Definition 2.1, and for each $1 \leq i \leq m$, $L_i \subseteq T_\Sigma$ is a language in C. The look-ahead tree languages $L_1, \ldots, L_m$ act as "guards" for the application of the above rule.

The one-step computation of $\mathcal{A}$ is the binary relation $\Rightarrow_\mathcal{A}$ on $T_\Delta(A(T_\Sigma))$ defined such that $t \Rightarrow_\mathcal{A} s$ if and only if

(a) there is a rule $< a(\sigma(x_1, \ldots, x_m)) \to r;\ L_1, \ldots, L_m >$ in P, and
(b) t has a subtree $t' = a(\sigma(t_1, \ldots, t_m))$ with $t_i \in L_i$ $(1 \leq i \leq m)$ and s is obtained by substituting $r[t_1, \ldots, t_m]$ for an occurrence of t' in t.

It can be seen from the definition of $\Rightarrow_\mathcal{A}$ what the notion look-ahead means: a rule can be applied at a node of a tree only if the direct subtrees of that node are in the tree languages given in the rule. As usual, $\Rightarrow^*_\mathcal{A}$, the reflexive, transitive closure of $\Rightarrow_\mathcal{A}$, formalizes the concept of computation of tt^C's, and the binary relation

$$\tau_\mathcal{A} = \{\, (t, s) \in T_\Sigma \times T_\Delta \mid a(t) \underset{\mathcal{A}}{\overset{*}{\Rightarrow}} s \ \text{ for some } a \in A_0 \,\}$$

defines the tree transformation induced by $\mathcal{A}$.

We define the following varieties of tt^C. Let $\mathcal{A} =< \Sigma, \Delta, A, A_0, P >$ be a tt^C. We say that $\mathcal{A}$ is

(a) a *top-down tree automaton with C look-ahead* (ta^C) if $\mathcal{A}$ is a tt^C with $\Sigma = \Delta$ and each rule in P is of the form $< a(\sigma(x_1, \ldots, x_m)) \to \sigma(a_1(x_1), \ldots, a_m(x_m));$ $L_1, \ldots, L_m >$ where $a_1, \ldots, a_m \in A$;

(b) *a deterministic top-down tree transducer with C look-ahead* (dtt^C) if A_0 is a singleton set and $L_i \cap L'_i = \emptyset$ holds for some i, $1 \le i \le m$, whenever $< a(\sigma(x_1, \ldots, x_m)) \to r_1;\ L_1, \ldots, L_m >$ and $< a(\sigma(x_1, \ldots, x_m)) \to r_2;\ L'_1, \ldots, L'_m >$ are different rules in P;

(c) *a deterministic top-down tree automaton with C look-ahead* (dta^C) if $\mathcal{A}$ is a ta^C and a dtt^C.

Note that if $\mathcal{A}$ is deterministic, then $\mathcal{A}$ can apply at most one rule at any given node. This is because for any two different rules in P with the same left-hand side there exists a variable x_i such that the two look-ahead sets corresponding to x_i are disjoint. The tree language recognized by a ta^C $\mathcal{A}$ is $L(\mathcal{A}) = dom(\tau_{\mathcal{A}})$. The class of all tree transformations defined by all tt^C's (respectively dtt^C's, ta^C's, and dta^C's) is denoted by TT^C (respectively DTT^C, TA^C, and DTA^C).

Example 2.2 Let $\Sigma = \Sigma_0 \cup \Sigma_2$ be a ranked alphabet, where $\Sigma_0 = \{1, 0\}$ and $\Sigma_2 = \{\sigma\}$. For each $m \ge 0$, define the tree $e_m \in T_\Sigma(X_{m+1})$ as follows: $e_0 = x_1$ and, for $m \ge 1$, $e_m = \sigma(x_1, e_{m-1}[x_2, \ldots, x_{m+1}])$, i.e., e_m is the tree $\sigma(x_1, \ldots, \sigma(x_m, x_{m+1}) \ldots)$. We say that a tree in T_Σ is even (odd) if it contains even (odd) number of 1's. We denote by L_e (L_o) the set of all even (odd) trees over Σ. Note that $0 \in L_e$ and $1 \in L_o$. For each integer $n \ge 0$, the tree language $C_n \subseteq T_\Sigma$ is defined as follows:

(a) $C_0 = \{1, 0\}$,

(b) for $n \ge 1$, C_n is the smallest set satisfying

  (i) $1, 0 \in C_n$ and

  (ii) $\sigma(t, r) \in C_n$ whenever $t \in C_{n-1}$ and $r \in C_n$.

The elements of C_n are called *n-nested combs*. Note that $C_n = \{ e_m(t_1, \ldots, t_m, y) \mid m \ge 0, y \in \{1, 0\} \text{ and } t_1, \ldots, t_m \in C_{n-1} \}$. Obviously, for $i < j$, we have $C_i \subset C_j$. We put $C_n^e = C_n \cap L_e$ and $C_n^o = C_n \cap L_o$.

The following result was proved in [5].

Proposition 2.3 Let $\mathcal{A}$ be a tt^R. Then $dom(\tau_{\mathcal{A}}) \in R$.

By Proposition 2.3, we can iterate the look-ahead tree languages, without leaving R, as follows. Let DR_0 be DR and let, for $n \ge 1$, DR_n be the class of tree languages recognizable by deterministic top-down tree automata with DR_{n-1} look-ahead. By Proposition 2.3, $DR_n \subseteq R$ for every $n \ge 0$. Fülöp and Vágvölgyi [11] proved the following result.

Proposition 2.4 For each $n \ge 1$, $C_n^e \in DR_n - DR_{n-1}$. Moreover, for every $n \ge 0$, DR_n is closed under intersection.

2.2 Thue Systems and String Rewriting Systems

Let Σ be an alphabet. The *empty string* and the *length* of a string $w \in \Sigma^*$ are denoted, respectively, by λ and $|w|$. Recall that Σ^* is the free monoid generated by Σ under the operation of concatenation with λ as identity. A *Thue system* T over Σ is a finite subset of $\Sigma^* \times \Sigma^*$ and each element (u, v) of T is called a *rewriting rule*. The *Thue congruence* generated by T is the reflexive, transitive closure $\leftrightarrow_T^*$ of the relation $\leftrightarrow_T$ defined as follows: for any $w, z \in \Sigma^*$, $w \leftrightarrow_T z$ if and only if there exist $x, y \in \Sigma^*$ and $(u, v) \in T$ such that either $w = xuy$ and $z = xvy$, or, $w = xvy$ and $z = xuy$. It is well-known that $\leftrightarrow_T^*$ is the least congruence over Σ^* containing T. The *reduction relation induced by* T is denoted by $\rightarrow_T$ and defined as follows: for any $w, z \in \Sigma^*$, $w \rightarrow_T z$ if and only if $w \leftrightarrow_T z$ and $|w| > |z|$. A word $w \in \Sigma^*$ is *irreducible* for T (or T*-irreducible*) if there is no $z \in \Sigma^*$ such that $w \rightarrow_T z$. The set of all irreducible strings for T is denoted by $IRR(T)$. We say that T is *Church-Rosser* if for all $w, z \in \Sigma^*$, if $w \leftrightarrow_T^* z$, then there exists an $x \in \Sigma^*$ such that $w \rightarrow_T^* x$ and $z \rightarrow_T^* x$.

A *string rewriting system* S (over Σ) is a "one-way" version of a Thue system in that its finite set of rewriting rules can be used in one direction only. The relation $\rightarrow_S^*$ is the reflexive, transitive closure of the relation $\rightarrow_S$ defined by: for $w, z \in \Sigma^*$, $w \rightarrow_S z$ if there exist $x, y \in \Sigma^*$ and $(u, v) \in S$ such that $w = xuy$ and $z = xvy$. We say that z can be *derived* from w in S, if $w \rightarrow_S^* z$ holds. The symmetric, reflexive and transitive closure $\leftrightarrow_S^*$ of $\rightarrow_S$ is a congruence over Σ^*. It is called the *Thue congruence generated* by S. We say that

(a) S is *noetherian* if there are no infinite chains of the form $w_1 \rightarrow_S w_2 \rightarrow_S \ldots$,
(b) S is *Church-Rosser* if for every $w, z \in \Sigma^*$, $w \leftrightarrow_S^* z$ implies that $w \rightarrow_S^* x$ and $z \rightarrow_S^* x$ for some $x \in \Sigma^*$.

A word w is called *irreducible with respect to* S (or S*-irreducible*) if there is no z such that $w \rightarrow_S z$. The set of all S-irreducible words is denoted by $IRR(S)$.

We now mention a sufficient condition for S to be noetherian. A *weight function* is a mapping $\rho : \Sigma \rightarrow \{1, 2, \ldots\}$, where for $a \in \Sigma$, $\rho(a)$ is the *weight* of a. It can be extended to a mapping $\rho : \Sigma^* \rightarrow \{1, 2, \ldots\}$ by letting $\rho(\lambda) = 0$ and, inductively, defining $\rho(wa) = \rho(w) + \rho(a)$ for any $w \in \Sigma^*$ and $a \in \Sigma$. For example, if $\rho(a) = 1$ for each $a \in \Sigma$, then $\rho(w) = |w|$. We say that S is *weight reducing with respect* to ρ if, for each $(u, v) \in S$, $\rho(u) > \rho(v)$ holds. S is *weight reducing* if there is a weight function with respect to which S is weight reducing. It should be clear that each weight reducing string rewriting system is noetherian. The following theorem gives a necessary and sufficient condition for the Church-Rosser property.

Theorem 2.5 ([14], [2]) A Thue system T (noetherian string rewriting system S) is Church-Rosser if and only if each class of the congruence $\leftrightarrow_T^*$ ($\leftrightarrow_S^*$) contains exactly one T-irreducible (S-irreducible) element.

3 The Outline of the Method

In this section we define precisely the problem we propose to solve and outline an approach to its solution. The remainder of the paper will, in essence, implement the methodology outlined here. Let M be a finite set of tree transformation classes. We consider two monoids defined in terms of M: the *free monoid* M^* (with the operation of concatenation denoted by "$\cdot$") and $[M]$, the monoid finitely generated by M (with the operation of composition denoted by "$\circ$"). Strings over M represent transformation classes in $[M]$ by means of a homomorphism $|| \, || : M^* \to [M]$ defined by

$$||Y_1 \cdot Y_2 \cdot \ldots \cdot Y_m|| \; = \; Y_1 \circ Y_2 \circ \ldots \circ Y_m \quad .$$

We denote by $I \in [M]$ the tree transformation class consisting of all identity tree transformations, i.e., $I = ||\lambda||$. Let θ be the *kernel* of $|| \, ||$, i.e., the *congruence* relation induced by the homomorphism $|| \, ||$:

$$\theta = ker(|| \, ||) = \{(v, w) \in M^* \times M^* \mid \; ||v|| \; = \; ||w|| \} \quad .$$

Let N be a set of representatives of the congruence classes of θ. The elements of N are called *normal forms* with respect to θ. Fülöp and Vágvölgyi [12] posed the problem of constructing an algorithm which, given $Y_1, \ldots, Y_m, Z_1, \ldots, Z_n \in M$, decides which one of the following four mutually exclusive conditions holds:

(i) $Y_1 \circ \ldots \circ Y_m \; = \; Z_1 \circ \ldots \circ Z_n$, (ii) $Y_1 \circ \ldots \circ Y_m \; \subset \; Z_1 \circ \ldots \circ Z_n$,
(iii) $Z_1 \circ \ldots \circ Z_n \; \subset \; Y_1 \circ \ldots \circ Y_m$, (iv) $Y_1 \circ \ldots \circ Y_m \; \bowtie \; Z_1 \circ \ldots \circ Z_n$,

where "$\bowtie$" stands for the incomparability relationship. In [12] they suggested a *methodology*, by which such an algorithm can be constructed provided M is not "too general". Specifically, they suggested the following approach.

(a) Give a set of representatives N for the congruence classes of θ.
(b) Give the inclusion diagram of the set $||N|| = \{ \, ||u|| \mid u \in N \, \}$, i.e., of the set of tree transformation classes represented by normal forms. Note that $||N|| = [M]$. Also note that having this inclusion diagram, for any given $u, v \in N$, we can read from can read from the diagram, which one of the following conditions holds:

(i′) $||u|| \; \subset \; ||v||$ (ii′) $||v|| \; \subset \; ||u||$
(iii′) $||u|| \; = \; ||v||$ (iv′) $||u|| \; \bowtie \; ||v||$

(c) Give a finite set $T \subseteq M^* \times M^*$ of generators of θ (i.e., a Thue system T over M such that $\leftrightarrow_T^* = \theta$) and give an algorithm that for any $w \in M^*$, by a suitable sequence of substitutions induced by T, computes the normal form of w, i.e., the unique $u \in N$ for which $w \leftrightarrow_T^* u$.

Now we prove that once tasks (a)-(c) are accomplished, we have an algorithm that decides, given $Y_1, \ldots, Y_m, Z_1, \ldots, Z_n \in M$, which one of the conditions (i)-(iv) holds. First, by the algorithm in (c), compute the normal forms $u, v \in N$ such that

$$Y_1 \cdot \ldots \cdot Y_m \overset{*}{\underset{T}{\leftrightarrow}} u \ \text{ and } \ Z_1 \cdot \ldots \cdot Z_n \overset{*}{\underset{T}{\leftrightarrow}} v \ .$$

Since, by (c), we have $\leftrightarrow_T^* = \theta$, we also have

$$Y_1 \circ \ldots \circ Y_m = ||u|| \ \text{ and } \ Z_1 \circ \ldots \circ Z_n = ||v||.$$

Thus, one of the conditions (i)-(iv) holds for $Y_1 \circ \ldots \circ Y_m$ and $Z_1 \circ \ldots \circ Z_n$ if and only if the corresponding condition of (i′)-(iv′) holds for $||u||$ and $||v||$. Moreover, having the inclusion diagram, by (b), we can read from the diagram which one of the conditions (i′)-(iv′) holds. Hence we obtained the following.

Theorem 3.1 ([12]) Suppose that the tasks (a)-(c) have been executed for M. Then, there is an algorithm which decides, given any tree transformation classes $Y_1, \ldots, Y_m, Z_1, \ldots, Z_n \in M$, which one of the conditions (i)-(iv) holds.

When applying the above general method to concrete choices of M, it transpired that it is useful to implement the tasks (a)-(c) by performing the following five steps.

(I) Give a finite relation $T \subseteq M^* \times M^*$ which is our candidate for the set of generators of θ. (Here we are advised to take into consideration known decomposition results that hold among elements of M.)

(II) Prove that for every $(u, v) \in T$, $||u|| = ||v||$. (Otherwise, T cannot be a set of generators of θ. In this way elements of T represent equalities over $[M]$.) We note that (II) implies the inclusion $\leftrightarrow_T^* \subseteq \theta$ because, by (II), $T \subseteq \theta$ and $\leftrightarrow_T^*$ is the smallest congruence on M^* containing T.

(III) Give a subset N of M^* which is a candidate for a set of representatives for the congruence classes of θ.

(IV) Give the inclusion diagram of the set $||N|| = \{ \, ||u|| \mid u \in N \}$, which is, in fact, the set of tree transformation classes represented by the elements of N. By using this inclusion diagram show that for any $u, v \in N$, if $u \neq v$, then $||u|| \neq ||v||$.

(V) Give an algorithm that for every $w \in M^*$ computes a $u \in N$ such that $w \leftrightarrow_T^* u$.

Next we prove that once we have successfully implemented steps (I)-(V), we have also accomplished the tasks (a)-(c).

Lemma 3.2 ([12]) Suppose that we have carried out the steps (I)-(V). Then the tasks (a)-(c) have also been accomplished.

Proof. We first show that $\theta = \leftrightarrow_T^*$. By (I) and (II) we have $\leftrightarrow_T^* \subseteq \theta$. For the other direction let $w, w' \in M^*$ be such that $w \theta w'$. Then construct, by (V), the normal forms u

and u' for which $w \leftrightarrow_T^* u$ and $w' \leftrightarrow_T^* u'$. Since $\leftrightarrow_T^* \subseteq \theta$, we also have $w\theta u$ and $w'\theta u'$ from which $u\theta u'$ follows. Then, by (III), $u = u'$ and thus $w \leftrightarrow_T^* u = u' \leftrightarrow_T^* w'$. Hence $\theta \subseteq \leftrightarrow_T^*$. Consequently we obtain $\leftrightarrow_T^* = \theta$, which together with (V) yields (c). Moreover, by (III) and (V), it follows that N is indeed a set of representatives of θ, hence we have (a). Finally, by (IV), we have (b).

$\square$

Remark Theorem 3.1 and Lemma 3.2 were proved in [12]. Because the proofs are short (and relevant) we have reproduced them for the sake of completeness.

Our aim is to apply the (I)-(V) method to the set

$$M = \{\, DTA, DTT, DTT^{DR}, DTT^R \,\},$$

where DTA, DTT, DTT^{DR}, DTT^R stand for the classes of all tree transformations defined in Section 2.1. Our choice of M is motivated by the known composition and decomposition results and by the interesting hierarchy results we obtain in Section 5. We shall follow the approach of Fülöp and Vágvölgyi [12] and perform the the steps (I)-(V) for the monoid $[M]$ induced by M.

4 The Thue System T_M

Consider the set of tree transformations $M = \{\, DTA, DTT, DTT^{DR}, DTT^R \,\}$. We first define a finite Thue system $T_M \subseteq M^* \times M^*$ whose Thue congruence is equal to θ, the kernel of the homomorphism $|| \; || : M^* \to [M]$ defined in Section 3. T_M consists of the following 13 rewriting rules.

(1) $(DTA \cdot DTT^R, DTT^R)$

(2) $(DTT^R \cdot DTA, DTT^R)$

(3) $(DTT \cdot DTT^R, DTT^R)$

(4) $(DTT^R \cdot DTT, DTT^R)$

(5) $(DTT^{DR} \cdot DTT^R, DTT^R)$

(6) $(DTT^R \cdot DTT^{DR}, DTT^R)$

(7) $(DTT^R \cdot DTT^R, DTT^R)$

(8) $(DTA \cdot DTT^{DR}, DTT^{DR})$

(9) $(DTT^{DR} \cdot DTA, DTT^{DR})$

(10) $(DTT \cdot DTT^{DR}, DTT^{DR})$

(11) $(DTT \cdot DTT, DTA \cdot DTT)$

(12) $(DTT \cdot DTA, DTT)$

(13) $(DTA \cdot DTA, DTA)$

Next we will argue that for every $(\alpha, \beta) \in T_M$, $||\alpha|| = ||\beta||$, or equivalently, $(\alpha, \beta) \in \theta$. This will establish parts (I) and (II) of our method. For each i $(1 \leq i \leq 13)$, if the i-th rewriting rule of T_M is (α, β), then the corresponding claim $||\alpha|| = ||\beta||$ will be denoted by (i'). We thus have to prove that (i') holds for $1 \leq i \leq 13$. Almost all these claims are well-known results which we summarize in the following lemma.

Lemma 4.1

(a) [5] $DTT^R \circ DTT^R = DTT^R$. This establishes (7′).

(b) $DTA \circ DTT^R = DTT^R \circ DTA = DTT \circ DTT^R = DTT^R \circ DTT$
$= DTT^{DR} \circ DTT^R = DTT^R \circ DTT^{DR} = DTT^R$.

This follows from (a) and establishes (1′), (2′), (3′), (4′), (5′), and (6′).

(c) $DTA \circ DTT^{DR} = DTT^{DR}$. This follows from (10′) and establishes (8′).

(d) $DTT^{DR} \circ DTA = DTT^{DR}$. This follows by an easy construction and establishes (9′).

(e) [9] $DTT \circ DTT = DTA \circ DTT$. This establishes (11′).

(f) [12] $DTT \circ DTA = DTT$. This establishes (12′).

(g) [Folklore] $DTA \circ DTA = DTA$. This establishes (13′).

□

It remains to prove (10′). We will need the following concept. Let $\mathcal{A} = (\Sigma, \Delta, A, a_0, P)$ be a dtt$^{\mathrm{DR}}$ and $p \in \bar{T}_\Sigma(X_n)$. We introduce the relation $\mapsto_{\mathcal{A},p} \subseteq \Rightarrow^*_{\mathcal{A}}$. Intuitively, the notation $a(p[p_1, \ldots, p_n]) \mapsto_{\mathcal{A},p} r[a_1(p_{i_1}), \ldots, a_m(p_{i_m})]$ means that $r[a_1(p_{i_1}), \ldots, a_m(p_{i_m})]$ is the tree resulting from the partial computation of $\mathcal{A}$ on $p[p_1, \ldots, p_n]$ starting in state a and down to the leaves of p without entering any of the subtrees p_i. More formally,

(i) if $p = x_1$, then $p[p_1, \ldots, p_n] = p_1$ and $a(p_1) \mapsto_{\mathcal{A},p} a(p_1)$;

(ii) let $p = \sigma(t_1, \ldots, t_m)$, $\sigma \in \Sigma_m$, $m \geq 0$,

$$a(\sigma(t_1, \ldots, t_m)[p_1, \ldots, p_n]) \Rightarrow_{\mathcal{A}} (q[a_1(t_{i_1}), \ldots, a_k(t_{i_k})])[p_1, \ldots, p_n]$$

for some $q \in \bar{T}_\Delta(X_k)$, $k \geq 0$, and $a_j(t_{i_j}[p_1, \ldots, p_n]) \mapsto_{\mathcal{A},t'_{i_j}} r_{i_j}$ for $1 \leq j \leq k$, where for each $t_{i_j} \in T_\Sigma(X_{l_j})$, $t'_{i_j} \in \bar{T}(X_{l_j})$ is t_{i_j} with its variables reindexed so that their successive occurrences form left-to right are $x_1, x_2, \ldots, x_{l_j}$. Then

$$a(p[p_1, \ldots, p_n]) \mapsto_{\mathcal{A},p} (q[r_{i_1}, \ldots, r_{i_k}])[p_1, \ldots, p_n].$$

Lemma 4.2 Let $\mathcal{B} = (\Delta, \Gamma, B, b_0, P)$ be a dtt^{DR} and $g \geq 0$ an integer. Then for any $q \in \bar{T}_\Delta(X_g)$ and $b \in B$, there exists an integer $k \geq 0$, and for each $1 \leq i \leq k$, there exists a tree $r^{(i)} \in T_\Gamma(B(X_g))$ and tree languages $L_1^{(i)}, \ldots, L_g^{(i)} \in DR$ ($L_1^{(i)}, \ldots, L_g^{(i)} \subseteq T_\Delta$) such that the following conditions hold.

(a) For every $1 \leq i < j \leq k$, there exists $1 \leq l \leq g$ with $L_l^{(i)} \cap L_l^{(j)} = \emptyset$.

(b) For all $q_1, \ldots, q_g \in T_\Delta$ and $1 \leq i \leq k$, if $q_1 \in L_1^{(i)}, \ldots, q_g \in L_g^{(i)}$, then

$$b(q[q_1, \ldots, q_g]) \mapsto_{\mathcal{B},q} r^{(i)}[q_1, \ldots, q_g].$$

(c) For all $q_1, \ldots, q_g \in T_\Delta$, $r \in T_\Gamma(B(X_g))$, if

$$b(q[q_1, \ldots, q_g]) \mapsto_{\mathcal{B},q} r[q_1, \ldots, q_g],$$

then there exists $1 \leq i \leq k$ such that $r = r^{(i)}$, and $q_1 \in L_1^{(i)}, \ldots, q_g \in L_g^{(i)}$.

Proof. For $b \in B$, $q \in \bar{T}_\Delta(X_g)$, $r \in T_\Gamma(B(X_g))$, and $L_1, \ldots, L_g \in DR$, a construct

$$< b(q) \to r; L_1, \ldots, L_g > \qquad (*)$$

is called an *extended rule of* $\mathcal{B}$ *for* b *and* q. While the "rule part" $b(q) \to r$ of $(*)$ is intended to represent the computation of $\mathcal{B}$ down to the leaves of q, the languages $L_1, \ldots, L_g$ represent the "cumulative look-ahead" conditions on the trees that can be substituted for $x_1, \ldots, x_g$ in order to enable $\mathcal{B}$ to reach the leaves labeled $x_1, \ldots, x_g$ in q. For each node of q we will construct a set of extended rules of $\mathcal{B}$ so that conditions (a)-(c) hold. The proof proceeds by induction on the structure of $q \in \bar{T}_\Delta(X_g)$.

If $q \in X_g$ then $q = x_1$ and we let $k = 1$, $r^{(1)} = b(q) = b(x_1)$, and $L_1^{(1)} = T_\Delta$. The set of extended rules, in this case, has just the (trivial) rule : $< b(x_1) \to b(x_1);\ T_\Delta >$. Condition (a) holds trivially and condition (c) follows from the definition of $\mapsto_{\mathcal{B},x_1}$. Let $q_1, \ldots, q_g \in T_\Delta$. Then

$$\begin{aligned} b(q[q_1, \ldots, q_g]) = b(q_1) \mapsto_{\mathcal{B},q} & \ b(q_1) \\ = & \ b(q[q_1, \ldots, q_g]) \\ = & \ b(q)[q_1, \ldots, q_g] \\ = & \ r^{(1)}[q_1, \ldots, q_g] \ , \end{aligned}$$

establishing (b). Now suppose $q \in \Delta_0$. If $\mathcal{B}$ has no rule with left-hand side $b(q)$, then let $k = 0$ and (a)-(c) hold trivially. Otherwise, let $< b(q) \to u; >$ be a rule in P (note that the list of look-ahead languages is empty in this case). Since $\mathcal{B}$ is deterministic, it has no other rules with $b(q)$ as a left-hand side. Define $k = 1$ and $r^{(1)} = u$. The set of extended rules of $\mathcal{B}$ for b and q consists, in this case, of a single rule $< b(q) \to u; >$. Conditions (a)-(c) hold trivially.

For the inductive step suppose $q = \sigma(t_1, \ldots, t_m) \in \bar{T}_\Delta(X_g)$ for some $\sigma \in \Delta_m$, $m \geq 1$. Suppose that t_i has g_i occurrences of variables; thus $g = g_1 + g_2 + \ldots + g_m$. For each t_i, let $t'_i \in \bar{T}_\Delta(X_{g_i})$ be t_i with its variables reindexed so that their successive occurrences from left to right are $x_1, x_2, \ldots, x_{g_i}$. Let $1 \leq h \leq g$. The variable x_h in q is reindexed into $x_{s(h)}$ so that if x_h occurs in t_j then $x_{s(h)}$ occurs in t'_j and $1 \leq s(h) \leq g_j$. Let $j(h)$ be the j such that x_h occurs in t_j.

Now, if $\mathcal{B}$ has no rule with left-hand side $b(\sigma(x_1, \ldots, x_m))$ then let $k = 0$ and conditions (a)-(c) are satisfied trivially. Otherwise, consider a rule of P:

$$< b(\sigma(x_1, \ldots, x_m)) \to u[b_1(x_{i_1}), \ldots, b_l(x_{i_l})]; L_1, \ldots, L_m > \qquad (**)$$

where $u \in \bar{T}_\Gamma(X_l)$, and $b_1, \ldots, b_l \in B$. We now explain how to construct the set of extended rules of $\mathcal{B}$ for b and q, that are associated with the rule $(**)$. Taking a union of all these sets of extended rules, for various rules $(**)$ for $b(q(x_1, \ldots, x_m))$, gives the required set of extended rules of $\mathcal{B}$ for b and q. The cardinality of this set is the required k. Let the dta $\mathcal{A}_n =< \Delta, \Delta, A_n, a_0^n, P_n >$, $1 \leq n \leq m$, recognize the look-ahead language L_n from $(**)$. Moreover, suppose that for $1 \leq n \leq m$, and arbitrary trees $p_1, \ldots, p_{g_n} \in T_\Sigma$,

$$a_0^n(t'_n[p_1,\ldots,p_n]) \underset{\mathcal{A}_n}{\overset{*}{\Rightarrow}} (t'_n[a_1^n(x_1),\ldots,a_{g_n}^n(x_{g_n})])[p_1,\ldots,p_{g_n}].$$

Now, by induction hypothesis, for each $1 \le j \le l$, pick an extended rule of $\mathcal{B}$ for b_j and t'_{i_j}:

$$< b_j(t'_{i_j}) \to u'_j;\ M_1^{(j,i_j)}, M_2^{(j,i_j)}, \ldots, M_{g_{i_j}}^{(j,i_j)} > \ ,$$

and let u_j be defined from u'_j by restoring the original indexing of variables (i.e., replacing index $s(h)$ by h). Define a corresponding extended rule of $\mathcal{B}$ for b and q

$$< b(q) \to u[u_1,\ldots,u_l];\ N_1, N_2, \ldots, N_g >$$

where for $1 \le h \le g$,

$$N_h = (\bigcap_{i_j = j(h)} M_{s(h)}^{(j,i_j)}) \cap L(\mathcal{A}_{j(h)}(a_{s(h)}^{j(h)}))$$

where $\mathcal{A}_n(a_{s(h)}^n)$ $(1 \le n \le m)$ is the dta $\mathcal{A}_n$ with initial state $a_{s(h)}^n$ instead of a_0^n.

It is now easy to check that condition (a) holds by induction hypothesis and because $\mathcal{B}$ is deterministic. Similarly, condition (c) holds by induction hypothesis and because we consider all the rules $(**)$ for $b(\sigma(x_1,\ldots,x_m))$. It is also easy see that if $q_1 \in N_1, \ldots, q_g \in N_g$ then

$$b(q[q_1,\ldots,q_g]) \mapsto_{\mathcal{B},q} r[q_1,\ldots,q_g]$$

where $r = u[u_1,\ldots,u_l]$; thus condition (b) holds.

$\square$

Lemma 4.3 $DTT \circ DTT^{DR} = DTT^{DR}$.

Proof. Let $\mathcal{A} = (\Sigma, \Delta, A, a_0, P_1)$ be a dtt and $\mathcal{B} = (\Delta, \Gamma, B, b_0, P_2)$ be a dtt^{DR}. We construct the dtt^R $\mathcal{C} = (\Sigma, \Gamma, A \times B, (a_0, b_0), P_3)$ as follows. The rule

$$< (a,b)(\sigma(x_1,\ldots,x_m)) \to r[(a_{\psi(1)}, b_1)(x_{\rho(1)}), \ldots, (a_{\psi(g)}, b_g)(x_{\rho(g)})];\ J_1, \ldots, J_m >$$

is in P_3 where $\rho : \{1,\ldots,g\} \to \{1,\ldots,m\}$ if the following conditions hold.

(a) $a(\sigma(x_1,\ldots,x_m)) \to q[a_1(x_{\phi(1)}),\ldots,a_n(x_{\phi(n)})] \in P_1$, where $q \in \bar{T}_\Delta(X_n)$, $n \ge 0$, $\phi : \{1,\ldots,n\} \to \{1,\ldots,m\}$.

(b) $< b(q) \to r[b_1(x_{\psi(1)}),\ldots,b_g(x_{\psi(g)})]; L_1,\ldots,L_n >$ is an extended rule of $\mathcal{B}$ for b and q, where $g \ge 0$, $r \in \bar{T}_\Gamma(X_g)$, and $\psi : \{1,\ldots,g\} \to \{1,\ldots,n\}$.

(c) For each $1 \le j \le g$, $\rho(j) = \phi(\psi(j))$.

(d) For each $1 \le j \le m$, $J_j = \bigcap\{dom(\tau_{\mathcal{A}(a_i)} \circ ID(L_i)) \mid \phi(i) = j, 1 \le i \le n\}$ if there exists an $1 \le i \le n$, such that $\phi(i) = j$, and $J_j = T_\Sigma$ otherwise.

By part (f) of Lemma 4.1, for each $1 \le i \le n$, $\tau_{\mathcal{A}(a_i)} \circ ID(L_i) \in DTT$. Thus, by Theorem 3.1 in [5], for each $1 \le i \le n$, $\tau_{\mathcal{A}(a_i)} \circ ID(L_i) \in DR$. As DR is closed under composition, see (g) of Lemma 4.1, for each $1 \leq j \leq m$, $J_j \in DR$. Hence $\mathcal{C}$ is a dtt^{DR}. In order to show that $\tau_{\mathcal{C}} = \tau_{\mathcal{A}} \circ \tau_{\mathcal{B}}$, it is enough to prove that for an arbitrary state (a, b) of $\mathcal{C}$ and trees $p \in T_\Sigma$, $r \in T_\Gamma$, the equivalence

$$(a, b)(p) \overset{*}{\underset{\mathcal{C}}{\Rightarrow}} r \text{ if and only if } (\exists q \in T_\Delta)(a(p) \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} q \text{ and } b(q) \overset{*}{\underset{\mathcal{B}}{\Rightarrow}} r).$$

holds. This can be done by induction on the structure of p.

□

We summarize the results of this section in the following theorem which is a simple consequence of Lemma 4.1 and Lemma 4.3.

Theorem 4.4 For every $(\alpha, \beta) \in T_M$, $||\alpha|| = ||\beta||$, or equivalently, $(\alpha, \beta) \in \theta$.

5 The Inclusion Diagram

In this section we continue to implement the methodology outlined in Section 3, by executing steps (III) and (IV). For (III) we have to give a subset $N \subseteq M^*$, our candidate for the set of representatives of the congruence classes of θ. Here is our candidate:

$$\begin{aligned} N = \{\, & I, DTA, DTT, DTA \cdot DTT, DTT^R \,\} \cup \\ & \{\, (DTT^{DR})^n \mid n \geq 1 \,\} \cup \\ & \{\, (DTT^{DR})^n \cdot DTT \mid n \geq 1 \,\} \ . \end{aligned}$$

According to (IV), we have to give an inclusion diagram for the set $||N|| = \{\, ||w|| \mid w \in N \,\}$. Indeed we will show that the elements of $||N||$ can be arranged into a proper hierarchy, all inside DTT^R. This hierarchy result is the main technical contribution of this paper. It is displayed in Figure 1.

The properness of inclusion for the initial levels of the hierarchy is trivial:

$$I \subset DTA \subset DTT \tag{1}$$

and Rounds [15] and Fülöp and Vágvölgyi [9], [10], have shown

$$DTT \subset DTT \circ DTT = DTA \circ DTT \subset DTT^{DR} \ . \tag{2}$$

To establish the hierarchy result it suffices to prove the two proper inclusions in

$$(DTT^{DR})^n \subset (DTT^{DR})^n \circ DTT \subset (DTT^{DR})^{n+1} \tag{3}$$

for all $n \geq 1$, and

$$\bigcup_{n=0}^{\infty} (DTT^{DR})^n \subset DTT^R \ . \tag{4}$$

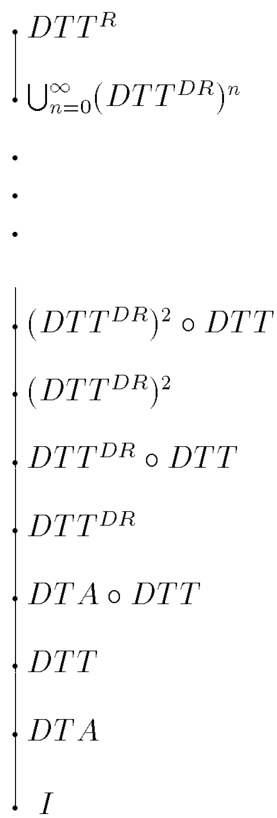

Fig. 1. The inclusion diagram of the monoid $[M]$.

Note that the union on the left-hand side of (4) is not an element of $||N||$. The inclusions in (3) and (4) are obvious so we will concentrate on the properness issue. We will use the classes of tree languages DR_n, $n \geq 0$, defined in Section 2.1, and the languages of n-nested combs, see Example 2.2 and Proposition 2.4 of Section 2.1.

We first discuss and prove some results about the domains of deterministic top-down tree transducers with DR_n look-ahead. An early result (without look-ahead) $dom(DTT) = DR$ is proved in [5]. This was extended in [10]: $dom(DTT^{DR}) \subseteq DR_1$. We generalize these results in the following lemma.

Lemma 5.1 For every $n \geq 0$, $dom(DTT^{DR_n}) \subseteq DR_{n+1}$.

Proof. We proceed by an induction on n. For $n = 0$ the result follows from [10]. Suppose that the result holds for integers smaller than n. We apply the usual power set construction. Let $\mathcal{A} =< \Sigma, \Delta, A, a_0, P >$ be a dtt^{DR_n}. Define the dta^{DR_n} $\mathcal{B} =< \Sigma, \Sigma, B, b_0, P' >$ where $B = P(A)$, the power set of A, $b_0 = \{ a_0 \}$, and P' is the set of all rules of the form

$$< b(\sigma(x_1, \ldots, x_m)) \to \sigma(b_1(x_1), \ldots, b_m(x_m)); K_1, \ldots, K_m >$$

constructed in the following way:

(i) Let $b \in P(A)$, $m \geq 0$, and $\sigma \in \Sigma_m$ be such that for each $a \in b$ there exists at least one rule with left-hand side $a(\sigma(x_1, \ldots, x_m))$ in P.
(ii) For each $a \in b$ choose a rule $< a(\sigma(x_1, \ldots, x_m)) \rightarrow t^a; L_1^a, \ldots, L_m^a >$ from P. For each $1 \leq i \leq m$, define $b_i = \{ a' \in A \mid a'(x_i) \text{ occurs in } t^a \}$ and $K_i = \bigcap_{a \in b} L_i^a$.

The following fact can be proved by induction on the structure of trees: for each $t \in T_\Sigma$, and $b \in B$, $b(t) \Rightarrow_{\mathcal{B}}^* t$ if and only if for each $a \in b$ there exists $r \in T_\Delta$ with $a(t) \Rightarrow_{\mathcal{A}}^* r$. Therefore, $\tau_{\mathcal{B}} = \{ (t,t) \mid t \in dom(\tau_{\mathcal{A}}) \}$ and so $dom(\tau_{\mathcal{A}}) = dom(\tau_{\mathcal{B}})$.

□

Next we look at the relationship between DTT^{DR_n} and the composition classes $(DTT^{DR})^n$. The proof of the first lemma is straightforward and we omit it.

Lemma 5.2 For every $n \geq 0$, $DTT^{DR_n} \circ DTA = DTT^{DR_n}$.

Lemma 5.3 For every $n \geq 1$, $DTT^{DR_{n-1}} \circ DTT^{DR} \subseteq DTT^{DR_n}$.

Proof. Let $\mathcal{A} = (\Sigma, \Delta, A, a_0, P_1)$ be a dtt$^{DR_{n-1}}$ and $\mathcal{B} = (\Delta, \Gamma, B, b_0, P_2)$ be a dtt^{DR}. We construct the dtt^R $\mathcal{C} = (\Sigma, \Gamma, A \times B, (a_0, b_0), P_3)$ as follows. The rule

$$< (a,b)(\sigma(x_1, \ldots, x_m)) \rightarrow r[(a_1,b_1)(x_{\rho(1)}), \ldots, (a_g,b_g)(x_{\rho(g)})]; M_1, \ldots, M_m >$$

with $r \in T_\Delta(X_g)$, $g \geq 0$, and $\rho : \{1, \ldots, g\} \rightarrow \{1, \ldots, m\}$, is in P_3 if the following conditions hold.

(a) $< a(\sigma(x_1, \ldots, x_m)) \rightarrow q[a_1(x_{\phi(1)}), \ldots, a_n(x_{\phi(n)})]; K_1, \ldots, K_m > \in P_1$, where $q \in \bar{T}_\Delta(X_n)$, $\phi : \{1, \ldots, n\} \rightarrow \{1, \ldots, m\}$.
(b) $< b(q) \rightarrow r[b_1(x_{\psi(1)}), \ldots, b_g(x_{\psi(g)})]; L_1, \ldots, L_n >$ is a generalized rule of $\mathcal{B}$, where $\psi : \{1, \ldots, g\} \rightarrow \{1, \ldots, n\}$.
(c) For each j, $1 \leq j \leq g$, $\rho(j) = \phi(\psi(j))$.
(d) For each $1 \leq j \leq m$, $M_j = K_j \cap (\bigcap \{ dom(\tau_{\mathcal{A}(a_i)} \circ ID(L_i)) \mid \phi(i) = j, 1 \leq i \leq g \})$ if there exists an i, $1 \leq i \leq g$, such that $\phi(i) = j$, and $M_j = K_j$ otherwise.

By Lemma 5.2, for each $1 \leq i \leq g$, $\tau_{\mathcal{A}(a_i)} \circ ID(L_i) \in DTT^{DR_{n-1}}$ and by Lemma 5.1, the domain of a dtt$^{DR_{n-1}}$ is in DR_n. Moreover, $DR_{n-1} \subset DR_n$ and DR_n is closed under intersection. Hence $M_j \in DR_n$ for $1 \leq j \leq m$. Thus $\mathcal{C}$ is a dtt^{DR_n}.

In order to show that $\tau_{\mathcal{C}} = \tau_{\mathcal{A}} \circ \tau_{\mathcal{B}}$, it is enough to prove that for arbitrary state (a,b) of $\mathcal{C}$ and trees $p \in T_\Sigma$, $r \in T_\Gamma$, the equivalence

$$(a,b)(p) \overset{*}{\underset{\mathcal{C}}{\Rightarrow}} r \text{ if and only if } (\exists q \in T_\Delta)(a(p) \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} q \text{ and } b(q) \overset{*}{\underset{\mathcal{B}}{\Rightarrow}} r).$$

holds. This can be done by induction on the structure of p.

□

Lemma 5.4 For every $n \geq 1$, $(DTT^{DR})^n \subseteq DTT^{DR_{n-1}}$.

Proof. By induction on n. For $n = 1$ the statement is trivial. Let us assume that $n \geq 2$, and $(DTT^{DR})^{n-1} \subseteq DTT^{DR_{n-2}}$. Then

$$\begin{aligned}
(DTT^{DR})^n &= (DTT^{DR})^{n-1} \circ DTT^{DR} & \\
&\subseteq DTT^{DR_{n-2}} \circ DTT^{DR} & \text{by induction hypothesis} \\
&\subseteq DTT^{DR_{n-1}} & \text{by Lemma 5.3.}
\end{aligned}$$

□

An immediate consequence of Lemma 5.1 and Lemma 5.4 is the following corollary.

Corollary 5.5 For every $n \geq 0$, $dom((DTT^{DR})^n) \subseteq DR_n$.

To set up the proper inclusion results we will need the following key lemma about C_n^e, the language of n-nested combs with even number of 1's.

Lemma 5.6 For every $n \geq 1, C_n^e \in dom((DTT^{DR})^n)$.

Proof. It suffices to show that $\{ (t, 0) \mid t \in C_n^e \} \in (DTT^{DR})^n$. We prove this by induction on n. Let $n = 1$. Recall that $C_1 = \{ e_m(t_1, \ldots, t_m, y) \mid m \geq 0, \text{ and } t_1, \ldots, t_m, y \in \{ 1, 0 \} \}$. Define the dtt^{DR} $\mathcal{A} =< \Sigma, \Sigma, A, a_e, P >$, where

(a) $\Sigma = \Sigma_0 \cup \Sigma_2$, $\Sigma_0 = \{ 1, 0 \}$, $\Sigma_2 = \{ \sigma \}$, and $A = \{ a_e, a_o \}$

(b) P consists of the following rules:

$< a_e(\sigma(x_1, x_2)) \rightarrow a_o(x_2); \{ 1 \}, T_\Sigma >$, $\quad < a_e(\sigma(x_1, x_2)) \rightarrow a_e(x_2); \{ 0 \}, T_\Sigma >$,
$< a_o(\sigma(x_1, x_2)) \rightarrow a_e(x_2); \{ 1 \}, T_\Sigma >$, $\quad < a_o(\sigma(x_1, x_2)) \rightarrow a_o(x_2); \{ 0 \}, T_\Sigma >$,
$< a_o(1) \rightarrow 0; >$, $\quad < a_e(0) \rightarrow 0; >$.

Intuitively, the transducer $\mathcal{A}$ comes down the "spine" of the input tree (which should be a comb) and checks, with its look-ahead, the left child of the current node. Moreover, $\mathcal{A}$ memorizes in its state the parity of the number of 1's encountered so far. $\mathcal{A}$ is in state a_e (respectively, a_o) if the number of the already read 1's is even (odd). Finally, when $\mathcal{A}$ arrives at the nullary symbol occurring at the end of the spine, $\mathcal{A}$ finds the parity of the total number of occurrences of 1 in the input tree. If there are even number of 1's in the input tree, then $\mathcal{A}$ outputs 0, otherwise $\mathcal{A}$ halts without output. It is easy to see that $\tau_{\mathcal{A}} = \{ (t, 0) \mid t \in C_1^e \}$.

Suppose that $n \geq 2$, and that the claim holds for $n-1$. Recall that $C_n = \{ e_m(t_1, \ldots, t_m, y) \mid m \geq 0, y \in \{ 1, 0 \} \text{ and } t_1, \ldots, t_m \in C_{n-1} \}$. Define the dtt^{DR} $\mathcal{A} =< \Sigma, \Sigma, A, a_1, P >$, where

(a) $\Sigma = \Sigma_0 \cup \Sigma_2$, $\Sigma_0 = \{ 1, 0 \}$, $\Sigma_2 = \{ \sigma \}$, and $A = \{ a_1, a_2, \ldots, a_{n-1}, a_e, a_o \}$,

(b) P consists of the following rules: for each $1 \leq i \leq n-2$,
$< a_i(\sigma(x_1, x_2)) \rightarrow \sigma(a_{i+1}(x_1), a_i(x_2)); T_\Sigma, T_\Sigma >$,

$< a_{n-1}(\sigma(x_1, x_2)) \to \sigma(a_e(x_1), a_{n-1}(x_2)); T_\Sigma, T_\Sigma >$,
$< a_i(1) \to 1; >$, $< a_i(0) \to 0; >$, $< a_{n-1}(1) \to 1; >$, $< a_{n-1}(0) \to 0; >$,
$< a_e(\sigma(x_1, x_2)) \to a_o(x_2); \{1\}, T_\Sigma >$, $< a_e(\sigma(x_1, x_2)) \to a_e(x_2); \{0\}, T_\Sigma >$,
$< a_o(\sigma(x_1, x_2)) \to a_e(x_2); \{1\}, T_\Sigma >$, $< a_o(\sigma(x_1, x_2)) \to a_o(x_2); \{0\}, T_\Sigma >$,
$< a_o(1) \to 0; >$, $< a_o(0) \to 1; >$, $< a_e(1) \to 1; >$, $< a_e(0) \to 0; >$.

Roughly, $\mathcal{A}$ trims the "outermost" 1-combs off an input tree $t \in C_n$ and replaces them with a 0 or 1 depending on the parity of the number of 1's. $\mathcal{A}$ does the trimming only to the extent necessary to make the output tree an element of C_{n-1}. Note that $\tau_{\mathcal{A}(a_e)} = \{(t, 0) \mid t \in C_1^e\} \cup \{(t, 1) \mid t \in C_1^o\}$. It follows that for each tree $t = e_m(t_1, \ldots, t_m, y)$ with $m \geq 0$, $y \in \{1, 0\}$ and $t_1, \ldots, t_m \in C_{n-1}$, $a_1(t) \Rightarrow^*_{\mathcal{A}} e_m(t'_1, \ldots, t'_m, y)$ where for each i, $1 \leq i \leq m$,

(i) $t'_i \in C_{n-2}$, (ii) $t'_i = t_i$ if $t_i \in C_{n-2}$, and (iii) t'_i is even if and only if t_i is even.

Thus the tree $e_m(t'_1, \ldots, t'_m, y) \in C_{n-1}$ and it is even if and only if t is even. By the induction hypothesis we are done.

□

Lemma 5.7 For each $n \geq 1$, $dom((DTT^{DR})^n) \subset dom((DTT^{DR})^{n+1})$.

Proof. Obviously, $dom((DTT^{DR})^n) \subseteq dom((DTT^{DR})^{n+1})$. By Proposition 2.4, $C^e_{n+1} \notin DR_n$, and by Corollary 5.5, $dom((DTT^{DR})^n) \subseteq DR_n$. Hence $C^e_{n+1} \notin dom((DTT^{DR})^n)$. On the other hand, by Lemma 5.6, $C^e_{n+1} \in dom((DTT^{DR})^{n+1})$.

□

Lemma 5.8 For each $n \geq 1$, $dom((DTT^{DR})^n \circ DTT) = dom((DTT^{DR})^n)$.

Proof. For any two relations τ_1 and τ_2, $dom(\tau_1 \circ \tau_2) = dom(\tau_1 \circ ID(dom(\tau_2)))$. Since $dom(DTT) = dom(DTA) = DR$ and $ID(dom(DTA)) = DTA$, we have

$$\begin{aligned} dom((DTT^{DR})^n \circ DTT) &= dom((DTT^{DR})^n \circ DTA) \\ &= dom((DTT^{DR})^{n-1} \circ DTT^{DR} \circ DTA) \\ &= dom((DTT^{DR})^{n-1} \circ DTT^{DR}) \qquad \text{by Lemma 5.2} \\ &= dom((DTT^{DR})^n). \end{aligned}$$

□

The following theorem is an immediate consequence of Lemma 5.7 and Lemma 5.8. It settles the second inclusion of (3).

Theorem 5.9 For each $n \geq 1$, $(DTT^{DR})^n \circ DTT \subset (DTT^{DR})^{n+1}$.

To prove the first inclusion of (3): $(DTT^{DR})^n \subset (DTT^{DR})^n \circ DTT$, we need some further preparation. For every pair (n, k) of nonnegative integers, we define the tree language $C_{n,k}$ as follows:

(a) $C_{0,k} = C_{n,0} = \{1, 0\}$, for all $k, n \geq 0$.

(b) $C_{n,k} = \{\, \sigma(t,r) \mid t \in C_{n-1,k-1}$ and $r \in C_{n,k-1} \,\}$ for $k, n \geq 1$.

We observe that for each k and n, $C_{n,k}$ is a finite tree language and that $C_{n,k} \subset C_n$; obviously, C_n is infinite if $n \geq 1$. Moreover, it can be easily shown that a tree t in C_n belongs to $C_{n,k}$ if and only if the following conditions hold for each root-to-leaf path of t.

(a) The length of the path is at most k.
(b) The path chooses the left child at most n times.
(c) If the path chooses the left son less than n times then its length is exactly k.

By the above characterization of $C_{n,k}$ we also observe that $\bigcup_{k=0}^{\infty} C_{n,k} \subset C_n$. We put $C^e_{n,k} = C_{n,k} \cap L_e$ and $C^o_{n,k} = C_{n,k} \cap L_o$. Obviously we have $\sharp(C^e_{n,k}) = \sharp(C^o_{n,k})$. Fülöp and Vágvölgyi [11] have proved the following helpful result.

Lemma 5.10 For every $n \geq 1$ and every tree language $L \in T_\Sigma$, if $L \in DR_{n-1}$, then

$$\lim_{k \to \infty} \frac{\sharp(L \cap C^e_{n,k}) - \sharp(L \cap C^o_{n,k})}{\sharp(C_{n,k})} = 0 \ .$$

We are now ready to prove the first inclusion in (3)

Theorem 5.11 For each $n \geq 1$, $(DTT^{DR})^n \subset (DTT^{DR})^n \circ DTT$.

Proof. By contradiction. Let $\Sigma = \Sigma_0 \cup \Sigma_2$, $\Sigma_0 = \{\, 1, 0 \,\}$, $\Sigma_2 = \{\, \sigma \,\}$ and $\Delta = \Delta_0 = \{\, \$ \,\}$. By the proof of Lemma 5.6, it is easy to see that the relation $\{(\sigma(p,q), \sigma(0,0)) \mid p, q \in C^e_n\} \in (DTT^{DR})^n$. Moreover, the tree transformation $\{\, (\sigma(p,q), \$) \mid p, q \in T_\Sigma \,\}$ is in DTT. Hence the tree transformation $\rho = \{\, (\sigma(p,q), \$) \mid p, q \in C^e_n \,\} \subseteq T_\Sigma \times T_\Delta$ is in $(DTT^{DR})^n \circ DTT$. Let us suppose that $\rho \in (DTT^{DR})^n$. Then by Theorem 5.5, there is a $\mathrm{dtt}^{DR_{n-1}}$ $\mathcal{A} =< \Sigma, \Gamma, A, a, P >$ such that $\rho = \tau_{\mathcal{A}}$. Without loss of generality we may assume that each rule of $\mathcal{A}$ with left-hand side $a(\sigma(x_1, x_2))$ can be applied in the first step of some derivation $a(\sigma(p,q)) \Rightarrow^*_{\mathcal{A}} \$$ of $\mathcal{A}$. Hence each rule with left-hand side $a(\sigma(x_1, x_2))$ may have either one of the following three forms:

$$\begin{array}{ll} < a(\sigma(x_1,x_2)) \to b(x_1); L_1, L_2 > & \text{where } b \in A, \\ < a(\sigma(x_1,x_2)) \to b(x_2); L_1, L_2 > & \text{where } b \in A, \text{ or} \\ < a(\sigma(x_1,x_2)) \to \$; L_1, L_2 > . & \end{array}$$

Consider all rules with left-hand side $a(\sigma(x_1, x_2))$ that delete the variable x_1 :

$$< a(\sigma(x_1,x_2)) \to a_1(x_2); K_1, L_1 > , \ldots, \ < a(\sigma(x_1,x_2)) \to a_i(x_2); K_i, L_i >,$$
$$< a(\sigma(x_1,x_2)) \to \$; K_{i+1}, L_{i+1} > , \ldots, \ < a(\sigma(x_1,x_2)) \to \$; K_j, L_j >,$$

where $0 \leq i \leq j$ and $K_m, L_m \in DR_{n-1}$ for $1 \leq m \leq j$. Suppose that there is an m, $1 \leq m \leq j$, such that $K_m \not\subseteq C^e_n$ and let $r \in K_m - C^e_n$. By our assumption, there are trees p_m, q_m such that the mth rule can be applied in the first step of some derivation

$a(\sigma(p_m, q_m)) \Rightarrow^*_{\mathcal{A}} \$$ of $\mathcal{A}$. Since x_1 is deleted and $r \in K_m$, $a(\sigma(r, q_m)) \Rightarrow^*_{\mathcal{A}} \$$ holds as well. This contradicts the definition of ρ. Thus, for each $1 \leq m \leq j$, $K_m \subseteq C^e_n$.
Hence, by Lemma 5.10, for each $1 \leq m \leq j$,

$$\lim_{k\to\infty} \frac{\sharp(K_m \cap C^e_{n,k})}{\sharp(C_{n,k})} = 0.$$

This being true for every $1 \leq m \leq j$, it follows that $\cup^j_{m=1} K_m \subset C^e_n$. Pick a tree $u \in C^e_n - \cup^j_{m=1} K_m$. In an analogous fashion consider all the rules with left-hand side $(a(\sigma(x_1, x_2))$ which delete the variable x_2 :

$$< a(\sigma(x_1, x_2)) \to a_1(x_1); M_1, N_1 > , \ldots, \ < a(\sigma(x_1, x_2)) \to a_k(x_1); M_k, N_k >,$$
$$< a(\sigma(x_1, x_2)) \to \$; M_{k+1}, N_{k+1} > , \ldots, \ < a(\sigma(x_1, x_2)) \to \$; M_l, N_l >,$$

where $0 \leq k \leq l$ and M_m, $N_m \in DR_{n-1}$ for $1 \leq m \leq l$. Note that $l = k + j - i$ and $K_{i+m} = M_{k+m}$, $L_{i+m} = N_{k+m}$ for $1 \leq m \leq j - i$. By analogous arguments one can easily show that there exists a tree v such that $v \in C^e_n - \cup^l_{m=1} N_m$. Consider the tree $\sigma(u, v)$. It should be clear that the tree $\sigma(u, v) \in dom(\rho)$ is not in $dom(\tau_A)$. Contradiction.

□

We now show (4).

Lemma 5.12 $\bigcup^\infty_{n=0} (DTT^{DR})^n \subset DTT^R$.

Proof.

$$\begin{aligned}
\textstyle\bigcup^\infty_{n=0} dom((DTT^{DR})^n) &\subseteq \textstyle\bigcup^\infty_{n=0} DR_n && \text{by Corollary 5.5.} \\
&\subset R && \text{by Theorem 4.6 in [11]} \\
&= dom(DTT^R) && \text{by results in [5].}
\end{aligned}$$

□

We now summarize the results of this section in the following theorem.

Theorem 5.13 The diagram in Figure 1 is an inclusion diagram for $\{ ||u|| \mid u \in N \}$.

In the light of Theorem 5.13, the following result is obtained by direct inspection of the inclusion diagram of Figure 1.

Consequence 5.14 For any $u, v \in N$, $||u|| = ||v||$ if and only if $u = v$.

Lemma 5.4 leaves open the question of equality of the classes $(DTT^{DR})^n$ and DTT^{DR_n}. We conjecture that these classes are not equal. In fact we make the following stronger conjecture.

Conjecture 5.15 $DTT^{DR_2} - \bigcup^\infty_{n=0} (DTT^{DR})^n \neq \emptyset$.

6 The Rewriting System S_M

In this section we implement step (V), the last step, of the (I)-(V) method suggested in Section 3. That is, we have to give an algorithm which for every $w \in M^*$ computes a normal form of w, i.e. a string $u \in N$ such that $w \leftrightarrow^*_{T_M} u$, where T_M is the Thue system constructed in Section 4. We will need the following theorem.

Theorem 6.1 ([1], [3]) For any Thue system T (weight reducing string rewriting system S) there is a linear time algorithm which for every word w computes a T-irreducible (S-irreducible) element u such that $w \leftrightarrow^*_T u$ ($w \leftrightarrow^*_S u$).

Note that if the system T (S) in Theorem 6.1 is Church-Rosser, then, by Theorem 2.5, u is the unique irreducible element in the congruence class containing w. However, our Thue system T_M is not Church-Rosser because $DTA \cdot DTT$ and $DTT \cdot DTT$ are both irreducible while being congruent with respect to T_M. Fortunately, this is not a serious obstacle as we are able to define a weight reducing (and hence noetherian) rewriting system S_M such that T_M and S_M induce identical congruence classes (i.e., $\leftrightarrow^*_{T_M}$ and $\leftrightarrow^*_{S_M}$ are equal), each such class contains a unique irreducible element with respect to S_M, and $N = IRR(S_M)$. Once such an S_M has been defined and shown to have the required properties, Theorem 6.1 gives us the linear time algorithm needed to satisfy step (V) of our method.

Define the rewrite system S_M to consist of exactly the pairs of T_M (of course S_M uses those pairs "one-way" only, while T_M can use them in a "two-way" fashion), and let the weight function $\rho : M \to \{1, 2, \dots\}$ be defined by: $\rho(DTA) = 1$ and $\rho(DTT) = \rho(DTT^{DR}) = \rho(DTT^R) = 2$. It is easy to check that S_M is weight reducing with respect to ρ, and that $\leftrightarrow^*_{S_M}$ is identical to $\leftrightarrow^*_{T_M}$.

Theorem 6.2 $N = IRR(S_M)$.

Proof. It should be clear that $N \subseteq IRR(S_M)$. Conversely, we show that $IRR(S_M) \subseteq N$, that is, for each word $w \in IRR(S_M)$, $w \in N$. We proceed by induction on the length of w. If $|w| \le 1$, then

$$w \in \{I, DTA, DTT, DTT^{DR}, DTT^R\} \subset N.$$

Now suppose that $|w| = n > 1$ and our statement holds for all words v with $|v| \le n-1$. Then $w = v \cdot U$, where $U \in M$, v is irreducible with respect to S_M, and $1 \le |v| = n-1$. It follows that $U \notin \{DTT^R, DTA\}$, because for each $V \in M$ and $U \in \{DTT^R, DTA\}$, $V \cdot U$ is a left-hand side of some rule in S_M. Thus either $U = DTT$ or $U = DTT^{DR}$.

First, consider the case where $U = DTT$. For each $V \in \{DTT, DTT^R\}$, $V \cdot DTT$ is a left-hand side of some rule of S_M. Hence the last letter V of v is either DTA or DTT^{DR}. By the induction hypothesis, $v \in N$; hence, if $V = DTA$ then $v = DTA$, and if $V = DTT^{DR}$, then $v = (DTT^{DR})^k$ for some $k \ge 1$. Thus, either $w = DTA \circ DTT \in N$ or $w = (DTT^{DR})^k \circ DTT \in N$.

Now suppose that $U = DTT^{DR}$. For each $V \in \{DTA, DTT, DTT^R\}$, $V \circ DTT^{DR}$ is a left-hand side of some rule of S_M. Hence the last letter V of v is DTT^{DR}. By the induction hypothesis, $v \in N$, and hence $v = (DTT^{DR})^k$ for some $k \geq 1$. Thus $w = (DTT^{DR})^{k+1} \in N$. □

It follows from Theorem 6.2 that in every congruence class of $\leftrightarrow^*_{S_M}$ (and hence in every congruence class of $\leftrightarrow^*_{T_M}$) there is exactly one S_M-irreducible element which is also T_M-irreducible . (Incidentally, by Theorem 2.5, this implies that S_M is Church-Rosser.) Moreover, the algorithm in Theorem 6.1 computes, for every word $w \in M^*$, the unique normal form $u \in N$ of w (because $N = IRR(S_M)$). By Theorems 6.1 and 6.2 we have obtained the following result.

Theorem 6.3 There is a linear time algorithm that for every word w computes an S_M-irreducible element $u \in N$ such that $w \leftrightarrow^*_{S_M} u$.

7 Summary

In this section we summarize our results. We have carried out the implementation of steps (I)-(V), of the methodology described in Section 3, that is to say,

(I) We gave a finite relation $T_M \subseteq M^* \times M^*$ which is our candidate for the set of generators of θ.

(II) We proved that for every $(u, v) \in T_M$, $||u|| = ||v||$.

(III) We gave a subset $N \subseteq M^*$ which is a candidate for a set of representatives of θ.

(IV) We gave the inclusion diagram of the set $||N|| = \{ \, ||u|| \mid u \in N \, \}$, which is the set of tree transformation classes represented by the elements of N. By using this inclusion diagram we verified that for any $u, v \in N$, $||u|| = ||v||$ if and only if $u = v$.

(V) We have shown that the linear time algorithm of [3] can be applied such that for every $w \in M^*$ the algorithm computes the unique $u \in N$ such that $w \leftrightarrow^*_T u$.

By Lemma 3.2, this implies the following result.

Theorem 7.1

(a) N, the set of normal forms, is a set of representatives for the congruence classes of θ.

(b) The diagram of Figure 1 is the inclusion diagram of the set $||N|| = \{ \, ||u|| \mid u \in N \, \}$, i.e., of the set of tree transformation classes represented by normal forms.

(c) $T_M \subseteq M^* \times M^*$ is a finite set of generators of θ (i.e., a Thue system over M such that $\leftrightarrow^*_T = \theta$) and we have shown the applicability of an algorithm which for every $w \in M^*$, by a suitable sequence of substitutions induced by T_M, computes the normal form of w, i.e., the unique $u \in N$ for which $w \leftrightarrow^*_T u$.

Therefore, by Theorem 3.1, we obtain the following theorem which answers the question posed in Section 3.

Theorem 7.2 There is a linear time algorithm which for any tree transformation classes $Y_1, \ldots, Y_m, Z_1, \ldots, Z_n \in M$ decides which one of the following four mutually exclusive conditions holds.

(i) $Y_1 \circ \ldots \circ Y_m \; = \; Z_1 \circ \ldots \circ Z_n$, (ii) $Y_1 \circ \ldots \circ Y_m \subset Z_1 \circ \ldots \circ Z_n$,
(iii) $Z_1 \circ \ldots \circ Z_n \subset Y_1 \circ \ldots \circ Y_m$, (iv) $Y_1 \circ \ldots \circ Y_m \bowtie Z_1 \circ \ldots \circ Z_n$.

References

[1] R. V. Book, Confluent and other types of Thue systems, *J. Assoc. Comput. Mach.* **29** (1982) 171-182.

[2] R. V. Book, Thue systems and the Church-Rosser property: replacement systems, specification of formal languages and presentations of monoids, in *Progress in Combinatorics on Words*, (L. Cummings ed.) Academic Press, 1983, New York, pp. 1-38.

[3] V. Book and P. O'Dunlaing, Testing for the Church-Rosser property, *Theoret. Comput. Sci.* **16** (1981) 223-229.

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

[5] J. Engelfriet, Top-down tree transducers with regular look-ahead, *Mathematical Systems Theory*, **10** (1976/1977) 289-303.

[6] J. Engelfriet and H. Vogler, Macro tree transducers, *J. Comput. System Sci.* **31** (1985) 71-146.

[7] J. Engelfriet and H. Vogler, Pushdown machines for the macro tree transducer, *Theoret. Comput. Sci.* **42** (1986) 251-368.

[8] J. Engelfriet and H. Vogler, High level tree transducers and iterated pushdown transducers, *Acta Inform.* **26** (1988) 131-192.

[9] Z. Fülöp and S. Vágvölgyi, Top-down tree transducers with determinstic top-down look-ahead, *Inform. Process. Lett.* **33** (1989/90) 3-5.

[10] Z. Fülöp and S. Vágvölgyi, Variants of top-down tree transducers with look-ahead, *Mathematical Systems Theory* **21** (1989) 125-145.

[11] Z. Fülöp and S. Vágvölgyi, Iterated deterministic top-down look-ahead, in *Proc. FCT'89*, *Lecture Notes in Computer Science*, Vol. **380**, Springer-Verlag, 1989, Berlin, pp. 175-184.

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

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

[14] M. Jantzen, *Confluent String Rewriting*, Springer-Verlag, Berlin, 1988.

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

Tech Report: TR 93-10a
Submission Date: April 13, 1993