Deterministic top-down tree transducers with iterated look-ahead

Giora Slutzki, Sándor Vágvolgyi* 1

Department of Computer Science, Iowa State University, Ames, IA 50011, USA

Received May 1993; revised March 1994
Communicated by A. Salomaa

Abstract

It is known that by iterating the look-ahead tree languages for deterministic top-down tree automata, more and more powerful recognizing devices are obtained. Let \( DR_0 = DR \), where \( DR \) is the class of all tree languages recognizable by deterministic top-down tree automata, and let, for \( n \geq 1 \), \( DR_n \) be the class of all tree languages recognizable by deterministic top-down tree automata with \( DR_{n-1} \) look-ahead. Then \( DR_0 \subseteq DR_1 \subseteq DR_2 \subseteq \ldots \). Slutzki and Vágvolgyi (1993) showed that the composition powers of the class of all deterministic top-down tree transformations with deterministic top-down look-ahead (\( DTT^{DR} \)) form a proper hierarchy, i.e. 
\[
(DTT^{DR})^* \subseteq (DTT^{DR})^{n+1} \quad n \geq 0.
\]
Along the proof they studied the notion of the deterministic top-down tree transducer with \( DR_n \) look-ahead (\( dtt^{DR_n} \)) and showed that 
\[
(DTT^{DR})^{n+1} \subseteq DTT^{DR_n} \quad (n \geq 0),
\]
where \( DTT^{DR_n} \) stands for the class of all tree transformations induced by \( dtt^{DR_n} \)'s. Our aim is to show the reversed inclusion, i.e. 
\[
DTT^{DR_n} \subseteq (DTT^{DR})^{n+1} \quad (n \geq 0).
\]
This implies a precise characterization \( DTT^{DR_n} = (DTT^{DR})^{n+1} \quad (n \geq 0) \), and implies that the classes \( DTT^{DR_n} \quad (n \geq 0) \) form a proper hierarchy.

1. Introduction

Top-down tree transducers (the induced class of tree transformations is denoted by \( TT \)) were originally introduced [16, 18] as models of syntax-directed translation [13]. It was immediately shown in [16, 18] that top-down tree transformations are not closed under composition, i.e. \( TT \subset TT^2 \), and it was conjectured in [1, 14, 15] that iterating composition of \( TT \) gives rise to a proper hierarchy. This conjecture was finally proved by Engelfriet [4], see also [5]; that is, it was proved that for all \( n \geq 0 \), \( TT^n \subset TT^{n+1} \). Interestingly, deterministic top-down tree transformations (denoted by \( DTT \)) are also not closed under composition, i.e. \( DTT \subset DTT^2 \), but Fülöp and

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

0304-3975/95/$09.50 © 1995—Elsevier Science B.V. All rights reserved
SSDI 0304-3975(94)00111-U
Vágvölgyi [9] have shown that $DTT^2 = DTT^3$. Thus, in the deterministic case, the hierarchy $DTT^n (n \geq 1)$ collapses to the second level.

Top-down (deterministic and nondeterministic) tree transducers with regular look-ahead (the classes of the induced tree transformations are denoted, respectively, by $DTT^R$ and $TT^R$) were introduced and studied in [3]. (The regular look-ahead is a look-ahead computable by a nondeterministic top-down tree automaton, see, for example, [19].) It was shown there that $DTT^R$ is closed under composition, whereas $TT^R$ is not. That is, $(DTT^R)^2 = DTT^R$ and $TT^R \subset (TT^R)^2$. Indeed it easily follows from the results of [3, 4] that, as in the case without look-ahead, iterating composition of top-down tree transducers with regular look-ahead produces a proper hierarchy, i.e. for all $n \geq 0$, $(TT^R)^n \subset (TT^R)^{n+1}$. Of course, because of closure under composition, the corresponding deterministic hierarchy collapses to the first level.

In [10] Fülöp and Vágvölgyi introduced and studied deterministic and nondeterministic top-down tree transducers with deterministic top-down look-ahead (the classes of the induced tree transformations are denoted, respectively, by $DTT^{DR}$ and $TT^{DR}$). The deterministic top-down look-ahead is a look-ahead computable by a deterministic top-down tree automaton. It is again easy to show that $TT^{DR}$ is not closed under composition and that iterating composition gives rise to a proper hierarchy, i.e. for every $n \geq 0$, $(TT^{DR})^n \subset (TT^{DR})^{n+1}$. Indeed, by the results in [3, 4], the three “iterated-composition” hierarchies $TT^n (n \geq 1)$, $(TT^R)^n (n \geq 1)$, and $(TT^{DR})^n (n \geq 1)$, each being proper on its own, mesh into a single hierarchy:

$$TT^n \subset (TT^{DR})^n \subset (TT^R)^n \subset TT^{n+1} \quad (n \geq 1)$$

which is (of course) infinite, but not known to be proper at every one of its inclusions.

The main thrust in [10] was the study of deterministic top-down tree transducers with deterministic top-down look-ahead (denoted, as mentioned above, by $DTT^{DR}$). It was shown there that $DTT^{DR}$ is not closed under composition, but the question of whether iterating composition leads to a proper hierarchy was left open. This question was recently settled in [17]: for every $n \geq 0$, $(DTT^{DR})^n \subset (DTT^{DR})^{n+1}$. The proof in [17] uses the following classes of tree languages defined previously in [11]:

$$DR_0 = DR,$$

$$DR_{n+1} = \text{dom}(DTA^{DR_n}) \quad (n \geq 1),$$

where $\text{dom}(DTA^{DR_n})$ is the class of all tree languages recognized by deterministic top-down tree automata with look-ahead languages from the class $DR_n$ (as previously mentioned, $DR$ is the class of all tree languages recognized by deterministic top-down tree automata without look-ahead). Fülöp and Vágvölgyi [11] proved that these classes form a proper hierarchy within the class of recognizable tree languages (which, recall, was denoted by $R$), i.e. for every $n \geq 0$, $DR_n \subset DR_{n+1}$.

In [17] the authors defined and studied deterministic top-down tree transducers with $DR_n$ look-ahead (the class of the induced tree transformations is denoted by
and have shown that
\[(DTT^{DR})^{n+1} \subseteq DTT^{DR^n},\]  
which intuitively means that composition of \((n + 1)\) many deterministic top-down tree transducers with \(DR\) look-ahead can be computed by a single deterministic top-down tree transducer with sufficiently powerful look-ahead. For some (too long) time the authors have suspected (and attempted, in vain, to prove) that
\[DTT^{DR^2} \nsubseteq C \cap O(DTT^{DR}) \neq 0.\]
In this paper we show that this inequality does not hold; indeed we are able to prove the converse of \((\dagger\dagger)\), i.e., for every \(n \geq 0\),
\[DTT^{DR^n} \subseteq (DTT^{DR})^{n+1}.\]  
This proof is the main technical result of this paper. It is rather long and involved. By \((\dagger\dagger)\) we have a full characterization
\[DTT^{DR^n} = (DTT^{DR})^{n+1}\]
and the consequence that \(DTT^{DR^n} (n \geq 1)\) forms a proper hierarchy. The proof of \((\dagger)\) is based on the decomposition result
\[DTT^{DR^n} \subseteq DTT^{DR} \circ DTT^{DR}_{n-1}\]  
which, in turn, is proved by induction. For the base level, \(n = 1\), we have to argue how to trade the look-ahead power \(DR_1 = \text{dom}(DTA^{DR})\), with the composition operation involving two deterministic top-down tree transducers with only \(DR = DR_0\) look-ahead. In the general case \(n \geq 2\), in a nutshell, we "unfold" the look-ahead all the way to the \(n = 1\) case, use the base level result, and then "fold" the look-ahead back.

The paper is organized as follows. In Section 2 we give precise definitions of the various classes of tree transducers discussed in this paper. In Section 3 we present the main technical results, viz. the proof of \((\dagger)\). Because of its length, this section is split into two parts; Part A deals with the base case, \(n = 1\), and Part B treats the case \(n \geq 2\). Section 4 summarizes the results, draws some immediate consequences, and poses some open problems.

2. Preliminaries

A ranked alphabet \(\Sigma\) 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 \(\Sigma_m\) the set of symbols in \(\Sigma\) which have rank \(m\). For a ranked alphabet \(\Sigma\) and a set \(H\), the set of trees (or terms) over \(\Sigma\) 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 \( \Sigma \) is \( T_\Sigma(\emptyset) \), and we write \( T_\Sigma \) for \( T_\Sigma(\emptyset) \). We specify a countable set \( X = \{x_1, x_2, \ldots\} \) 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 \( \Sigma = \{a\} \) and \( \Sigma_1 = \{a\} \) then \( \sigma(x_1, a(x_1)) \in T_\Sigma(X_1) \) but \( \sigma(x_1, a(x_1)) \notin \bar{T}_\Sigma(X_1) \). On the other hand, \( \sigma(x_1, a(x_2)) \in \bar{T}_\Sigma(X_2) \). For a unary ranked alphabet \( A \) and a set \( L \) of terms, \( A(L) \) denotes the set \( \{a(t) \mid a \in A \text{ and } t \in L\} \).

The notion of tree substitution is defined as follows. Let \( m \geq 0, t \in T_\Sigma(X_m) \), and \( t_1, \ldots, t_m \in T_\Sigma \). We denote by \( t[t_1, \ldots, t_m] \) the tree over \( \Sigma \) which is obtained from \( t \) by replacing each occurrence of \( x_i \) in \( t \) by \( t_i \) for every \( 1 \leq i \leq m \).

A partition of \( T_\Sigma \) is a set \( \Pi \) of nonempty subsets of \( T_\Sigma \) such that

(i) for any two different sets \( A, B \in \Pi \), \( A \cap B = \emptyset \),
(ii) \( T_\Sigma = \bigcup_{A \in \Pi} A \).

For each partition \( \Pi \) of \( T_\Sigma \), the corresponding equivalence relation on \( T_\Sigma \) is denoted by \( \rho(\Pi) \). For a set \( U \) of partitions, let the coarsest refinement of \( U \), denoted by

\[ \bigwedge_{\Pi \in U} \Pi, \]

be the partition corresponding to the equivalence relation \( \bigcap_{\Pi \in U} \rho(\Pi) \).

Let \( \Sigma \) and \( \Delta \) be two ranked alphabets. Then any subset of \( T_\Sigma \times T_\Delta \) is a tree transformation from \( T_\Sigma \) to \( T_\Delta \). For a tree language \( L \), the partial identity \( \{(t, t) \mid t \in L\} \) is denoted by \( ID(L) \).

**Definition 2.1.** A top-down tree transducer (tt for short) is a system \( \mathcal{A} = \langle \Sigma, \Delta, A, A_0, P \rangle \), where

1. \( \Sigma \) is a ranked input alphabet;
2. \( \Delta \) is a ranked output alphabet;
3. \( A \) is a unary ranked state alphabet such that \( A \cap (\Sigma \cup \Delta \cup X) = \emptyset \);
4. \( A_0 \) is a subset of \( A \), the set of initial states;
5. \( P \) is a finite set of rules of the form

\[ a(\sigma(x_1, \ldots, x_m)) \rightarrow t, \]

where \( m \geq 0, a \in \Sigma_m, a \in A \), and \( t \in T_\Delta(A(X_m)) \).

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 \( a(\sigma(x_1, \ldots, x_m)) \rightarrow r \) 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

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

We now introduce some special types of $\mathcal{T}$'s. Let $\mathcal{A} = \langle \Sigma, A, A_0, P \rangle$ be a $\mathcal{T}$. 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 $A = \emptyset$ and each rule in $P$ is of the form $a(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(a_1(x_1), \ldots, a_m(x_m))$ where $a, a_1, \ldots, a_m \in A$; in that case, the tree transformation $\tau_{\mathcal{A}}$ is a partial identity on $T_\Sigma$;

(c) a deterministic top-down tree automaton (dta) if $\mathcal{A}$ is a ta and a dtt.

The class of all tree transformations computed by $\mathcal{T}$'s (respectively, dtt's, ta's, and dta's) is denoted by $\mathcal{T}$ (respectively, $\mathcal{DTT}$, $\mathcal{TA}$, and $\mathcal{DTA}$). The tree language recognized by ta $\mathcal{A}$ is $L(\mathcal{A}) = \text{dom}(\tau_{\mathcal{A}})$. The classes of tree languages recognized by ta's and dta's are

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

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 $DR \subseteq R$ or equivalently $\mathcal{DTA} \subseteq \mathcal{TA}$; a proof can be found in [2] or [12].

Top-down tree transducers with look-ahead, one of the main topics of this paper, were defined in [S]. It transpired that they have a number of nice properties, especially in the deterministic case. For example, the class of deterministic top-down tree transformations with regular look-ahead is closed under composition. The concept of look-ahead also proved useful in other contexts [6-8]. Following [3], 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 $C$ be a class of tree languages. A top-down tree transducer with $C$ look-ahead (tt$^C$) is a system $\mathcal{A} = \langle \Sigma, A, A_0, P \rangle$, where the components are defined exactly as in Definition 2.1, except that the rules in $P$ are of the form

$$\langle a(\sigma(x_1, \ldots, x_m)) \rightarrow t; L_1, \ldots, L_m \rangle,$$

where

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

is an ordinary $\mathcal{T}$-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 communication of $\mathcal{A}$ is the binary relation $\Rightarrow_{\mathcal{A}}$ on $T_\Sigma$ defined such that $t \Rightarrow_{\mathcal{A}} s$ if and only if

(a) there is a rule $\langle a(\sigma(x_1, \ldots, x_m)) \rightarrow r; L_1, \ldots, L_m \rangle$ 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

$$\tau_{ \mathcal{A} } = \{(t, s) \in T_\Sigma \times T_A | a(t) \Rightarrow_{ \mathcal{A} } s \text{ for some } a \in A_0\}$$

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

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

(a) a top-down tree automaton with $C$ look-ahead ($ta^C$) if $\Sigma = \Delta$ and each rule in $P$ is of the form $\langle a(\sigma(x_1, \ldots, x_m)) \Rightarrow \sigma(a_1(x_1), \ldots, a_m(x_m)); L_1, \ldots, L_m \rangle$ 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 $i, 1 \leq i \leq m$, whenever two different rules in $P$: $\langle a(\sigma(x_1, \ldots, x_m)) \Rightarrow r_1; L_1, \ldots, L_m \rangle$ and $\langle a(\sigma(x_1, \ldots, x_m)) \Rightarrow r_2; L_1', \ldots, L_m' \rangle$ have the same left-hand side,

(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 $L(\mathcal{A}) = \text{dom}(\tau_{ \mathcal{A} })$. Given a $ta^C$ $\mathcal{A} = \langle \Sigma, \Delta, A, A_0, P \rangle$, and a state $a \in A$, let $\mathcal{A}' = \langle \Sigma, \Delta, A, a, P \rangle$, and $L(\mathcal{A}, a) = L(\mathcal{A}')$. Thus $L(\mathcal{A}, a)$ stands for the tree language recognized by $\mathcal{A}$ starting from the state $a$.

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 $DTA^C$). The following result was proved in [3].

**Proposition 2.2.** Let $\mathcal{A}$ be a $tt^R$. Then $\text{dom}(\tau_{ \mathcal{A} }) \subseteq R$.

By Proposition 2.2, we can iterate the look-ahead tree languages, without leaving $R$, as follows. Let $DR_0 = DR$ and let, for $n \geq 1$, $DR_n$ be the class of tree languages recognizable by deterministic top-down tree automata with $DR_{n-1}$ look-ahead. By Proposition 2.2, $DR_n \subseteq R$ for every $n \geq 0$. Fülop and Vágvolgyi [11] proved the following result.

**Proposition 2.3.** For each $n \geq 1$, $DR_{n-1} \subseteq DR_n$. 
3. The results

First we prove two preparatory lemmas.

**Lemma 3.1.** Let $\mathcal{A} = \langle \Sigma, \Sigma, A, a_0, P_a \rangle$, $\mathcal{B} = \langle \Sigma, \Sigma, B, b_0, P_b \rangle$ be two dta's. Let $a \in A$ and $b \in B$ be states such that $L(\mathcal{A}, a) \cap L(\mathcal{B}, b) = \emptyset$. Then for each $\sigma \in \Sigma_m$ ($m \geq 1$), if

$$a(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(a_1(x_1), \ldots, a_m(x_m)) \in P_a$$

and

$$b(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(b_1(x_1), \ldots, b_m(x_m)) \in P_b,$$

then there exists an $i$ ($1 \leq i \leq m$) such that $L(\mathcal{C}, a_i) \cap L(\mathcal{D}, b_i) = \emptyset$.

**Proof.** If for each $i$ ($1 \leq i \leq m$) there is a tree $p_i$ in the intersection $L(\mathcal{C}, a_i) \cap L(\mathcal{D}, b_i)$, then the tree $\sigma(p_1, \ldots, p_m)$ is in $L(\mathcal{A}, a) \cap L(\mathcal{B}, b)$. Contradiction. □

**Lemma 3.2.** Let $L, M \in DR$ be two nonempty tree languages over $\Sigma$ such that $L \cap M = \emptyset$. Then there exist tree languages $L'$, $M'$, $N \in DR$ such that $L \subseteq L'$, $M \subseteq M'$ and either $\{L', M', N\}$ or $\{L', M'\}$ is a partition of $T_\Sigma$.

**Proof.** Let $\mathcal{A} = \langle \Sigma, \Sigma, A, a_0, P_a \rangle$, $\mathcal{B} = \langle \Sigma, \Sigma, B, b_0, P_b \rangle$ be two dta's such that $L = L(\mathcal{A})$ and $M = L(\mathcal{B})$. Without loss of generality we may assume that $\mathcal{A}$ and $\mathcal{B}$ reject only at the leaves, that is, for each $\sigma \in \Sigma_m$ ($m \geq 1$), $a \in A$ and $b \in B$, there exists a rule with left-hand side $a(\sigma(x_1, \ldots, x_m))$ in $P_a$ and there exists a rule with left-hand side $b(\sigma(x_1, \ldots, x_m))$ in $P_b$. We now define dta's $\mathcal{A}'$, $\mathcal{B}'$ and $\mathcal{C}$ such that $L(\mathcal{A}') = L'$, $L(\mathcal{B}') = M'$, and $L(\mathcal{C}) = N$. The dta's $\mathcal{A}'$, $\mathcal{B}'$ and $\mathcal{C}$ will have a common set of states:

$$C = \{[a, b] \in A \times B | L(\mathcal{A}, a) \cap L(\mathcal{B}, b) = \emptyset\} \cup \{id\},$$

and the same initial state $[a_0, b_0] \in C$. The three dta's will also have almost identical rules. We define now the set $P$ of those rules which are present in all three dta's. For each $m \geq 0$, $\sigma \in \Sigma_m$, we let

$$id(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(id(x_1), \ldots, id(x_m)) \in P.$$  

For each $m \geq 1$, $\sigma \in \Sigma_m$, and $[a, b] \in C$, consider two rules

$$a(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(a_1(x_1), \ldots, a_m(x_m)) \in P_a$$

and

$$b(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(b_1(x_1), \ldots, b_m(x_m)) \in P_b.$$  

By Lemma 3.1, let $j$ ($1 \leq j \leq m$) be smallest index such that

$L(\mathcal{A}, a_j) \cap L(\mathcal{B}, b_j) = \emptyset.$
For every $1 \leq i \leq m$ define

$$c_i = \begin{cases} [a_j, b_j] & \text{if } i = j, \\ \text{id} & \text{otherwise} \end{cases}$$

and put the rule \([a, b](\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(c_1(x_1), \ldots, c_m(x_m))\) in \(P\). Now define dta's \(\mathcal{A}' = \langle \Sigma, \Sigma, C, [a_0, b_0], P_{\mathcal{A}'} \rangle\), \(\mathcal{B} = \langle \Sigma, \Sigma, C, [a_0, b_0], P_{\mathcal{B}} \rangle\), and \(\mathcal{C} = \langle \Sigma, \Sigma, C, [a_0, b_0], P_{\mathcal{C}} \rangle\), where

$$P_{\mathcal{A}'} = P \cup \{[a, b](\sigma) \rightarrow \sigma \mid \sigma \in \Sigma_0, [a, b] \in C, a(\sigma) \rightarrow \sigma \in P_{\mathcal{A}'}\},$$

$$P_{\mathcal{B}} = P \cup \{[a, b](\sigma) \rightarrow \sigma \mid \sigma \in \Sigma_0, [a, b] \in C, b(\sigma) \rightarrow \sigma \in P_{\mathcal{B}}\},$$

$$P_{\mathcal{C}} = P \cup \{[a, b](\sigma) \rightarrow \sigma \mid \sigma \in \Sigma_0, [a, b] \in C, a(\sigma) \rightarrow \sigma \notin P_{\mathcal{C}} \& b(\sigma) \rightarrow \sigma \notin P_{\mathcal{C}}\}.$$

Now, if \(P_{\mathcal{C}} = P\), then \(L(\mathcal{C}) = N = \emptyset\) and \(N\) is not an element of the partition. To conclude the proof of Lemma 3.2, it suffices to prove the following claim.

Claim A. For every \([a, b] \in C, \text{ and } p \in T_\Sigma\), exactly one of the conditions: \(p \in L(\mathcal{A}', [a, b])\), \(p \in L(\mathcal{B}', [a, b])\), and \(p \in L(\mathcal{C}, [a, b])\) holds.

Proof (By induction on the structure of trees). If \(p \in \Sigma_0\), then the rule \([a, b](p) \rightarrow p\) is in exactly one of the sets \(P_{\mathcal{A}'}\), \(P_{\mathcal{B}'}\), \(P_{\mathcal{C}}\) and hence \(p\) belongs to exactly one of the languages \(L(\mathcal{A}', [a, b])\), \(L(\mathcal{B}', [a, b])\), and \(L(\mathcal{C}, [a, b])\).

Suppose \(p = \sigma(p_1, \ldots, p_m)\) for some \(\sigma \in \Sigma_m (m \geq 1)\). Since \(\mathcal{A}\) and \(\mathcal{B}\) reject only at the leaves, there are rules:

\[
\begin{align*}
   a(\sigma(x_1, \ldots, x_m)) &\rightarrow \sigma(a_1(x_1), \ldots, a_m(x_m)) \in P_{\mathcal{A}'}, \\
   b(\sigma(x_1, \ldots, x_m)) &\rightarrow \sigma(b_1(x_1), \ldots, b_m(x_m)) \in P_{\mathcal{B}'}.
\end{align*}
\]

As \([a, b] \in C\), \(L(\mathcal{A}', a) \cap L(\mathcal{B}', b) = \emptyset\). So, by Lemma 3.1, let \(j (1 \leq j \leq m)\) be smallest such that \(L(\mathcal{A}', a_j) \cap L(\mathcal{B}', b_j) = \emptyset\). By the definition of \(P\),

\[\sigma(x_1, \ldots, x_m) \rightarrow \sigma(c_1(x_1), \ldots, c_m(x_m)) \in P,\]

where for \(i \neq j, c_i = \text{id} \) and \(c_j = [a_j, b_j]\). By the induction hypothesis exactly one of the conditions:

\[p_j \in L(\mathcal{A}', [a_j, b_j]), \quad p_j \in L(\mathcal{B}', [a_j, b_j]), \quad p_j \in L(\mathcal{C}, [a_j, b_j])\]

holds. Moreover, for \(i \neq j, \text{id}(p_i) \Rightarrow * p_j\) holds for \(\mathcal{A}', \mathcal{B}', \text{ and } \mathcal{C}\). It follows that exactly one of the languages \(L(\mathcal{A}', [a, b])\), \(L(\mathcal{B}', [a, b])\), and \(L(\mathcal{C}, [a, b])\) contains \(p\). This proves Claim A.

Proof of the Lemma 3.2 is now complete. \(\square\)

In the remainder of this section we show that for each \(n \geq 1, DTT^{DR_n} \subseteq DTT^{DR} \circ DTT^{DR_{n-1}}\). Because of the length and complexity of the argument
we organize the presentation into two parts. In Part A, we argue the base case \( n = 1 \), which itself is rather involved. In Part B we present the general case, \( n \geq 2 \).

3.1. The Case \( n = 1 \) (Part A)

**Theorem 3.3.** \( \text{DTT}^{DR_1} \subseteq (\text{DTT}^{DR})^2 \).

**Proof.** Roughly, given a \( \text{dtt}^{DR_1} \mathcal{A} \), we construct two \( \text{dtt}^{DR} \)'s \( \mathcal{D} \) and \( \mathcal{E} \) such that \( \tau_{\mathcal{A}} = \tau_{\mathcal{D}} \circ \tau_{\mathcal{E}} \). \( \mathcal{D} \) will be a one-state (total) relabeling which for an input tree \( p \), using its capacity of \( DR \) look-ahead, partially simulates all look-ahead automata of \( \mathcal{A} \) and puts enough information in the tree, resulting in tree \( p' \), so as to enable \( \mathcal{E} \) to simulate \( \mathcal{A} \) on \( p \). Along the computation on an input tree, \( \mathcal{A} \) keeps on carrying out look-ahead tests. The look-ahead automata of \( \mathcal{A} \), say \( \mathcal{A}_1, \ldots, \mathcal{A}_K \), may arrive at a node \( v \), labelled by \( \sigma \in \Sigma_m \) \((m \geq 1)\), in several states. As \( \mathcal{D} \) does not know the states in which \( \mathcal{A}_j \) \((1 \leq j \leq K)\) arrives at \( v \), \( \mathcal{D} \) rewrites \( \sigma \) into a function symbol \( \psi_{\sigma} \), where \( \psi_{\sigma} \) is a finite mapping from the set of all states of the \( \mathcal{A}_j \)'s. The mapping \( \psi_{\sigma} \) assigns either \( \# \) or a rule of \( \mathcal{A}_j \) to a state \( b \) of \( \mathcal{A}_j \) \((1 \leq j \leq K)\). Here \( \psi_{\sigma}(b) = \# \) means that if \( \mathcal{A}_j \) arrives at \( v \) in state \( b \), then it cannot apply any rule, and its computation comes to a halt. Moreover, \( \psi_{\sigma}(b) = r \) means that if \( \mathcal{A}_j \) arrives at \( v \) in state \( b \), then \( \mathcal{A}_j \) either applies the rule \( r \), or cannot apply any rule because for each rule with left-hand side \( b(\sigma(x_1, \ldots, x_m)) \), some look-ahead tree language given in the rule does not contain the corresponding direct subtree of \( v \). We compute the look-ahead sets appearing in the rules of \( \mathcal{D} \) with left-hand side \( \langle d(\sigma(x_1, \ldots, x_m)), m \geq 1 \rangle \), as follows. For each \( 1 \leq i \leq m \), consider all pairs of rules of \( \mathcal{A}_j \)'s with the same left-hand side \( b(\sigma(x_1, \ldots, x_m)) \), \( \langle b(\sigma(x_1, \ldots, x_m)) \rightarrow u_1; L_{11}^{b_1}, \ldots, L_{1m}^{b_1} \rangle, \langle b(\sigma(x_1, \ldots, x_m)) \rightarrow u_k; L_{k1}^{b_k}, \ldots, L_{km}^{b_k} \rangle \), such that \( L_{ij}^{b_i} \cap L_{kj}^{b_k} = \emptyset \). Then, by Lemma 3.2, \( L_{ij}^{b_i} \) and \( L_{kj}^{b_k} \) induce a partition of \( T_x \). We take the coarsest refinement \( V_{\sigma} \) of all these partitions. The look-ahead set \( M_i \) assigned to the variable \( x_i \) in a rule \( \langle d(\sigma(x_1, \ldots, x_m)) \rightarrow \psi_{\sigma}(d(x_1, \ldots, d(x_m)); M_1, \ldots, M_m) \rangle \) is an element of \( V_{\sigma} \). Moreover, \( M_i \) is not disjoint from the \( i \)th look-ahead sets of the rules of appearing in the range of the mapping \( \psi_{\sigma} \). In fact, they all contain the subtree at the \( i \)th son of \( v \). On the basis of all this information written by \( \mathcal{D} \) in the tree, the look-ahead automata of \( \mathcal{E} \) are able to simulate the look-ahead automata of \( \mathcal{A} \) and hence \( \mathcal{E} \) can simulate \( \mathcal{A} \). The details of the construction and its correctness are given below.

Let \( \mathcal{A} = < \Sigma, \Delta, A, a_0, P_{\mathcal{A}} > \) be a \( \text{dtt}^{DR_1} \). Let \( L_i = L(\mathcal{A}), 1 \leq i \leq K \), be all the look-ahead sets appearing in the rules of \( \mathcal{A} \) where \( \mathcal{A} = < \Sigma, \Sigma, B^i, b_0, P_{\mathcal{A}} > \) is a \( \text{dtt}^{DR} \). Without loss of generality, \( L_i \neq \emptyset (1 \leq i \leq K) \) and all the state sets \( B^i \) \((1 \leq i \leq K)\) are pairwise disjoint. Let \( \hat{B} = B^1 \cup \cdots \cup B^K \) and let \( \hat{P} = P_{\mathcal{A}}, \cup \cdots \cup P_{\mathcal{A}} \). For each \( b \in \hat{B}, m \geq 1, \) and \( \sigma \in \Sigma_m \), consider the set of all rules in \( \hat{P} \) with left-hand side \( b(\sigma(x_1, \ldots, x_m)) \):

\[
\langle b(\sigma(x_1, \ldots, x_m)) \rightarrow u_1; L_{11}^{b_1}, \ldots, L_{1m}^{b_1} \rangle,
\vdots
\vdots
\langle b(\sigma(x_1, \ldots, x_m)) \rightarrow u_n; L_{n1}^{b_n}, \ldots, L_{nm}^{b_n} \rangle.
\]
where \( n \geq 0 \) depends on \( b \) and \( \sigma \), and define a set of triples:

\[
I_{ba} = \{ [i, j, k] | 1 \leq i \leq m, 1 \leq j < k \leq n, L_{ji}^{ba} \cap L_{ki}^{ba} = \emptyset \}.
\]

For each triple \([i, j, k] \in I_{ba}\), using Lemma 3.2, there are tree languages \( L_{ji}^{ba}, L_{ki}^{ba}, L_{jk}^{ba} \in DR \) such that

\[
L_{ji}^{ba} \subseteq L_{ji}^{ba}, \quad L_{ki}^{ba} \subseteq L_{ki}^{ba}
\]

with either the three languages \( L_{ji}^{ba}, L_{ki}^{ba}, L_{jk}^{ba} \) or the two languages \( L_{ji}^{ba}, L_{ki}^{ba} \) forming a partition of \( T_\Sigma \). Denote this partition by \( \Pi_{ijk}^{ba} \). For each \( m \geq 1 \), \( \sigma \in \Sigma_m \), and \( 1 \leq i \leq m \), let

\[
V_{\sigma i} = \bigwedge_{b \in \hat{B}} \bigwedge_{[i,j,k] \in I_{ba}} \Pi_{ijk}^{ba}.
\]

Thus, each \( V_{\sigma i} \) is a (finite) partition of \( T_\Sigma \) which satisfies the following property:

For each \( m \geq 1 \), \( \sigma \in \Sigma_m \), \( b \in \hat{B}, [i, j, k] \in I_{ba} \), and \( M \in V_{\sigma i} \), exactly one element of the partition \( \Pi_{ijk}^{ba} \) includes \( M \). (*)

It is well-known that if \( L_1, L_2 \in DR \) are tree language over \( \Sigma \), then \( L_1 \cap L_2 \in DR \).

Each tree language \( M \in V_{\sigma i} \) is an intersection of finitely many tree languages in \( DR \), hence each \( M \in V_{\sigma i} \) is in \( DR \).

Let \( \sigma \in \Sigma_m, m \geq 1 \). A function \( \phi_\sigma : \hat{B} \to \hat{P} \cup \{\#\} \) is said to be consistent if the left-hand side of \( \phi_\sigma(b) \) is \( b(\sigma(x_1, \ldots, x_m)) \) whenever \( \phi_\sigma(b) \in P \). For each \( \sigma \in \Sigma_m (m \geq 1) \), let \( \Phi_\sigma \) be the set of all consistent functions \( \phi_\sigma \). Define a ranked alphabet \( \Gamma \) such that \( \Gamma_0 = \Sigma_0 \) and for \( m \geq 1 \),

\[
\Gamma_m = \bigcup_{\sigma \in \Sigma_m} \Phi_\sigma.
\]

We now define a one-state \( t^{DR}_\hat{\Sigma} = \langle \Sigma, \Gamma, \{d\}, d, P_\varnothing \rangle \) by specifying its rules as follows.

(a) For all \( \sigma \in \Sigma_0 \), \( \langle d(\sigma) \to \sigma \rangle \) is in \( P_\varnothing \).

(b) Let \( \sigma \in \Sigma_m (m \geq 1) \) and \( M_i \in V_{\sigma i} (1 \leq i \leq m) \). We define the following function \( \psi_\sigma : \hat{B} \to \hat{P} \cup \{\#\} \). Let \( b \in \hat{B} \) and consider the set of rules (in \( \hat{P} \)) with left-hand side \( b(\sigma(x_1, \ldots, x_m)) \). If there exist rules: \( \langle b(\sigma(x_1, \ldots, x_m)) \to u_j ; L_{ji1}^{ba}, \ldots, L_{jim}^{ba} \rangle \) such that \( M_i \cap L_{ji}^{ba} \neq \emptyset \) for all \( 1 \leq i \leq m \), then pick one such rule, say \( r \), and set \( \psi_\sigma(b) = r \). Otherwise, set \( \psi_\sigma(b) = \# \). Obviously, \( \psi_\sigma \in \Phi_\sigma \). Put the rule

\[
\langle d(\sigma(x_1, \ldots, x_m)) \to \psi_\sigma(d(x_1), \ldots, d(x_m)); M_1, \ldots, M_m \rangle
\]

in \( P_\varnothing \).

The following two claims establish that \( \hat{\Sigma} \) is deterministic and \( dom(\tau_\varnothing) = T_\Sigma \).

**Claim B.** The \( t^{DR}_\hat{\Sigma} \) is deterministic.
Proof. Consider two different rules in $P_{\mathcal{G}}$ with the same left-hand side:

\[
\langle d(\sigma(x_1, \ldots, x_m)) \rightarrow \phi_{\sigma}(\sigma(x_1, \ldots, x_m)); M_1, \ldots, M_m \rangle,
\]

\[
\langle d(\sigma(x_1, \ldots, x_m)) \rightarrow \psi_{\sigma}(\sigma(x_1, \ldots, x_m)); M'_1, \ldots, M'_m \rangle.
\]

We need to show that these rules do not violate the requirement of determinism, i.e. that there exists some $l (1 \leq l \leq m)$ such that $M_l \cap M'_l = \emptyset$. First suppose that for some $i (1 \leq i \leq m)$, $M_i \neq M'_i$. Since $M_i, M'_i \in V_{\sigma_i}$, we must have $M_i \cap M'_i = \emptyset$, and in this case we are done, because we can choose $l = i$. Now suppose that the look-ahead sets in the two rules are the same: $M_i = M'_i (1 \leq l \leq m)$, but $\phi_{\sigma} \neq \psi_{\sigma}$. We will show that these assumptions lead to a contradiction. Let $b \in \hat{B}$ such that $\phi_{\sigma}(b) \neq \psi_{\sigma}(b)$ and let

\[
\phi_{\sigma}(b) = \langle b(\sigma(x_1, \ldots, x_m)) \rightarrow u_j, L^b_{j_1}, \ldots, L^b_{j_m} \rangle.
\]

Case 1: $\psi_{\sigma}(b) = \langle b(\sigma(x_1, \ldots, x_m)) \rightarrow u_k; L^b_{k_1}, \ldots, L^b_{k_m} \rangle$. Since the $\mathcal{G}'$'s are $\text{dta}^{DR}$, there exists an $i (1 < i < m)$ such that $L^b_{j_i} \cap L^b_{k_i} = \emptyset$. Then, we have

- $[i, j, k] \in \mathcal{I}_{\mathcal{G}}$,
- the partition $\Pi^b_{ijk}$ of $T_{\Sigma}$ has either the form $\Pi^b_{ijk} = \{L^{b_{jk}}, L^{k_{ij}}, L^{b_{ij}}\}$ or the form $\Pi^b_{ijk} = \{L^{b_{jk}}, L^{k_{ij}}\}$,
- $L^b_{j_i} \subseteq L^b_{k_l}$ and $L^b_{k_l} \subseteq L^b_{l_{ij}}$.

Since $M_i = M'_i \in V_{\sigma_i}$, by property (*), $M_i$ is a subset of exactly one element of the partition $\Pi^b_{ijk}$. Now, by the definition of $\mathcal{G}$, $M_i \cap L^b_{j_i} \neq \emptyset$ which implies $M_i \subseteq L^b_{j_i}$,

and $M'_i \cap L^b_{k_i} \neq \emptyset$ which implies $M'_i \subseteq L^b_{k_i}$. Since $M_i = M'_i$, we have a contradiction.

Case 2: $\psi_{\sigma}(b) = \emptyset$. In this case, on the one hand $M_i \cap L^b_{j_i} \neq \emptyset$ for all $i (1 < i < m)$, and on the other hand, by the definition of $\mathcal{G}$, $M_i \cap L^b_{j_i} = \emptyset$ for some $l (1 \leq l \leq m)$.

Since $M_i = M'_i$, we have a contradiction. \(\square\)

Claim C. For each tree $p \in T_{\Sigma}$, there is a unique tree $p' \in T_{\hat{T}}$ such that $d(p) = \#_p p'$.

Proof. First, by induction on the structure of trees, we show that $\text{dom}(\tau_{\mathcal{G}}) = T_{\Sigma}$. For $\sigma \in \Sigma_0$, by the construction of $\mathcal{G}$, $d(\sigma) \rightarrow \sigma \in P_{\mathcal{G}}$. Let $p = \sigma(p_1, \ldots, p_m)$ for some $\sigma \in \Sigma_m (m \geq 1)$. Since $V_{\sigma_i} (1 \leq i \leq m)$ is a partition of $T_{\Sigma}$, for each $i (1 \leq i \leq m)$ there exists $M_i \in V_{\sigma_i}$ such that $p_i \in M_i$. By the construction of $\mathcal{G}$, there exists a rule

\[
r = \langle d(\sigma(x_1, \ldots, x_m)) \rightarrow \phi_{\sigma}(\sigma(x_1, \ldots, x_m)); M_1, \ldots, M_m \rangle
\]

in $P_{\mathcal{G}}$ and hence

\[
d(p) = d(\sigma(p_1, \ldots, p_m))
\]

\[
\vdash_{\mathcal{G}} \phi_{\sigma}(d(p_1), \ldots, d(p_m)) \quad \text{by rule } r
\]

\[
\vdash_{\mathcal{G}} \phi_{\sigma}(p'_1, \ldots, p'_m) (p'_1, \ldots, p'_m \in T_{\hat{T}}) \quad \text{by induction hypothesis}
\]

Thus we have shown that $\text{dom}(\tau_{\mathcal{G}}) = T_{\Sigma}$, i.e. the existence part of the claim. Uniqueness follows from Claim B. \(\square\)
Proof of Theorem 3.3 (continued). Recall the dtt\textsuperscript{DR\textsubscript{i}} $\mathcal{A} = \langle \Sigma, \Delta, A, a_0, P_0 \rangle$ and its look-ahead sets $L_i = L(\mathcal{A}_i)$ where each $\mathcal{A}_i = \langle \Sigma, \Sigma, B, b_0, P_i \rangle$ is a dtt\textsuperscript{DR} ($1 \leq i \leq K$). We first define a dta $\mathcal{Q}_i$, corresponding to $\mathcal{A}_i$ ($1 \leq i \leq K$) and denote $L_i = L(\mathcal{Q}_i)$ (the formal relationship will be given in Claim D). Since the construction of $\mathcal{Q}_i$ (from $\mathcal{A}_i$) is rather involved (but uniform in $i$), and in order to avoid using multiple indices in the construction, we shall (temporarily) omit the index $i$ from our notation (and trust the reader's ability to reinsert it wherever necessary). Let $\mathcal{A} = \langle \Sigma, \Sigma, B, b_0, P_0 \rangle$ be one of the $\mathcal{A}_i$'s. Let $N_j = L(\mathcal{F}_j)$ ($1 \leq j \leq J$) be all the look-ahead sets appearing in the rules of $\mathcal{A}$, where $\mathcal{F}_j = \langle \Sigma, \Sigma, F_j, f_j, P_0 \rangle$ is a dta ($1 \leq j \leq J$). We define $\hat{F} = F^1 \cup \cdots \cup F^J$ and $\hat{P}_0 = P_0 \cup \cdots \cup P_0$, and assume, without loss of generality, that the state sets $F_j$ are pairwise disjoint. Since for any $f \in \hat{F}$ there is a unique $j$ ($1 \leq j \leq J$) such that $f \in F_j$, we shall write $f(p) = \hat{f}_j p$, without causing ambiguity, to mean $f(p) = \hat{f}_j p$ for the unique $j$ such that $f \in F_j$.

We now define the dta $\mathcal{G} = \langle \Gamma, \Gamma, G, g_0, P_0 \rangle$ corresponding to $\mathcal{A}$. The set of states and the initial state are

$$G = B \times P(\hat{F}) \quad \text{and} \quad g_0 = [b_0, \emptyset].$$

Intuitively, in the first component of its state the dta $\mathcal{G}$ simulates the computation of $\mathcal{A}$ while in its second component $\mathcal{G}$ simulates the computations of the look-ahead dtas $\mathcal{F}_1, \ldots, \mathcal{F}_J$. We proceed to give a formal definition of the rules of $\mathcal{G}$.

(a) For each $\sigma \in F_0$ ($= \Sigma_0$) and each $[b, S] \in G$, we put the rule

$$[b, S] (\sigma) \rightarrow \sigma$$

into $P_0$ if and only if $b(\sigma) \rightarrow \sigma$ is in $P_0$ and for each $f \in S$ there is a rule $f(\sigma) \rightarrow \sigma$ in $P_0$.

(b) Let $\phi_\sigma \in F_m$ ($m \geq 1$) and $[b, S] \in G$. We put the rule

$$[b, S](\phi_\sigma(x_1, \ldots, x_m)) \rightarrow \phi_\sigma([b_1, S_1](x_1), \ldots, [b_m, S_m](x_m))$$

in $P_0$ if and only if the following two conditions hold.

- $\phi_\sigma(b)$ is a rule in $P_0$, of the form:
  $$\langle h(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(h_1(x_1), \ldots, h_m(x_m)); N_{k_1}, \ldots, N_{k_m} \rangle$$

  where $1 \leq k_j \leq J$ ($1 \leq j \leq m$).

- For each $f \in S$ there is a rule of the form
  $$f(\sigma(x_1, \ldots, x_m)) \rightarrow tf$$

  in $\hat{P}_0$, and for each $j$ ($1 \leq j \leq m$)
  $$S_j = \{ f^{(k_j)} \} \cup \bigcup_{f \in S} \{ f' \in \hat{F} | f'(x_j) \text{ occurs in } tf \}.$$

This completes the definition of $\mathcal{G}$. The formal relationship between $\mathcal{A}$ and $\mathcal{G}$ is given in the following claim.
Claim D. Let $p \in T$ be a tree such that $d(p) \Rightarrow p' \in T$. Then

\[ p \in L(\mathcal{A}) \quad \text{if and only if} \quad p' \in L(\mathcal{A}). \]

Proof. Note that $p \in L(\mathcal{A})$ if and only if $b_0(p) \Rightarrow \mathcal{A} p$, and similarly, $p' \in L(\mathcal{A})$ if and only if $[b_0, \emptyset] (p') \Rightarrow \mathcal{A} p'$. Claim D will follow from the following, more general, statement.

For each $b \in B$, each $S \subseteq \mathcal{F}$, and $p \in T$ such that $d(p) = g p'$:

\[ b(p) \Rightarrow \mathcal{A} p' \quad \text{for each } f \in S \quad \text{if and only if} \quad [b, S](p') \Rightarrow \mathcal{A} p'. \quad (**) \]

(only if) This direction is proved by induction on the structure of trees. Suppose $p = \sigma \in \Sigma$ and that $b(\sigma) \Rightarrow \mathcal{A} \sigma$ and that for each $f \in S$, $f(\sigma) \Rightarrow \mathcal{A} \sigma$. Since both derivations must be one-step derivations we have $b(\sigma) \rightarrow \sigma \in P_\mathcal{A}$ and $f(\sigma) \rightarrow \sigma \in \hat{P}_\mathcal{A}$. By the definition of $\mathcal{A}$, $P_\mathcal{A}$ will include a rule $[b, S](\sigma) \rightarrow \sigma$. Since $p = \sigma = p'$, the proof of the base case is complete.

Now suppose $\sigma \in \Sigma_m (m \geq 1)$, $p = \sigma(p_1, \ldots, p_m)$, and let

\[ \langle b(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(b_1(x_1), \ldots, b_m(x_m)); \ N_1, \ldots, N_{k_m} \rangle \]

be the rule (of $P_\mathcal{A}$) applied in the first step of the derivation $b(p) \Rightarrow \mathcal{A} p$, where $1 \leq k_j \leq J (1 \leq j \leq m)$. Then $p_j \in N_k, (1 \leq j \leq m)$ and hence

\[ j^k_i(p_j) \Rightarrow \mathcal{A} p_j \quad (1 \leq j \leq m) \]

and

\[ b(\sigma(p_1, \ldots, p_m)) \Rightarrow \mathcal{A} (b_1(p_1), \ldots, b_m(p_m)) \Rightarrow \mathcal{A} (\sigma(p_1, \ldots, p_m)) \]

which implies

\[ b_j(p_j) \Rightarrow \mathcal{A} p_j \quad (1 \leq j \leq m). \]

(1')

Recall now the construction of $\mathcal{A}$. Let $M_j \in V_{sj}$ be such that $p_j \in M_j (1 \leq j \leq m)$. Then $\mathcal{A}$ has a rule

\[ \langle d(\sigma(x_1, \ldots, x_m)) \rightarrow \phi_\sigma(d(x_1), \ldots, d(x_m)); M_1, \ldots, M_m \rangle. \]

Since $p_j \in M_j \cap N_k, (1 \leq j \leq m)$, by Claim B,

\[ \phi_\sigma(b) = \langle b(\sigma(x_1, \ldots, x_m)) \rightarrow \sigma(b_1(x_1), \ldots, b_m(x_m)); N_1, \ldots, N_{k_m} \rangle. \]

It follows that the derivation $d(\sigma(p_1, \ldots, p_m)) \Rightarrow \mathcal{A} p'$ can be decomposed

\[ d(\sigma(p_1, \ldots, p_m)) \Rightarrow \mathcal{A} \phi_\sigma(d(p_1), \ldots, d(p_m)) \Rightarrow \mathcal{A} \phi_\sigma(p_1, \ldots, p_m) = p', \]

which implies

\[ d(p_j) \Rightarrow \mathcal{A} p'_j \quad (1 \leq j \leq m). \]

(2')
Now, since for each \( f \in S \), \( f(p) \Rightarrow \top p \), it follows that for each \( f \in S \) there is a unique
rule \( f(\sigma(x_1, \ldots, x_m)) \Rightarrow \top f \) in \( \hat{P}_\mathcal{G} \). For each \( j (1 \leq j \leq m) \) define
\[ S_j = \{ f_0^{b_j} \} \cup \bigcup_{f \in S} \{ f' \in \hat{F} | f'(x_j) \text{ occurs in } f \} \].

By (1) and by the definition of \( S_j \) we have that for each \( j (1 \leq j \leq m) \) and each \( f \in S_j \)
\[ f(p_j) \Rightarrow \top p_j \].

By the definition of rules of \( \mathcal{G} \), the rule
\[ [b, S](\phi_\sigma(x_1, \ldots, x_m)) \Rightarrow \phi_\sigma([b_1, S_1](x_1), \ldots, [b_m, S_m](x_m)) \]
is in \( P_\mathcal{G} \). From (2'), (1'), and (3) it follows, by induction hypothesis, that for each \( j (1 \leq j \leq m) \)
\[ [b_j, S_j](p_j) \Rightarrow \top p_j \].

Hence,
\[ [b, S](p') = [b, S](\phi_\sigma(p_1', \ldots, p_m')) \]
\[ \Rightarrow \top \phi_\sigma([b_1, S_1](p_1'), \ldots, [b_m, S_m](p_m')) \] by (4)
\[ \Rightarrow \top \phi_\sigma(p_1', \ldots, p_m') \] by (5)
\[ = p' \].

(If') This direction is also proved by induction on the structure of trees. Suppose that \( p = \sigma \in \Sigma_0 \) and let \( b \in B, S \subseteq \hat{F} \) be such that
\[ d(p) \Rightarrow \top p' \] and \[ [b, S](p') \Rightarrow \top p' \].

In this case both derivations are one-step derivations, \( p = p' = \sigma \), and so \([b, S](\sigma) \Rightarrow \top \sigma \)
is a rule of \( P_\mathcal{G} \). By the definition of \( \mathcal{G} \), this implies that \( b(\sigma) \Rightarrow \sigma \) is a rule in \( P_\mathcal{G} \) and that for each \( f \in S \) there is a rule \( f(\sigma) \Rightarrow \top f \) in \( \hat{P}_\mathcal{G} \). This shows that the implication holds in
the base case.

Now suppose that \( \sigma \in \Sigma_m \) \((m \geq 1)\) and \( p = \sigma(p_1, \ldots, p_m) \). Assume that
\[ d(p) = d(\sigma(p_1, \ldots, p_m)) \Rightarrow \top \phi_\sigma(d(p_1), \ldots, d(p_m)) \Rightarrow \top \phi_\sigma(p_1', \ldots, p_m') = p' \]
and
\[ [b, S](p') = [b, S](\phi_\sigma(p_1', \ldots, p_m')) \Rightarrow \top \phi_\sigma([b_1, S_1](p_1'), \ldots, [b_m, S_m](p_m')) \]
\[ \Rightarrow \top \phi_\sigma(p_1', \ldots, p_m') = p' \]
where
\[ \phi_\sigma(b) = \langle b(\sigma(x_1, \ldots, x_m)) \Rightarrow \sigma(b_1(x_1), \ldots, b_m(x_m)); N_1, \ldots, N_k \rangle \]
is a rule in $P_{\mathcal{G}}$. From (7), by the definition of rules in $P_{\mathcal{G}}$, we have for each $f \in S$,

$$f(\sigma(x_1, \ldots, x_m)) \rightarrow t^f$$

is in $\hat{F}_\mathcal{G}$

and for each $j (1 \leq j \leq m)$

$$S_j = \{ f_0^j \} \cup \bigcup_{j \in S} \{ f' \in \hat{F} \mid f'(x_j) \text{ occurs in } t^f \}.$$  

(Note that $t^f$ is of the form $\sigma(f_1(x_1), \ldots, f_m(x_m))$.) Now, for each $j (1 \leq j \leq m)$ we obtain from (6)

$$d(p_j) \Rightarrow_{\mathcal{G}} p_j,'$$  

and from (7)

$$[b_j, S_j](p_j) \Rightarrow_{\mathcal{G}} p_j.'$$

By induction hypothesis we have for each $j (1 \leq j \leq m) b_j(p_j) \Rightarrow_{\mathcal{G}} p_j,$ and for each $f' \in S_j (1 \leq j \leq m) f'(p_j) \Rightarrow_{\mathcal{G}} p_j.$ Thus, $p_j \in N_{k_j}$ for $1 \leq j \leq m$. Hence, we have

$$b(p) = b(\sigma(p_1, \ldots, p_m)) \Rightarrow_{\mathcal{G}} \sigma(b_1(p_1), \ldots, b_m(p_m)) \Rightarrow_{\mathcal{G}} \sigma(p_1, \ldots, p_m) = p$$

and for each $f \in S$

$$f(p) = f(\sigma(p_1, \ldots, p_m)) \Rightarrow_{\mathcal{G}} t^f[p_1, \ldots, p_m] \Rightarrow_{\mathcal{G}} \sigma(p_1, \ldots, p_m) = p. \quad \square$$

**Proof of Theorem 3.3 (continued).** Define the dtt$^{DR}$ $\mathcal{G} = \langle \Gamma, A, \mathcal{A}, a_0, P_{\mathcal{G}} \rangle$ as follows. For each $a \in A, m \geq 1, \sigma \in \Sigma_m, \phi_a \in \Gamma_m$, the rule

$$\langle a(\phi_a(x_1, \ldots, x_m)) \rightarrow q; Z_1', \ldots, Z_m' \rangle \in P_{\mathcal{G}}$$

if and only if

$$\langle a(\sigma(x_1, \ldots, x_m)) \rightarrow q; Z_1, \ldots, Z_m \rangle \in P_{\mathcal{G}}$$

where for each $1 \leq j \leq m, Z_j'$ corresponds to $Z_j$ via the construction presented above, i.e. $Z_j' = L(\Theta_k) = L_k$ if and only if $Z_j = L(\Theta_k) = L_k$. Moreover, for each $a \in A, \sigma \in \Sigma_0$, the rule

$$\langle a(\sigma) \rightarrow q; \rangle \in P_{\mathcal{G}} \quad \text{if and only if} \quad \langle a(\sigma) \rightarrow q; \rangle \in P_{\mathcal{G}}.$$

**Claim E.** $\tau_{\mathcal{G}'} = \tau_{\mathcal{G}} \circ \tau_{\mathcal{G}}$.

**Proof.** It is sufficient to show that for each $a \in A, p \in T_{\Sigma}, p' \in T_{\Gamma}$ with $d(p) \Rightarrow p'$, and $q \in T_{\mathcal{A}}$, the following equivalence holds:

$$a(p) \Rightarrow_{\mathcal{G}} q \quad \text{if and only if} \quad a(p') \Rightarrow_{\mathcal{G}} q.$$  

(***
(only if) We proceed by induction on the structure of trees. Suppose that \( p \in \Sigma_0 \). Then the rule \( \langle a(p) \rightarrow q; \rangle \) is in \( P_{df} \) and so the rule \( d(p) \rightarrow p \) is in \( P_{\delta} \). Hence \( \langle a(p) \rightarrow q; \rangle \) is in \( P_{df} \).

Now suppose that \( m \geq 1 \), \( \sigma \in \Sigma_m \), and \( p = \sigma(p_1, \ldots, p_m) \). Then the derivation \( a(p) \Rightarrow^*_{df} q \) can be written in the following way:

\[
\begin{align*}
a(p) &= a(\sigma(p_1, \ldots, p_m)) \Rightarrow^*_{df} q_0[a_1(p_{i_1}), \ldots, a_k(p_{i_k})] \Rightarrow^*_{df} q_0[q_1, \ldots, q_k] = q,
\end{align*}
\]

where

\[
\langle a(\sigma(x_1, \ldots, x_m)) \rightarrow q_0[a_1(x_{i_1}), \ldots, a_k(x_{i_k})]; Z_1, \ldots, Z_m \rangle \in P_{df},
\]

and \( p_1 \in Z_1, \ldots, p_m \in Z_m \), \( q_0 \in T_{\alpha}(X_k) \), \( k \geq 0 \), \( q_1, \ldots, q_k \in T_{\alpha} \). Moreover, for each \( 1 \leq j \leq k \),

\[
a_j(p_{i_j}) \Rightarrow^*_{df} q_j.
\]

Consider the derivation

\[
d(\sigma(p_1, \ldots, p_m)) \Rightarrow_{\phi} \phi_{\sigma}(d(p_1), \ldots, d(p_m)) \Rightarrow^*_{\phi} \phi_{\sigma}(p'_1, \ldots, p'_m)
\]

where for \( 1 \leq j \leq m \), \( d(p_j) \Rightarrow^*_{\phi} p'_j \). By the construction of \( \phi \), the rule

\[
\langle a(\phi_{\sigma}(x_1, \ldots, x_m)) \rightarrow q_0[a_1(x_{i_1}), \ldots, a_k(x_{i_k})]; Z_1, \ldots, Z_m \rangle
\]

is in \( P_{\phi} \), where for each \( 1 \leq j \leq m \), \( Z_j \) corresponds to \( Z_j \) as explained above. By Claim D, \( p'_j \in Z_j \) (\( 1 \leq j \leq m \)). Thus

\[
a(p') = a(\phi_{\sigma}(p'_1, \ldots, p'_m)) \Rightarrow_{\phi} q_0[a_1(p'_{i_1}), \ldots, a_k(p'_{i_k})].
\]

By the induction hypothesis, for each \( 1 \leq j \leq k \), \( a_j(p'_j) \Rightarrow^*_{\phi} q_{ij} \), and hence

\[
a(p') = a(\phi_{\sigma}(p'_1, \ldots, p'_m)) \Rightarrow_{\phi} q_0[a_1(p'_{i_1}), \ldots, a_k(p'_{i_k})] \Rightarrow^*_{\phi} q_0[q_1, \ldots, q_k] = q.
\]

(if) We proceed by induction on the structure of trees. Let \( p \in \Sigma_0 \). Then the rule \( \langle d(p) \rightarrow p; \rangle \) is in \( P_{\phi} \), hence \( p' = p \) and \( \langle a(p) \rightarrow q; \rangle \) is in \( P_{df} \). By the construction of \( \phi \), the rule \( \langle a(p) \rightarrow q; \rangle \) is in \( P_{df} \).

Now suppose that \( m \geq 1 \), \( \sigma \in \Sigma_m \), \( p = \sigma(p_1, \ldots, p_m) \), \( d(p) \Rightarrow^*_{\phi} p' \), and \( a(p') \Rightarrow^*_{\phi} q \). Then the derivation \( d(p) \Rightarrow^*_{\phi} p' \) can be written in the following way:

\[
d(p) = d(\sigma(p_1, \ldots, p_m)) \Rightarrow_{\phi} \phi_{\sigma}(d(p_1), \ldots, d(p_m)) \Rightarrow^*_{\phi} \phi_{\sigma}(p'_1, \ldots, p'_m) = p'.
\]
where for \(1 \leq j \leq m\), \(d(p_j) \Rightarrow \delta{p}_j\). Similarly, the derivation \(a(p') \Rightarrow \delta q\) can be written in the following way:

\[
a(p') = a(\phi(p_1', \ldots, p_m')) \Rightarrow \delta q_0[a_1(p_1'), \ldots, a_k(p_k')] \Rightarrow \delta q_0[q_1, \ldots, q_k] = q,
\]

where the rule

\[
\langle a(\phi(x_1, \ldots, x_m)) \Rightarrow q_0[a_1(x_1'), \ldots, a_k(x_k')] ; Z'_1, \ldots, Z'_m \rangle
\]

is in \(P\). Moreover, for each \(1 \leq j \leq m\), \(p_j') \in Z_j\), and for each \(1 \leq j \leq k\), \(a_j(p_j') \Rightarrow \delta q_j\).

By the construction of \(\delta\), the rule

\[
\langle a(\sigma(x_1, \ldots, x_m)) \Rightarrow q_0[a_1(x_1'), \ldots, a_k(x_k')] ; Z_1, \ldots, Z_m \rangle \in P_{\alpha'},
\]

where for each \(1 \leq j \leq m\), \(Z_j')\) corresponds to \(Z_j\) via the construction presented above. By Claim D, \(p_j \in Z_j\) for \(1 \leq j \leq m\). By the induction hypothesis, for each \(1 \leq j \leq k\),

\[
a_j(p_j') \Rightarrow \delta q_j.
\]

Hence

\[
a(p) = a(\sigma(p_1, \ldots, p_m)) \Rightarrow \delta q_0[a_1(p_1'), \ldots, a_k(p_k')] \Rightarrow \delta q_0[q_1, \ldots, q_k] = q. \quad \square
\]

This completes the proof of Theorem 3.3. \(\square\)

3.2. The Case \(n \geq 2\) (Part B)

**Theorem 3.4.** For each \(n \geq 2\), \(DTT^{DR_n} \subseteq DTT^{DR} \circ DTT^{DR_{n-1}}\).

**Proof.** Intuitively, for a given \(\text{dtt}^{DR_n} \mathcal{A}\), we define \(\text{dtt}^{DR} \mathcal{D}\) and \(\text{dtt}^{DR_{n-1}} \mathcal{A}'\) such that \(\tau_{\mathcal{A}'} = \tau_{\mathcal{A}} \circ \tau_{\mathcal{A}'}\). We "unfold" the look-ahead sets of \(\mathcal{A}\) and construct \(\mathcal{D}\) simultaneously for all look-ahead \(\text{dtt}^{DR_n}\)'s of \(\mathcal{A}\) similarly to the case \(n = 1\). As in the case \(n = 1\), a look-ahead \(\text{dtt}^{DR_1}\) of \(\mathcal{A}\) can be simulated by the composition of \(\mathcal{D}\) and a \(\text{dtt}^{DR}\). Hence \(\mathcal{A}'\) simulates \(\mathcal{A}\) on the output provided by \(\mathcal{D}\) roughly as follows: the look-ahead \(\text{dtt}^{DR_1}\)'s of \(\mathcal{A}'\) simulate the look-ahead \(\text{dtt}^{DR_1}\)'s of \(\mathcal{A}\), and hence the look-ahead \(\text{dtt}^{DR_1}\)'s of \(\mathcal{A}'\) simulate the look-ahead \(\text{dtt}^{DR_2}\)'s of \(\mathcal{A}, \ldots, \), and finally, the look-ahead \(\text{dtt}^{DR_{n-1}}\)'s of \(\mathcal{A}'\) simulate the look-ahead \(\text{dtt}^{DR_n}\)'s of \(\mathcal{A}\). Thus the rules of \(\mathcal{A}'\) are obtained from those of \(\mathcal{A}\) by substituting look-ahead sets in \(DR_{n-1}\) for look-ahead sets in \(DR_n\). The details of the construction follow.

Let \(\mathcal{A} = \langle \Sigma, A, A_0, R_{\mathcal{A}} \rangle\) be a \(\text{dtt}^{DR_n}\). For each \(i (1 \leq i \leq n)\) define a finite set of languages \(V_i \subseteq DR_i\), and a corresponding (equinumerous) set of automata \(W_i \subseteq DTA^{DR_{n-1}}\) as follows. Let \(V_n \subseteq DR_n\) be the set of look-ahead languages of \(\mathcal{A}\) and let \(W_n\) be a set of \(\text{dtt}^{DR_{n-1}}\)'s, one for each language in \(V_n\), such that

\[
V_n = \{ L(\mathcal{B}) \mid \mathcal{B} \in W_n \}.
\]
Proceeding inductively, suppose $V_{i+1}$ and $W_{i+1}$ have been defined. Let $V_i \subseteq DR_i$ be the set of all look-ahead languages in the tree automata $\mathcal{A} \in W_{i+1}$, and let $W_i \subseteq DTATR_{i+1}$ be a set of tree automata for languages in $V_i$, one for each language in $V_i$. We have thus defined finite sets of languages: $V_n, V_{n-1}, \ldots, V_2, V_1$, together with the corresponding sets of acceptors: $W_n, W_{n-1}, \ldots, W_2, W_1$.

Now let $W_1 = \{ \mathcal{A}_i | 1 \leq i \leq K \}$ and $V_1 = \{ L_i | L_i = L(\mathcal{A}_i), \mathcal{A}_i \in W_1 \}$, where $\mathcal{A}_i = \langle \Sigma, \Sigma, B_i, b_0, P_i \rangle$ is a dta$^{DR}$. Without loss of generality we may assume that $L_i \neq \emptyset (1 \leq i \leq K)$ and that the state sets $B_i (1 \leq i \leq K)$ are pairwise disjoint. Let $\hat{B} = B_1 \cup \ldots \cup B_K$ and $\hat{P} = P_1 \cup \ldots \cup P_K$.

From $\hat{B}$ and $\hat{P}$, define a tt$^{DR}$ $\mathcal{D} = \langle \Sigma, \Gamma, \{ d \}, d, P_{\mathcal{D}} \rangle$ using exactly the same construction as in Part A. The only difference is in the definition of $\hat{B}$ and $\hat{P}$: in Part A, $\mathcal{D}$ was defined from the look-ahead sets of a single dta$^{DR}$, while now $\mathcal{D}$ is defined from all the look-ahead sets of all the dta$^{DR}$'s in $W_2$. It follows, exactly as in Part A, that $\mathcal{D}$ is total and deterministic. That is, Claims B and C hold for (the new) $\mathcal{D}$.

Now, for each dta$^{DR}$ $\mathcal{A}_i = \langle \Sigma, \Sigma, H, h_0, P_{\mathcal{A}_i} \rangle \in W_2$, with $N = L(\mathcal{A}_i) \in V_2$, we shall define a dta$^{DR}$ $\mathcal{E}_{\mathcal{A}_i} = \langle \Gamma, \Gamma, H, h_0, P_{\mathcal{A}_i} \rangle$. The construction is very similar to the construction of the dta$^{DR}$ $\mathcal{E}$ from the tt$^{DR}$ $\mathcal{A}$ in Part A. First, for each look-ahead set $L_i = L(\mathcal{A}_i) \in V_1$ define a dta $\mathcal{A}_i$, with $L_i = L(\mathcal{A}_i)$, exactly as in Part A. Thus each pair $\mathcal{A}_i, \mathcal{G}_i (1 \leq i \leq K)$ satisfies Claim D in Part A. Now, the rules of $\mathcal{E}_{\mathcal{A}_i}$ are built as follows. For each $h \in H, m \geq 1, \sigma \in \Sigma_m, \phi_\sigma \in \Gamma_m$, the rule

$$\langle h(\phi_\sigma(x_1, \ldots, x_m)) \rangle \rightarrow \phi_\sigma(h_1(x_1), \ldots, h_m(x_m)); Z_1, \ldots, Z_m \rangle \in P_{\mathcal{E}_{\mathcal{A}_i}}$$

if and only if

$$\langle h(\sigma(x_1, \ldots, x_m)) \rangle \rightarrow \sigma(h_1(x_1), \ldots, h_m(x_m)); Z_1, \ldots, Z_m \rangle \in P_{\mathcal{A}_i},$$

where for each $1 \leq j \leq m$, $Z_j$ corresponds to $Z_j$ via the requirement: $Z_j = L(\mathcal{A}_k) = L_k$ if and only if $Z_j = L(\mathcal{A}_k) = L_k$ (exactly as in Part A). Moreover, for each $h \in H, \sigma \in \Sigma_0 = \Gamma_0$, we have

$$\langle h(\sigma) \rightarrow \sigma \rangle \in P_{\mathcal{E}_{\mathcal{A}_i}}$$

if and only if $\langle h(\sigma) \rightarrow \sigma \rangle \in P_{\mathcal{A}_i}$.

The correspondence between $\mathcal{H}$ and $\mathcal{E}_{\mathcal{A}_i}$ is formally expressed in the following claim, analogous to (***) in the proof of Claim E, in Part A.

**Claim F.** For each $p \in T_\Sigma$ with $d(p) \Rightarrow_{\mathcal{F}} p' \in T_\Gamma$, the following equivalence holds:

$$p \in L(\mathcal{H}) \iff p' \in L(\mathcal{E}_{\mathcal{A}_i}).$$

**Proof.** It suffices to prove the following, more general, statement: for every $h \in H, p \in T_\Sigma$ with $d(p) \Rightarrow_{\mathcal{F}} p' \in T_\Gamma$, the equivalence

$$h(p) \Rightarrow_{\mathcal{E}_{\mathcal{A}_i}} p$$

if and only if $h(p') \Rightarrow_{\mathcal{E}_{\mathcal{A}_i}} p'$.

This is proved in the same as equivalence (*** in Part A). □
Proof of Theorem 3.4 (continued). We are now almost ready to define the \( d\text{tt}^{DR_{n-1}} \) such that

\[
\tau'_{\mathcal{A}'} = \tau'_{\mathcal{A}} \circ \tau'_{\mathcal{A}}.
\]

Before specifying the rules of \( \mathcal{A}' \) we need to define its set of look-ahead languages. We will define sets \( W_2, \ldots, W_n \) of tree automata and corresponding sets \( V_2, \ldots, V_n \) of languages such that for \( 2 \leq i \leq n \):

(a) \( V_i' = \{ L(\mathcal{H}') | \mathcal{H}' \in W_i \} \),
(b) \( W_i' \subseteq DTAD^{DR_{i-2}} \) (and hence \( V_i' \subseteq DR_{i-1} \)),
(c) \( |W_i'| = |V_i'| = |V_i'| = |W_i'| \).

The set \( W_i' \) will be constructed inductively from \( W_{i-1}' \) and \( W_i \). The specific bijection between \( W_i \) and \( W_i' \) is denoted by \( \alpha_i \). For \( i = 2 \), define

\[
W_2' = \{ \mathcal{H}' \mid \mathcal{H}' \in W_2 \} \quad \text{and} \quad V_2' = \{ L(\mathcal{H}') \mid \mathcal{H}' \in W_2 \}.
\]

Clearly \( W_2' \subseteq DTAD^{DR_0} = DTAD^{DR} \), and the conditions (a)–(c) hold. In this case the bijection \( \alpha_2 \) assigns \( \mathcal{H}' \) to \( \mathcal{H} \). Proceeding inductively, let \( 3 \leq i \leq n-1 \), and assume that \( W_{i-1}' \subseteq DTAD^{DR_{i-3}} \) and \( V_{i-1}' \subseteq DR_{i-2} \) have been defined such that conditions (a)–(c) hold and bijection \( \alpha_{i-1} \) assigns \( \mathcal{H}' \) to \( \mathcal{H} \in W_{i-1} \). For each \( d\text{ta}^{DR_{i-1}} \mathcal{H} = \langle \Sigma, \Sigma, H, h_0, P \rangle \in W_{i-1} \) we define a corresponding \( d\text{ta}^{DR_{i-2}} \mathcal{H}' = \langle \Gamma, \Gamma, H, h_0, P' \rangle \in W_i \) as follows. For each \( h \in H, \sigma \in \Sigma_0 \), let

\[
h(\sigma) = \sigma \in P \quad \text{if and only if} \quad h(\sigma) = \sigma \in P',
\]

and for each \( h \in H, m \geq 1, \sigma \in \Sigma_m, \phi_\sigma \in \Gamma \), let

\[
\langle h(\phi_\sigma(x_1, \ldots, x_m)) \rangle \rightarrow \phi_\sigma(h_1(x_1), \ldots, h_m(x_m)); Y_1, \ldots, Y_m \in P'
\]

if and only if

\[
\langle h(\sigma(x_1, \ldots, x_m)) \rangle \rightarrow \sigma(h_1(x_1), \ldots, h_m(x_m)); Y_1, \ldots, Y_m \in P,
\]

where for each \( j \) (\( 1 \leq j \leq m \)), \( Y_j = L(\mathcal{H}_j) \) with \( \mathcal{H}_j \in W_{i-1} \), \( Y_j' = L(\mathcal{H}_j') \) with \( \mathcal{H}_j' \in W_{i-1} \), and \( \mathcal{H}_j' \) corresponds to \( \mathcal{H}_j \) under bijection \( \alpha_{i-1} \). This completes the construction of \( \mathcal{H}' \). We define

\[
W_i' = \{ \mathcal{H}' \mid \mathcal{H}' \in W_i \} \quad \text{and} \quad V_i' = \{ L(\mathcal{H}') \mid \mathcal{H}' \in W_i \},
\]

and the bijection \( \alpha_i \) between \( W_i \) and \( W_i' \), assigns \( \mathcal{H}' \) to \( \mathcal{H} \). The relationship between \( \mathcal{H}' \in W_i' \) and \( \mathcal{H} \in W_i \), which correspond to each other via \( \alpha_i \), is expressed in the following claim.

Claim G. Let \( p \in T_\Sigma \) and \( d(p) = \#_\mathcal{H} p' \in T_\Gamma \). Then for each \( 2 \leq i \leq n \), if \( \mathcal{H} \in W_i \) and \( \mathcal{H} \in W_i' \) correspond via bijection \( \alpha_i \), then

\[
p \in L(\mathcal{H}) \quad \text{if and only if} \quad p' \in L(\mathcal{H}').
\]
Proof. It suffices to prove the following, more general equivalence: for each \( p \in T_{\Sigma}, h \in H \),
\[
   h(p) \xrightarrow{\mathcal{R}} p \text{ if and only if } h(p') \xrightarrow{\mathcal{R}} p'.
\]

We prove this equivalence by induction on \( i \). For \( i = 2 \), the equivalence follows from Claim F. Suppose that \( 3 \leq i \leq n \) and that the equivalence holds for \( i - 1 \).

For the only-if-direction we proceed by induction on the structure of trees in \( T_{\Sigma} \).

For \( p \in \Sigma_0 \), the implication follows by the construction of \( \mathcal{H'} \). Let \( p = \sigma(p_1, \ldots, p_m) \) for some \( \sigma \in \Sigma_m \). Then the computation of \( \mathcal{D} \) on \( p \) is of the form
\[
   d(p) = d(\sigma(p_1, \ldots, p_m)) \Rightarrow \sigma(h_1(p_1), \ldots, h_m(p_m)) \xrightarrow{\mathcal{R}} \sigma(p_1, \ldots, p_m) = p,
\]
where for each \( j (1 \leq j \leq m) \), \( d(p_j) \Rightarrow \mathcal{R} p_j \). Consider the derivation
\[
   h(p) = h(\sigma(p_1, \ldots, p_m)) \Rightarrow \sigma(h_1(p_1), \ldots, h_m(p_m)) \xrightarrow{\mathcal{R}} \sigma(p_1, \ldots, p_m) = p,
\]
where the rule
\[
   \langle h(\sigma(x_1, \ldots, x_m)) \rangle \Rightarrow \sigma(h_1(x_1), \ldots, h_m(x_m)); \ Y_1, \ldots, Y_m \rangle \in P
\]
is applied in the first step of the derivation, and for each \( j (1 \leq j \leq m) \), \( p_j \in Y_j \) and
\[
   h_j(p_j) \Rightarrow \mathcal{R} p_j. \quad \text{Then, by the construction of } \mathcal{H'},
\]
\[
   \langle h(\phi_\sigma(x_1, \ldots, x_m)) \rangle \Rightarrow \phi_\sigma(h_1(x_1), \ldots, h_m(x_m)); \ Y_1, \ldots, Y_m \rangle \in P',
\]
where for each \( j (1 \leq j \leq m) \), \( Y_j = L(\mathcal{H}_j) \) with \( \mathcal{H}_j \in W_{i-1} \), \( Y_j = L(\mathcal{H}_j') \) with \( \mathcal{H}_j' \in W_{i-1} \), and \( \mathcal{H}_j \) and \( \mathcal{H}_j' \) correspond via the bijection \( \alpha_{i-1} \). By induction hypothesis, \( p_j \in Y_j \) and \( h_j(p_j) \Rightarrow \mathcal{R} p_j \ (1 \leq j \leq K) \), hence we have the following derivation in \( \mathcal{H}' \):
\[
   h(p') = h(\phi_\sigma(p_1', \ldots, p_m')) \Rightarrow \phi_\sigma(h_1(p_1'), \ldots, h_m(p_m')) \xrightarrow{\mathcal{R}} \phi_\sigma(p_1', \ldots, p_m') = p'.
\]

The if-direction proceeds by induction on the structure of trees in \( T_\Gamma \). For \( p' \in \Gamma_0 \), the implication follows from the definition of \( \mathcal{H}' \). Let \( p' = \phi_\sigma(p_1', \ldots, p_m') \) for \( m \geq 1 \). Consider the derivation
\[
   h(p') = h(\phi_\sigma(p_1', \ldots, p_m')) \Rightarrow \phi_\sigma(h_1(p_1'), \ldots, h_m(p_m')) \xrightarrow{\mathcal{R}} \phi_\sigma(p_1', \ldots, p_m'),
\]
where \( h_j(p_j) \Rightarrow \mathcal{R} p_j \ (1 \leq j \leq m) \), the rule
\[
   \langle h(\phi_\sigma(x_1, \ldots, x_m)) \rangle \Rightarrow \phi_\sigma(h_1(x_1), \ldots, h_m(x_m)); \ Y_1, \ldots, Y_m \rangle \in P'
\]
is applied in the first step, and \( p_j \in Y_j \ (1 \leq j \leq m) \). Let \( p = \sigma(p_1, \ldots, p_m) \in T_{\Sigma} \) such that \( d(p) \Rightarrow \mathcal{D} p' \) and therefore \( d(p_j) \Rightarrow \mathcal{D} p_j \ (1 \leq j \leq m) \). By induction hypothesis,
Proof of Theorem 3.4 (continued). We will now define the rules of $\mathcal{A}' = \langle \Gamma, A, A, a_0, P_{\mathcal{A}'} \rangle$. For each $a \in A$, $\sigma \in \Sigma_0$, and $q \in T_{\mathcal{A}}$, we let $\langle a(\sigma) \rightarrow q_i \rangle \in P_{\mathcal{A}'}$ if and only if $\langle a(\sigma) \rightarrow q_i \rangle \in P_{\mathcal{A}}$. For each $a \in A$, $\sigma \in \Sigma_m (m \geq 1)$, and $q \in T_{\mathcal{A}} (A(T_\Sigma(X_m)))$,

$$\langle a(\sigma(x_1, \ldots, x_m)) \rightarrow q; Y_1, \ldots, Y_m \rangle \in P_{\mathcal{A}}$$

if and only if

$$\langle a(\sigma(x_1, \ldots, x_m)) \rightarrow q; Y_1, \ldots, Y_m \rangle \in P_{\mathcal{A}'}$$

where for each $1 \leq j \leq m$, $Y_j = L(\mathcal{H}_j)$ with $\mathcal{H}_j \in W_{m-1}$, $Y_j' = L(\mathcal{H}_j')$ with $\mathcal{H}_j' \in W_j$, and the tree automata $\mathcal{H}_j$ and $\mathcal{H}_j'$ correspond via the bijection $x_{m-1}$. Hence,

$$h(p) = h(\sigma(p_1, \ldots, p_m)) \Rightarrow \sigma(h_1(p_1), \ldots, h_m(p_m)) \Rightarrow \sigma(p_1, \ldots, p_m) = p. \square$$

Proof. It is suffices to prove that for each $a \in A$, $p \in T_\Sigma$ with $d(p) \Rightarrow p' \in T_{\mathcal{A}}$, and $q \in T_{\mathcal{A}}$, the equivalence

$$a(p) \Rightarrow q \quad \text{if and only if} \quad a(p') \Rightarrow q$$

holds. The if-direction is proved by induction on the structure of trees. For $p \subset \Sigma_k$, the implication follows immediately from the definition of $\mathcal{A}'$. Let $m \geq 1$, $\sigma \in \Sigma_m$, $p = \sigma(p_1, \ldots, p_m)$, and consider the derivation

$$a(p) = a(\sigma(p_1, \ldots, p_m)) \Rightarrow_{\mathcal{A}} t[a_1(p_{i_1}), \ldots, a_k(p_{i_k})] \Rightarrow_{\mathcal{A}} t[q_{i_1}, \ldots, q_{i_k}] = q,$$

where the rule

$$\langle a(\sigma(x_1, \ldots, x_m)) \rightarrow t[a_1(x_{i_1}), \ldots, a_k(x_{i_k})]; Y_1, \ldots, Y_m \rangle \in P_{\mathcal{A}}$$

with $t \in T_\Sigma(X_k)$, $k \geq 0$, $a_1, \ldots, a_k \in A$, is applied in the first step. Moreover, for each $1 \leq j \leq m$, $p_j \in Y_j$, and for each $1 \leq j \leq k$, $a_j(p_{i_j}) \Rightarrow_{\mathcal{A}} q_{i_j}$, Consider the derivation of $\mathcal{G}$ on $p$:

$$d(p) = d(\sigma(p_1, \ldots, p_m)) \Rightarrow q \phi(d(p_1), \ldots, d(p_m)) \Rightarrow q \phi(p_1', \ldots, p_m') = p'.$$
By the construction of $\mathcal{A}'$,

$$\langle a(\phi_\sigma(x_1, \ldots, x_m)) \rightarrow t[a_1(x_{i_1}), \ldots, a_k(x_{i_k})]; Y'_1, \ldots, Y'_m \rangle \in P_{\mathcal{A}'}$$

where for each $j (1 \leq j \leq m)$, $Y_j = L(H'_j)$ with $H'_j \in W'_n$, $Y'_j = L(H''_j)$ with $H''_j \in W'_n$, and the automata $H'_j$ and $H''_j$ correspond via the bijection $\alpha_n$. By Claim G, $p_j \in Y'_j$ (1 ≤ j ≤ m). By induction hypothesis, for each 1 ≤ j ≤ k, $a_j(p'_j) \Rightarrow_{\mathcal{A}'} q_{ij}$. It follows that

$$a(p') = a(\phi_\sigma(p'_1, \ldots, p'_m)) \Rightarrow_{\mathcal{A}'} t[a_1(p'_{i_1}), \ldots, a_k(p'_{i_k})] \Rightarrow_{\mathcal{A}'} t[q_{i_1}, \ldots, q_{i_k}] = q.$$ 

The if-direction is also proved by induction on the structure of trees. For $p' \in \Gamma_0(= \Sigma_0)$, the implication follows immediately from the construction of $\mathcal{A}'$. Let $m \geq 1$, $\phi_\sigma \in \Gamma_m$ (hence $\sigma \in \Sigma_m$), $p' = \phi_\sigma(p'_1, \ldots, p'_m)$, and consider the derivation:

$$a(p') = a(\phi_\sigma(p'_1, \ldots, p'_m)) \Rightarrow_{\mathcal{A}'} t[a_1(p'_{i_1}), \ldots, a_k(p'_{i_k})] \Rightarrow_{\mathcal{A}'} t[q_{i_1}, \ldots, q_{i_k}] = q,$$

where the rule

$$\langle a(\phi_\sigma(x_1, \ldots, x_m)) \rightarrow t[a_1(x_{i_1}), \ldots, a_k(x_{i_k})]; Y'_1, \ldots, Y'_m \rangle \in P_{\mathcal{A}'}$$

with $t \in T_A(X_k)$, $k \geq 0$, $a_1, \ldots, a_k \in A$, is applied in the first step. It follows that for each $j (1 \leq j \leq m)$, $p'_j \in Y'_j$, and that for each $j (1 \leq j \leq k)$, $a_j(p'_j) \Rightarrow_{\mathcal{A}'} q_{ij}$.

Consider the derivation of $\mathcal{D}$ that produces $p'$:

$$d(p) = d(\sigma(p_1, \ldots, p_m)) \Rightarrow_{\mathcal{D}} \phi_\sigma(d(p_1), \ldots, d(p_m)) \Rightarrow_{\mathcal{D}} \phi_\sigma(p'_1, \ldots, p'_m) = p'.$$

By the construction of $\mathcal{A}'$, the rule

$$\langle a(\sigma(x_1, \ldots, x_m)) \rightarrow t[a_1(x_{i_1}), \ldots, a_k(x_{i_k})]; Y_1, \ldots, Y_m \rangle$$

must be in $P_{\mathcal{A}}$, and for each $j (1 \leq j \leq m)$, $Y_j = L(H'_j)$ with $H'_j \in W_n$, $Y'_j = L(H''_j)$ with $H''_j \in W'_n$, and tree automata $H'_j$ and $H''_j$ correspond via the bijection $\alpha_n$. By Claim G, $p_j \in Y'_j$ (1 ≤ j ≤ m). By the induction hypothesis, $a_j(p_{i_j}) \Rightarrow_{\mathcal{A}} q_{ij}$ (1 ≤ j ≤ k). Hence,

$$a(p) = a(\sigma(p_1, \ldots, p_m)) \Rightarrow_{\mathcal{A}} t[a_1(p_{i_1}), \ldots, a_k(p_{i_k})] \Rightarrow_{\mathcal{A}} t[q_{i_1}, \ldots, q_{i_k}] = q. \qed$$

This concludes the proof of Theorem 3.4.

4. Consequences and conclusion

In this section we summarize the consequences of Theorems 3.3 and 3.4.

**Consequence 4.1.** For every $n \geq 0$, $DTT^{DR_n} \subseteq (DTT^{DR})^{n+1}$. 
Proof. We proceed by an induction on \( n \). For \( n = 1 \), the inclusion follows from Theorem 3.3. Suppose that the result holds for \( n - 1 \). Then

\[
D^{DR_n} \subseteq D^{DR} \circ D^{DR_{n-1}} \quad \text{by Theorem 3.4}
\]

\[
\subseteq D^{DR} \circ (D^{DR})^n \quad \text{by the induction hypothesis}
\]

\[
= (D^{DR})^{n+1}.
\]

We now recall two results from [17]. The first one states that the composition of \( n + 1 \) \( dtt^{DR} \)'s can be simulated by a single \( dtt^{DR_n} \), and the second one states that iterating the compositions of \( dtt^{DR} \)'s gives rise to a proper hierarchy.

Proposition 4.2. For every \( n \geq 0 \), \( (D^{DR}_n)^{n+1} \subseteq D^{DR_n} \).

Proposition 4.3. For every \( n \geq 0 \), \( (D^{DR}_n)^n \subseteq (D^{DR})^{n+1} \).

Consequence 4.1 and Proposition 4.2 establish the main result of our paper which can be viewed as expressing a trade-off relationship between look-ahead and composition: \( DR_n \) look-ahead is equivalent to \( (n + 1) \)-fold composition.

Theorem 4.4. For every \( n \geq 0 \), \( D^{DR_n} = (D^{DR})^{n+1} \).

Theorem 4.4 and Proposition 4.3 imply the following decomposition and hierarchy results for the classes \( D^{DR_n} \).

Consequence 4.5. For every \( k, n \geq 0 \), \( D^{DR_k} \circ D^{DR_n} = D^{DR_{k+n+1}} \).

Consequence 4.6. For every \( n \geq 0 \), \( D^{DR_n} \subseteq D^{DR_{n+1}} \).

We conclude by asking whether the results proved in this paper for deterministic top-down tree transducers hold for nondeterministic transducers as well. For example, do the classes \( TT_{DR_n} (n \geq 0) \) form a proper hierarchy? Or, how is the class \( TT_{DR_n+1} \) related to \( (TT^{DR})^n \)? We will state these questions formally as a very strong conjecture.

Conjecture 4.7. For every \( n \geq 0 \), \( TT^{DR_n} = (TT^{DR})^{n+1} \).

We also conjecture that all the inclusions in (†), in Section 1, are proper.

References