Available online at www.sciencedirect.com

SCIENCE DIRECT

Theoretical Computer Science 329 (2004) 115–158

Theoretical Computer Science

www.elsevier.com/locate/tcs

Storage-to-tree transducers with look-ahead ☆

Tamás Hornung[a], Sándor Vágvölgyi[b,*]

[a] *Mathematics and Statistics Department, Budapest Business School, Zalaegerszeg, Gasparich utca 18, H-8900 Hungary*

[b] *Department of Foundations of Computer Science, University of Szeged, Szeged, Árpád tér 2, H-6720 Hungary*

Received 2 November 2002; received in revised form 2 July 2004; accepted 19 August 2004

Communicated by G. Rozenberg

Abstract

We generalize Engelfriet's decomposition result stating that the class of transformations induced by top–down tree transducers with regular look-ahead is equal to the composition of the class of top–down tree transformations and the class of linear tree homomorphisms. Replacing the input trees with an arbitrary storage type, the top–down tree transducers are turned into regular storage-to-tree transducers. We show that the class of transformations induced by regular storage-to-tree transducers with positive look-ahead is equal to the composition of the class of transformations induced by regular storage-to-tree transducers with the class of linear tree homomorphisms. We also show that the classes of transformations induced by both *IO* and *OI* context-free storage-to-tree transducers are closed under positive look-ahead, and are closed under composition with linear tree homomorphisms.

Keywords: Tree grammar; Storage type; Transducer; Look-ahead

1. Introduction

The notion of look-ahead is an efficient tool in tree language theory, see [1,4,5,11–13,19,20]. Top–down tree transducers with regular look-ahead are capable of

☆ This research was supported by the Széchenyi István Scholarship of the Hungarian Ministry of Education and the grant OTKA T 030084 of the Research Foundation of Hungary.

* Corresponding author. Tel.: +36-62-546-192; fax: +36-62-546-397.

E-mail addresses: hornung.tamas@pszfz.bgf.hu (T. Hornung), vagvolgy@inf.u-szeged.hu (S. Vágvölgyi).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2004.08.007

inspecting subtrees before deleting or translating them. Engelfriet [4] showed the decomposition result

$$T^R = T \circ LH, \tag{1}$$

where T^R, T, and LH denote the classes of transformations induced by top–down tree transducers with regular look-ahead, top–down tree transducers, and linear tree homomorphisms, respectively (cf. Corollary 2.13 in [4]).

A top–down tree transducer can be considered as a nondeterministic recursive program that acts on trees and generates trees. If the recursive functions in this program are provided with parameters, the macro tree transducer of [7] is obtained. We can describe such a program as a grammar which operates on a storage type. Thus, we obtain the concepts of regular, *IO* context-free and *OI* context-free storage-to-tree transducers which provide us with a general and uniform framework to transducers, see [5,8].

In a regular tree (*RT*) grammar the nonterminals have rank 0, see [14]. Context-free tree (*CFT*) grammars are obtained from *RT* grammars by allowing nonterminals of rank greater than 0. *CFT* grammars can be considered with unrestricted derivation and with two restricted modes of derivation: inside–out (*IO*) and outside–in (*OI*). A *CFT* grammar depending on the restricted mode of derivation is said to be either an inside–out (*IO*) context-free tree grammar or an outside–in (*OI*) context-free tree grammar, see [6].

The concept of a storage type was introduced in [5,8]. Roughly speaking, a storage type S consists of a set of input elements and a set of configurations. The input elements are encoded as configurations. The configurations can be tested by predicates of S and can be transformed by instructions of S. Let MOD be the set $\{RT, IO, OI, CF\}$ of modifiers, where CF denotes the type of context-free (string) grammars. Let modifier K range over MOD. A K S-to-tree transducer, or $K(S)$ transducer for short, is a K grammar of which every rule is provided with a test, and every nonterminal of the right-hand side of the rule has an instruction. Considering a derivation of the $K(S)$ transducer, each occurrence of a nonterminal A is associated with a configuration c, different occurrences may be associated with different configurations. A rule of the $K(S)$ transducer can be applied to the tuple $A(c)$ as a rule of a K grammar can be applied to the nonterminal A, provided that the test specified by the rule holds for c, and the instructions also specified by the rule are defined on c. The new configurations for the nonterminals of the right-hand side of the rule are obtained by transforming c according to the instructions of the respective nonterminals. The initial nonterminal of the grammar is associated with a configuration that is an encoded input element. Thus, the $K(S)$ transducer induces a transformation from the input set to the set of terminal trees or strings. If K or S is not specified, we speak about a storage-to-tree transducer. Note that the $RT(S)$ transducer can be viewed as either the $IO(S)$ or the $OI(S)$ transducer in which only rank 0 is allowed for nonterminals.

We also study deterministic transducers: $DK(S)$ transducers. In the sequel, $DMOD$ stands for the set $\{DRT, DIO, DOI, DCF\}$ of deterministic modifiers. For modifier $K \in MOD \cup DMOD$, the class of transformations induced by $K(S)$ transducers is denoted by $K(S)$. For example, $DIO(S)$ is the class of tree transformations induced by deterministic $IO(S)$ transducers.

For particular storage types S, $CF(S)$ transducers can be associated with indexed grammars, attribute grammars, generalized syntax directed translation schemes, etc., see [5].

Top–down tree transducers are $RT(S)$ transducers for a particular storage type S. The tree storage type, denoted by TR, is a storage type in which the input elements and the configurations are trees, the root of trees can be tested and the trees can be transformed into their immediate subtrees. Top–down tree transformations are the same as $RT(TR)$ transformations, and deterministic top–down tree transformations are the same as $DRT(TR)$ transformations, that is,

$$T = RT(TR) \quad \text{and} \quad DT = DRT(TR), \tag{2}$$

see Corollary 3.20 in [8].

The concept of a storage type S with look-ahead, denoted by S_{CF}, was introduced in [5,8] as a generalization of regular look-ahead. Storage type S_{CF} is obtained from S by adding special tests, so-called look-ahead tests, to the set of predicates of S. These look-ahead tests are written in the form $\langle \mathcal{L} \rangle$, where $\mathcal{L}$ is a $CF(S)$ transducer. Look-ahead test $\langle \mathcal{L} \rangle$ is true on a configuration c if and only if the $CF(S)$ transducer $\mathcal{L}$ can derive a terminal string from $A_{\text{in}}(c)$, where A_{in} is the initial nonterminal of $\mathcal{L}$. For each modifier $M \in MOD \cup DMOD$, we define the storage type S with M look-ahead, denoted by S_M, from S_{CF} by replacing the $CF(S)$ transducers in the look-ahead tests with $M(S)$ transducers. Recall that the class of domains of top–down tree transformations is exactly the class of recognizable tree languages, see [14]. The class of recognizable tree languages is closed under the Boolean operations. Since the test of a rule is a Boolean expression of predicates, one can show that

$$T^R = RT(TR_{RT}), \tag{3}$$

see the first line of p. 335 in [8]. Hence by (2), Engelfriet's decomposition theorem (1) takes the form

$$RT(TR_{RT}) = RT(TR) \circ LH. \tag{4}$$

We show that the RT, IO, OI, and CF look-ahead tests are all equivalent, and that the DRT, DIO, and DCF look-ahead tests are pairwise equivalent as well. On the basis of this result we show that for each storage type S, and for every modifier K in $MOD \cup DMOD$, $K(S_{RT}) = K(S_{IO}) = K(S_{OI}) = K(S_{CF})$ and $K(S_{DRT}) = K(S_{DIO}) = K(S_{DCF}) \subseteq K(S_{DOI})$.

We show that for each modifier $K \in \{RT, IO, OI\}$ and storage type S,

$$K(S_K) \supseteq K(S) \circ LH. \tag{5}$$

This result is a generalization of the inclusion $RT(TR_{RT}) \supseteq RT(TR) \circ LH$, see (4).

The reverse of inclusion (5) does not hold for the notion of look-ahead as defined in the literature. Hence the generalization of Engelfriet's result (4) does not hold. We, therefore, introduce the notion of positive look-ahead. In the literature, the test of a rule can contain the negation of a look-ahead test. This is nice and convenient for the particular storage types considered in, e.g., [5,8], but it is not acceptable for other storage types. In fact, if one considers the tape of a Turing machine as a storage type, then a look-ahead test can test membership in an arbitrary recursively enumerable set. Allowing negation, this means that a regular storage-to-tree transducer with look-ahead on this storage type can induce functions that are not computable (such as the characteristic function of a recursively enumerable set that is not recursive). This is undesirable and also means that the generalization of Engelfriet's result cannot hold for this notion of look-ahead because any composition of a regular storage-to-tree transducer with a linear tree homomorphism clearly induces computable functions only. For this reason, we forbid the negation of look-ahead tests and call it positive look-ahead. Obviously, storage-to-tree transducers with positive look-ahead induce computable functions only, and so positive look-ahead is, in general, less powerful than look-ahead. Positive look-ahead is still a very natural notion of look-ahead. In particular, the two notions of look-ahead are equivalent for top–down tree transducers (because the recognizable tree languages are closed under complement), and so Engelfriet's result can as well be viewed as a result on positive look-ahead.

Let $K \in MOD$ and $M \in MOD$. Let S be a storage type. A K S-to-tree transducer with positive M look-ahead ($K^+(S_M)$ transducer for short), is a $K(S_M)$ transducer, where the test of each rule is of the form

$$b \textbf{ and } \langle \mathcal{L}_1 \rangle \textbf{ and } \cdots \textbf{and } \langle \mathcal{L}_n \rangle.$$

Here b is a Boolean expression over the predicate symbols of S, $n \geqslant 0$, and for each $1 \leqslant i \leqslant n$, $\langle \mathcal{L}_i \rangle$ is a look-ahead test of the storage type S_M. Note that if negation of any M look-ahead test does not occur in any test of a $K(S_M)$ transducer, then that is a $K^+(S_M)$ transducer, because it can easily be shown that one can eliminate **or** from tests of rules. Now, it is immediate from the definition of T^R that $T^R \subseteq RT^+(TR_{RT})$. By (3), $RT(TR_{RT}) = T^R = RT^+(TR_{RT})$. By (4), Engelfriet's decomposition theorem (1) takes the form

$$RT^+(TR_{RT}) = RT(TR) \circ LH. \tag{6}$$

We generalize (6) by showing the following. For every storage type S, $RT^+(S_{RT}) = RT(S) \circ LH$.

We show that for every storage type S, $IO(S)$ is closed under positive look-ahead, is closed under composition with linear tree homomorphisms, and is even closed under composition with tree homomorphisms. That is, for every storage type S, $IO^+(S_{IO}) = IO(S) = IO(S) \circ LH = IO(S) \circ H$. We also show that for every storage type S, $OI(S)$ is closed under positive look-ahead, and is closed under composition with linear tree homomorphisms. That is, for every storage type S, $OI^+(S_{OI}) = OI(S) = OI(S) \circ LH$.

For the tree storage type $S = TR$ (discussed above), the $IO(S)$ and $OI(S)$ transducers are the same as the IO and OI macro tree transducers of [7] (see [8]). In [7] the classes of IO and OI macro tree transformations are denoted by MT_{IO} and MT_{OI}, respectively. Thus

$$IO(TR) = MT_{IO} \text{ and } OI(TR) = MT_{OI}$$

and similarly for the deterministic case. The above results on positive look-ahead generalize the fact that MT_{IO} and MT_{OI} are closed under regular look-ahead [7]. For the trivial storage type $S = S_0$, the $IO(S_0)$ and $OI(S_0)$ transducers are essentially the same as the IO and OI context-free tree grammars (cf. Lemma 3.9 of [8]). The above results on closure under composition with (linear) tree homomorphisms generalize the fact that the IO and OI context-free tree languages are closed under tree homomorphisms [6] and linear tree homomorphisms [17], respectively. They are new for macro tree transducers.

2. Preliminaries

In this section, we present a review of the notions, notations, and results used in the paper.

2.1. General notations

Let $n \geqslant 0$ and for each $1 \leqslant i \leqslant n$, let $b_i \in \{\mathbf{true}, \mathbf{false}\}$. We define the Boolean value $\bigwedge(b_1, \ldots, b_n)$ as follows. For $n \geqslant 2$, let $\bigwedge(b_1, \ldots, b_n) = b_1$ **and** $\cdots$ **and** b_n. For $n = 1$, let $\bigwedge(b_1) = b_1$. Finally, let $\bigwedge(\,) = \mathbf{true}$. Analogous notation will be used for Boolean expressions.

For two sets *A* and *B*, a *binary relation* ρ from *A* into *B* is any subset of $A \times B$. The *domain* $dom(\rho)$ of ρ is the set $\{ u \mid$ there is a $v \in B$ such that $(u, v) \in \rho \}$. For two binary relations ρ_1 and ρ_2, the *composition* $\rho_1 \circ \rho_2$ of ρ_1 and ρ_2 is the set $\{ (u, w) \mid (u, v) \in \rho_1$ and $(v, w) \in \rho_2$ for some $v \}$. Let $A = B$. Then ρ is a binary relation over *A*. For each $k \geqslant 1$, the *k*th power of ρ is denoted by ρ^k. The *reflexive, transitive closure* and the *transitive closure* of ρ are denoted by ρ^* and ρ^+, respectively. The domain $dom(C)$ of a relation class *C*, and the composition $U \circ V$ of relation classes *U* and *V* are defined in the natural way.

A *partial function f* from *A* into *B*, denoted by $f : A \to B$, is a subset of $A \times B$ such that for each $a \in A$ and $b, c \in B$, if $(a, b) \in f$ and $(a, c) \in f$, then $b = c$. For each $a \in A$, we say that partial function *f* is defined on *a* if $a \in dom(f)$. For $(a, b) \in f$, we write $b = f(a)$, as usual. A *function* or a *mapping f* from *A* into *B* is a partial function $f : A \to B$ such that $dom(f) = A$. For a set *A*, let $id_A = \{ (a, a) \mid a \in A \}$ denote the *identity function* on *A*.

2.2. Strings and trees

Let Σ be a set. As usual, Σ^* is the free monoid generated by Σ under the operation of concatenation, with the empty string, λ, as identity. The *length* of a string $w \in \Sigma^*$ is denoted by $|w|$. For the rest of the paper, let Ξ be an infinite set. We consider Ξ as a symbol base. An *alphabet* Σ is a finite subset of Ξ.

A *ranked set* Σ is a (possibly infinite) set in which every symbol has a unique *rank* (*arity*) in the set of nonnegative integers. For any $n \geqslant 0$, we denote by Σ_n the set of symbols in Σ which have rank *n*. The rank of a symbol is sometimes indicated as a superscript, that is, $\sigma^{(2)}$ means that σ is of rank 2. For the rest of the paper, Ω denotes an infinite ranked set such that, for every $n \geqslant 0$, Ω_n is infinite. We consider Ω, too, as a symbol base. A *ranked alphabet* Σ is a finite subset of Ω.

Definition 2.1. Let *Z* and *C* be arbitrary sets. Then $Z(C)$ is the set $\{ z(c) \mid z \in Z$ and $c \in C \}$. Here $z(c)$ is a string of length four over the set containing *Z*, *C*, and the left and right parentheses. We consider an element $z(c)$ of $Z(C)$ as a symbol rather than a string of length four. If *Z* is a ranked set, then the symbol $z(c)$ has the same rank as *z* has.

For a ranked set Σ and a set of variables *Y* with $Y \cap \Sigma = \emptyset$, *the set of trees* (or *terms*) *over* Σ *indexed by* *Y*, denoted by $T_\Sigma(Y)$, is the smallest set *U* satisfying the following two

conditions:
(i) $\Sigma_0 \cup Y \subseteq U$,
(ii) $\sigma(t_1, \dots, t_n) \in U$ whenever $n > 0$, $\sigma \in \Sigma_n$ and $t_1, \dots, t_n \in U$.
The *set of trees over* Σ is $T_\Sigma(\emptyset)$, and we simply write T_Σ for $T_\Sigma(\emptyset)$. A *tree language* is any subset of T_Σ for some ranked alphabet Σ.

For a tree $t \in T_\Sigma(Y)$, *root*(t) and the set $sub(t)$ of *subtrees* of t is defined by tree induction.
(i) If $t \in \Sigma_0 \cup Y$, then *root*$(t) = t$ and $sub(t) = \{t\}$.
(ii) If $t = \sigma(t_1, \dots, t_n)$ with $\sigma \in \Sigma_n$, $n > 0$, then *root*$(t) = \sigma$ and $sub(t) = \{t\} \cup (\bigcup_{i=1}^n sub(t_i))$.

We specify a countable set $X = \{x_1, x_2, \dots\}$ of variables and set $X_n = \{x_1, \dots, x_n\}$ for every $n \geqslant 0$. We distinguish a subset $\bar{T}_\Sigma(X_n)$ of $T_\Sigma(X_n)$ as follows: a tree $t \in T_\Sigma(X_n)$ is in $\bar{T}_\Sigma(X_n)$ if and only if each variable in X_n appears exactly once in t. 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, $\sigma(x_2, \sigma(a, x_1)) \in \bar{T}_\Sigma(X_2)$.

The notion of *tree substitution* is defined as follows. Let $n \geqslant 0$, $t \in T_\Sigma(X_n)$ and $h_1, \dots, h_n \in T_\Sigma(X)$. We denote by $t[h_1, \dots, h_n]$ the tree which is obtained from t by replacing each occurrence of x_i in t by h_i for every $1 \leqslant i \leqslant n$. Furthermore, let $1 \leqslant i \leqslant n$ and $h \in T_\Sigma(X)$ be arbitrary. We denote by $t[x_i \leftarrow h]$ the tree which is obtained from t by replacing each occurrence of x_i in t by h.

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.

2.3. Deterministic bottom–up tree automata

A *deterministic bottom–up tree automaton* is a tuple $\mathcal{A} = (Q, \Sigma, Q_f, \delta)$, where
(i) Q is the state set,
(ii) Σ is a ranked alphabet, $\Sigma \cap Q = \emptyset$,
(iii) Q_f is the set of final states, and
(iv) $\delta = (\delta_\sigma)_{\sigma \in \Sigma}$ is a family of transition functions $\delta_\sigma : Q^m \to Q$, where $\sigma \in \Sigma_m$ and $m \geqslant 0$.

Each tree $t \in T_\Sigma(X_n)$, $n \geqslant 0$, determines a mapping $t^{\mathcal{A}} : Q^n \to Q$ as follows. Let $q_1, \dots, q_n \in Q$ be arbitrary.
- If $t = x_i$, where $1 \leqslant i \leqslant n$, then $t^{\mathcal{A}}(q_1, \dots, q_n) = q_i$.
- If $t = \sigma(t_1, \dots, t_m)$, where $\sigma \in \Sigma_m$, $m \geqslant 0$, $t_1, \dots, t_m \in T_\Sigma(X_n)$, then $t^{\mathcal{A}}(q_1, \dots, q_n) = \delta_\sigma(t_1^{\mathcal{A}}(q_1, \dots, q_n), \dots, t_m^{\mathcal{A}}(q_1, \dots, q_n))$.

We will need the fact that, for $t_0 \in T_\Sigma(X_m)$, $t_1, \dots, t_m \in T_\Sigma$ and $q_1, \dots, q_n \in Q$,

$$(t_0[t_1, \dots, t_m])^{\mathcal{A}}(q_1, \dots, q_n) = t_0^{\mathcal{A}}(t_1^{\mathcal{A}}(q_1, \dots, q_n), \dots, t_m^{\mathcal{A}}(q_1, \dots, q_n)). \quad (7)$$

Obviously, for each tree $t \in T_\Sigma$, $t^{\mathcal{A}}$ is an element of Q. The *tree language recognized by* $\mathcal{A}$ is $L(\mathcal{A}) = \{t \in T_\Sigma \mid t^{\mathcal{A}} \in Q_f\}$. A tree language is *recognizable* if it is recognized by a deterministic bottom–up tree automaton. The class of recognizable tree languages is denoted by *RECOG*.

2.4. Grammars

A *context-free* (CF) *grammar* G is a tuple $(N, \Delta, A_{\text{in}}, R)$, where
(i) N is the nonterminal alphabet,
(ii) Δ is the terminal alphabet, $\Delta \cap N = \emptyset$,

(iii) A_{in} is the initial nonterminal, and
(iv) R is the finite set of rules. Each rule is of the form $A \to \gamma$ with $A \in N$ and $\gamma \in (N \cup \Delta)^*$.

The derivation relation of G, denoted by $\Rightarrow_G$, is a binary relation on $(N \cup \Delta)^*$, defined as follows. For all words $\alpha, \beta \in (N \cup \Delta)^*$, $\alpha \Rightarrow_G \beta$ if and only if
(i) there is a rule $A \to \gamma$ in R, and
(ii) $\alpha = \alpha_1 A \alpha_2$, $\beta = \alpha_1 \gamma \alpha_2$ for some $\alpha_1, \alpha_2 \in (N \cup \Delta)^*$.
The *language generated* by G is $L(G) = \{ w \in \Delta^* \mid A_{in} \Rightarrow_G^* w \}$.

A *context-free tree* (CFT) *grammar* G is a tuple (N, Δ, A_{in}, R), where
(i) N is the nonterminal ranked alphabet,
(ii) Δ is the terminal ranked alphabet, $\Delta \cap N = \emptyset$,
(iii) A_{in} is the initial nonterminal of rank 0, and
(iv) R is the finite set of rules. Each rule is of the form $A(x_1, \ldots, x_n) \to \gamma$ with $n \geqslant 0$, $A \in N_n$, and $\gamma \in T_{N \cup \Delta}(X_n)$.
If $N = N_0$, then G is called a *regular tree* (RT) *grammar*.

We define three binary relations: the *unrestricted*, the *inside–out* and *outside–in derivation* relation of G. The unrestricted derivation relation of G, denoted by $\Rightarrow_{G,U}$, is a binary relation on $T_{N \cup \Delta}(X)$ defined as follows. For all trees $\alpha, \beta \in T_{N \cup \Delta}(X)$, $\alpha \Rightarrow_{G,U} \beta$ if and only if
(i) there is a rule $A(x_1, \ldots, x_n) \to \gamma$ in R, and
(ii) $\alpha = \alpha_1[x_i \leftarrow A(\delta_1, \ldots, \delta_n)]$, $\beta = \alpha_1[x_i \leftarrow \gamma[\delta_1, \ldots, \delta_n]]$, where $\alpha_1 \in T_{N \cup \Delta}(X)$, $i \geqslant 1$, the variable x_i appears exactly once in α_1, and $\delta_1, \ldots, \delta_n \in T_{N \cup \Delta}(X)$.
The definition of the inside–out derivation relation of G, denoted by $\Rightarrow_{G,IO}$, is the same as that of $\Rightarrow_{G,U}$, except that $\delta_1, \ldots, \delta_n$ are required to be terminal trees, that is $\delta_1, \ldots, \delta_n \in T_\Delta$. The definition of outside–in derivation relation of G, denoted by $\Rightarrow_{G,OI}$, is the same as that of $\Rightarrow_{G,U}$, except that variable x_i does not occur in a subtree of α_1 with nonterminal root, i.e., x_i does not occur in a subtree of α_1 of the form $B(\beta_1, \ldots, \beta_k)$, where $k \geqslant 1$, $B \in N_k$, and $\beta_1, \ldots, \beta_k \in T_{N \cup \Delta}(X)$.

Let $M \in \{ U, IO, OI \}$. The *tree language M-generated* by G is

$$L_M(G) = \{ t \in T_\Delta \mid A_{in} \overset{*}{\underset{G,M}{\Rightarrow}} t \}.$$

For an *RT* grammar, the above three derivation relations obviously coincide. The class of tree languages generated by *RT* grammars is equal to *RECOG*, see [14]. Fischer [10], Engelfriet and Schmidt [6] showed that for any *CFT* grammar G, $L_{OI}(G) = L_U(G)$ and $L_{IO}(G) \subseteq L_{OI}(G)$. Whenever we want to consider a *CFT* grammar G with *OI*-derivation, we say that G is an *OI* (*context-free tree*) *grammar* and we denote $\Rightarrow_{G,OI}$ by $\Rightarrow_G$. Similarly, when we want to consider a *CFT* grammar G with *IO*-derivation, we say that G is an *IO* (*context-free tree*) *grammar* and we denote $\Rightarrow_{G,IO}$ by $\Rightarrow_G$.

In the sequel, we shall also consider, for $K \in \{RT, IO, OI\}$, a *generalized K grammar G* in which both the set of nonterminals and the set of rules can be finite or infinite. We extend the definitions of the relations $\Rightarrow_{G,U}$, $\Rightarrow_{G,IO}$, and $\Rightarrow_{G,OI}$ and of the definitions of the tree languages $L_U(G)$, $L_{IO}(G)$, and $L_{OI}(G)$ for this case in the natural way.

Let *MOD* be the set $\{ RT, IO, OI, CF \}$ of modifiers.

2.5. Storage types

We adopt the concept of a storage type from [5,8]. A *storage type S* is a tuple (C, P, F, m, I, E) where C is the nonempty set of configurations, P is the set of predicate symbols, F is the set of instruction symbols, $P \cap F = \emptyset$, I is a set called the input set of S, and E is a set of partial functions $e : I \to C$, every $e \in E$ is called an input encoding of S. Furthermore, m is the meaning function that associates with every $p \in P$ a mapping $m(p) : C \to \{\mathbf{true}, \mathbf{false}\}$, and with every $f \in F$ a partial function $m(f) : C \to C$. $BE(P)$ denotes the set of all Boolean expressions over P, with the usual Boolean operators **and**, **or**, **not**, **true**, and **false**. For $b \in BE(P)$, $m(b) : C \to \{\mathbf{true}, \mathbf{false}\}$ is defined in the obvious way. The elements of $BE(P)$ are also called *tests*. We say that $b_1, b_2 \in BE(P)$ are equivalent if $m(b_1) = m(b_2)$.

We also assume that C, P, F, I are subsets of Ξ introduced as a base set in the preliminaries.

The *trivial storage type* $S_0 = (C, P, F, m, I, E)$ is defined by $C = \{c\}$, where c is an arbitrary object, $P = \emptyset$, $F = \{id\}$, and $m(id) = id_C$, $I = \{c\}$, and $E = \{id_C\}$.

2.6. Transducers

Let $S = (C, P, F, m, I, E)$ be a storage type. A *context-free S transducer*, or $CF(S)$ transducer, is a tuple $\mathcal{A} = (N, e, \Delta, A_{\text{in}}, R)$, where

(i) N is the nonterminal alphabet,
(ii) $e \in E$ is the encoding,
(iii) Δ is the terminal alphabet (disjoint with N),
(iv) $A_{\text{in}} \in N$ is the initial nonterminal, and
(v) R is the finite set of rules. Every rule is of the form

$$A \to \ \mathbf{if}\ b\ \mathbf{then}\ \gamma$$

with $A \in N$, $b \in BE(P)$, and $\gamma \in (N(F) \cup \Delta)^*$. Here we call A the left-hand side of the rule and b the test of the rule.

Recall that, according to Definition 2.1,

$$N(F) = \{A(f) \mid A \in N \text{ and } f \in F\}$$

and

$$N(C) = \{A(c) \mid A \in N \text{ and } c \in C\}.$$

In what follows, we shall consider an element of $N(F) \cup N(C)$ as a symbol rather than a string of length four.

The derivation relation of $\mathcal{A}$, denoted by $\Rightarrow_{\mathcal{A}}$, is a binary relation on the set $(N(C) \cup \Delta)^*$, defined as follows. For all $A \in N$, $c \in C$, and $\xi_1, \xi_2 \in (N(C) \cup \Delta)^*$, if $A \to \ \mathbf{if}\ b\ \mathbf{then}\ \gamma$ is in R, $m(b)(c) = \mathbf{true}$, and $m(f)(c)$ is defined for all $f \in F$ occurring in γ, then $\xi_1 A(c) \xi_2 \Rightarrow_{\mathcal{A}} \xi_1 \gamma' \xi_2$ where γ' is obtained from γ by substituting $B(m(f)(c)) \in N(C)$ for every occurrence of $B(f) \in N(F)$ for all $B \in N$ and $f \in F$.

The *transformation induced* by $\mathcal{A}$ is $\tau(\mathcal{A}) = \{ (u, v) \in I \times \Delta^* \mid A_{\rm in}(e(u)) \Rightarrow^*_{\mathcal{A}} v \}$. Note that $\tau(\mathcal{A}) \subseteq dom(e) \times \Delta^*$. As usual, two $CF(S)$ transducers $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* if $\tau(\mathcal{A}) = \tau(\mathcal{B})$.

Let K range over the set of modifiers $\{IO, OI, RT\}$. A $K(S)$ *transducer* $\mathcal{A}$ is a tuple $(N, e, \Delta, A_{\rm in}, R)$, where

(i) N is the nonterminal ranked alphabet; if $K = RT$, then each element of N is of rank 0,
(ii) $e \in E$ is the encoding,
(iii) Δ is the terminal ranked alphabet (disjoint with N),
(iv) $A_{\rm in} \in N$ is the initial nonterminal of rank 0, and
(v) R is the finite set of rules. Every rule is of the form

$$A(x_1, \ldots, x_n) \to \textbf{ if } b \textbf{ then } \gamma$$

with $n \geqslant 0$, $A \in N_n$, $b \in BE(P)$, and $\gamma \in T_{N(F)\cup\Delta}(X_n)$. Again, we call A the left-hand side of the rule and b the test of the rule.

Recall that in Definition 2.1 we introduced the sets $N(F)$ and $N(C)$. For any $A \in N$ and $f \in F$ and $c \in C$, the symbols $A(f)$ and $A(c)$ have the same rank as A has.

To define the derivation relation of $\mathcal{A}$, we introduce the following notation.

Definition 2.2. Let $\gamma \in T_{N(F)\cup\Delta}(X)$, and $c \in C$ be arbitrary. If for every instruction $f \in F$ occurring in γ, $c \in dom(m(f))$, then we define γ_c from γ by substituting $B(m(f)(c)) \in N(C)$ for every occurrence of $B(f) \in N(F)$ for all $B \in N$ and $f \in F$. Otherwise, γ_c is undefined.

The derivation relation of $\mathcal{A}$, denoted by $\Rightarrow_{\mathcal{A}}$, is a binary relation on $T_{N(C)\cup\Delta}(X)$, defined as follows. We associate with $\mathcal{A}$ a generalized K grammar $G(\mathcal{A}) = (N(C), \Delta, A_{\rm in}(c_0), R_C)$, where c_0 is any element of C, i.e., c_0 is irrelevant. We define the set R_C as follows. For each $c \in C$, we put the rule

$$A(c)(x_1, \ldots, x_n) \to \gamma_c$$

in R_C if there is a rule

$$A(x_1, \ldots, x_n) \to \textbf{ if } b \textbf{ then } \gamma$$

in R such that

(i) $m(b)(c) = \textbf{true}$, and
(ii) for every instruction $f \in F$ occurring in γ, $c \in dom(m(f))$.

The derivation relation of $\mathcal{A}$, denoted by $\Rightarrow_{\mathcal{A}}$ is defined to be $\Rightarrow_{G(\mathcal{A})}$. The *transformation induced* by $\mathcal{A}$ is $\tau(\mathcal{A}) = \{ (u, v) \in I \times T_\Delta \mid A_{\rm in}(e(u)) \Rightarrow^*_{\mathcal{A}} v \}$. Note that $\tau(\mathcal{A}) \subseteq dom(e) \times T_\Delta$.

We call an $IO(S)$ transducer an *IO* context-free *S*-to-tree transducer. We call an $OI(S)$ transducer an *OI* context-free *S*-to-tree transducer. We call an $RT(S)$ transducer a regular *S*-to-tree transducer.

As usual, two $K(S)$ transducers $\mathcal{A}$ and $\mathcal{B}$ are *equivalent* if $\tau(\mathcal{A}) = \tau(\mathcal{B})$.

Definition 2.3. Let $K \in MOD$, and let S be an arbitrary storage type. A $K(S)$ transducer $\mathcal{A}$ is *deterministic* if, for every configuration $c \in C$ of S and any two different rules

$A(x_1, \ldots, x_n) \to$ **if** b_1 **then** γ and $A(x_1, \ldots, x_n) \to$ **if** b_2 **then** δ of $\mathcal{A}$, $m(b_1)(c) =$ **false** or $m(b_2)(c) =$ **false**. A deterministic $K(S)$ transducer is referred to as a $DK(S)$ transducer.

Let *DMOD* be the set $\{ DRT, DIO, DOI, DCF \}$ of modifiers. Let K range over the set of modifiers $MOD \cup DMOD$, and let S be an arbitrary storage type. The class of transformations induced by $K(S)$ transducers is denoted by $K(S)$. We note that Engelfriet and Vogler [8] denoted the class $OI(S)$ by $CFT_1(S)$.

Note that for every storage type S, $RT(S) \subseteq IO(S)$ and $RT(S) \subseteq OI(S)$. In fact, each $RT(S)$ transducer is also an $IO(S)$ transducer and an $OI(S)$ transducer, inducing the same transformation.

The *tree storage type* (*TR* for short) is (C, P, F, m, I, E), where

- $C = T_\Omega$,
- $P = \{ root = \sigma \mid \sigma \in \Omega \}$,
- $F = \{ sel_i \mid i \geqslant 1 \}$,
- for every $c = \omega(t_1, \ldots, t_k) \in T_\Omega$, with $\omega \in \Omega_k$, $k \geqslant 0$, and $t_1, \ldots, t_k \in T_\Omega$, $m(root = \sigma)(c) =$ **true** if and only if $\sigma = \omega$, and $m(sel_i)(c) = t_i$ if $1 \leqslant i \leqslant k$, otherwise $m(sel_i)(c)$ is undefined,
- $I = T_\Omega$, and
- $E = \{ id_{T_\Sigma} \mid \Sigma \text{ is a ranked alphabet} \}$.

Note that, for a ranked alphabet Σ, id_{T_Σ} is a partial function $I \to C$, because $\Sigma \subseteq \Omega$. The purpose of the encoding id_{T_Σ} of a $K(TR)$ transducer $\mathcal{A}$ ($K \in MOD \cup DMOD$) is to specify the ranked input alphabet Σ of $\mathcal{A}$.

Let $K \in MOD \cup DMOD$ and let $\mathcal{A} = (N, id_{T_\Sigma}, \Delta, A_{\text{in}}, R)$ be a $K(TR)$ transducer. We say that $\mathcal{A}$ is in *normal form* if each rule of A has the form

$$A(x_1, \ldots, x_n) \;\to\; \textbf{if } (root = \sigma) \textbf{ then } \gamma,$$

where $\sigma \in \Sigma_k$, $k \geqslant 0$, and for any instruction sel_i occurring in γ, $i \leqslant k$ holds. Engelfriet and Vogler showed the following. For each $K(TR)$ transducer $\mathcal{A}$, there is an equivalent $K(TR)$ transducer $\mathcal{B}$ in normal form, see Lemma 3.18 of [8]. Engelfriet and Vogler also showed that top–down tree transformations are the same as $RT(TR)$ transformations and that deterministic top–down tree transformations are the same as $DRT(TR)$ transformations, see Corollary 3.20 in [8]. It is easy to see that the $OI(TR)$ and $IO(TR)$ transducers are the *OI* and *IO* macro tree transducers of [7], respectively, see also Theorem 3.19 of [8]; this also holds for the deterministic transducers.

We now recall the notion of tree homomorphism from [14], applying the terminology of $RT(TR)$ transducers. An $RT(TR)$ transducer $\mathcal{H} = (N, id_{T_\Sigma}, \Delta, H_{\text{in}}, R)$ is a *tree homomorphism* if

- $\mathcal{H}$ is in normal form,
- $N = N_0 = \{ H_{\text{in}} \}$,
- every $\sigma \in \Sigma$ appears in exactly one rule, and the test of this rule is $root = \sigma$.

For every tree $\alpha \in T_\Sigma(X_n)$, $n \geqslant 0$, $\mathcal{H}(\alpha)$ denotes the unique tree $\beta \in T_\Delta(X_n)$ for which there is a tree $\gamma \in \bar{T}_\Delta(X_k)$, $k \geqslant 0$, such that

- $H_{\text{in}}(\alpha) \Rightarrow^*_{\mathcal{H}} \gamma[H_{\text{in}}(x_{i_1}), \ldots, H_{\text{in}}(x_{i_k})]$, $1 \leqslant i_1, \ldots, i_k \leqslant n$, and
- $\beta = \gamma[x_{i_1}, \ldots, x_{i_k}]$.

Clearly,

$$\tau(\mathcal{H}) = \{ (\alpha, \mathcal{H}(\alpha)) \mid \alpha \in T_\Sigma \}.$$

We observe that the mapping $\mathcal{H} : T_\Sigma(X) \to T_\Delta(X)$ is completely determined by the trees $\mathcal{H}(\sigma(x_1, \ldots, x_n))$, for every $\sigma \in \Sigma_n$, $n \geqslant 0$. In fact,

$$\mathcal{H}(\sigma(\alpha_1, \ldots, \alpha_n)) = \mathcal{H}(\sigma(x_1, \ldots, x_n))[\mathcal{H}(\alpha_1), \ldots, \mathcal{H}(\alpha_n)].$$

More generally, $\mathcal{H}$ distributes over substitution, i.e., for $t \in T_\Sigma(X_n)$,

$$\mathcal{H}(t[\alpha_1, \ldots, \alpha_n]) = \mathcal{H}(t)[\mathcal{H}(\alpha_1), \ldots, \mathcal{H}(\alpha_n)].$$

These facts will often be used in proofs.

We say that $\mathcal{H}$ is *linear* if for any integer $i \geqslant 1$, sel_i occurs at most once in any rule. The class of tree transformations induced by tree homomorphisms is denoted by *H*. Furthermore, the class of tree transformations induced by linear tree homomorphisms is denoted by *LH*.

Let $\mathcal{H} = (\{ H_{\text{in}} \}, id_{T_\Sigma}, \Delta, H_{\text{in}}, R)$ be a tree homomorphism and Θ be a ranked alphabet which is disjoint from $\Sigma \cup \Delta$. The *extension* of $\mathcal{H}$ for Θ is the $RT(TR)$ transducer $\mathcal{H}_\Theta = (\{ H_{\text{in}} \}, id_{T_{\Sigma\cup\Theta}}, \Delta \cup \Theta, H_{\text{in}}, R_\Theta)$ where

$$R_\Theta = R \cup \{ H_{\text{in}} \to \textbf{if } root = \theta \textbf{ then } \theta(H_{\text{in}}(sel_1), \ldots, H_{\text{in}}(sel_n)) \mid n \geqslant 0, \theta \in \Theta_n \}.$$

Obviously $\mathcal{H}_\Theta$ is a tree homomorphism. Note that $\mathcal{H}_\Theta(\theta(x_1, \ldots, x_n)) = \theta(x_1, \ldots, x_n)$ for all $\theta \in \Theta_n$.

Definition 2.4. Let $K \in MOD \cup DMOD$. Let $S = (C, P, F, m, I, E)$ be a storage type. The storage type *S with K look-ahead*, denoted by S_K, is the tuple (C, P_K, F, m_K, I, E), where

- $P_K = P \cup \{ \langle \mathcal{L} \rangle \mid \mathcal{L}$ is an $K(S)$ transducer $\}$,
- m_K restricted to $P \cup F$ is equal to m, and
- for every configuration $c \in C$, $m_K(\langle \mathcal{L} \rangle)(c) =$ **true** if and only if Condition (i) or (ii) holds:
  - (i) $K \in \{ CF, DCF \}$ and there is a string $w \in \Delta^*$ such that $A_{\text{in}}(c) \Rightarrow^*_{\mathcal{L}} w$, where A_{in} is the initial nonterminal of $\mathcal{L}$, and Δ is the terminal alphabet of $\mathcal{L}$,
  - (ii) $K \in \{ RT, IO, OI, DRT, DIO, DOI \}$ and there is a tree $t \in T_\Delta$ such that $A_{\text{in}}(c) \Rightarrow^*_{\mathcal{L}} t$, where A_{in} is the initial nonterminal of $\mathcal{L}$, and Δ is the terminal ranked alphabet of $\mathcal{L}$.

Predicate symbol $\langle \mathcal{L} \rangle$ is called a *K look-ahead test on S*. Let $K_1, K_2 \in MOD \cup DMOD$ and let $\mathcal{L}_i$ be a $K_i(S)$ transducer for $1 \leqslant i \leqslant 2$. We say that $\langle \mathcal{L}_1 \rangle$ and $\langle \mathcal{L}_2 \rangle$ are equivalent if $m_{K_1}(\langle \mathcal{L}_1 \rangle) = m_{K_2}(\langle \mathcal{L}_2 \rangle)$.

3. Results on look-ahead tests

Consider an arbitrary storage type *S*. We now show that the *RT*, *IO*, *OI*, and *CF* look-ahead tests are all equivalent. Furthermore, we also show that the *DRT*, *DIO*, and *DCF* look-ahead tests are pairwise equivalent as well. Observe that *DOI* is missing from the above list of deterministic look-ahead tests.

Let Σ be a ranked alphabet, Y be a set of variables, and $\Delta \subseteq \Sigma \cup Y$. The *$\Delta$-projection* $pr_\Delta(t) \in \Delta^*$ of a tree $t \in T_\Sigma(Y)$ is defined as follows.

(i) Let $t \in \Sigma_0 \cup Y$. If $t \in \Delta$, then $pr_\Delta(t) = t$ else $pr_\Delta(t) = \lambda$.

(ii) Let $t = \sigma(t_1, \ldots, t_n)$, where $\sigma \in \Sigma_n$, $n > 0$, and $t_1, \ldots, t_n \in T_\Sigma(Y)$. If $\sigma \in \Delta$, then $pr_\Delta(t) = \sigma pr_\Delta(t_1) \cdots pr_\Delta(t_n)$, otherwise $pr_\Delta(t) = pr_\Delta(t_1) \cdots pr_\Delta(t_n)$.

Lemma 3.1. *Let S be an arbitrary storage type. For any CF look-ahead test on S, there is an equivalent RT look-ahead test on S. Determinism is preserved, that is, for any DCF look-ahead test on S, there is an equivalent DRT look-ahead test on S.*

Proof. It is well known that the context-free languages are equal to the yield languages of recognizable tree languages. In the light of this result the lemma is quite obvious. Let $\mathcal{A} = (N, e, \Delta, A_{\text{in}}, R)$ be a $CF(S)$ transducer. We define the ranked alphabet Γ as follows. For each $n \geqslant 1$, let

$$\Gamma_n = \{ \langle A \to \textbf{if } b \textbf{ then } \gamma \rangle \mid A \to \textbf{if } b \textbf{ then } \gamma \text{ is a rule in } R \text{ and } |\gamma| = n \}.$$

Moreover, let

$$\Gamma_0 = \Delta \cup \{ \langle A \to \textbf{if } b \textbf{ then } \lambda \rangle \mid A \to \textbf{if } b \textbf{ then } \lambda \text{ is a rule in } R \}.$$

We construct the $RT(S)$ transducer $\mathcal{B} = (N, e, \Gamma, A_{\text{in}}, R')$, where for any $a_1, \ldots, a_n \in N(F) \cup \Delta$, $n \geqslant 0$, the rule

$$A \to \textbf{if } b \textbf{ then } \langle A \to \textbf{if } b \textbf{ then } a_1 \cdots a_n \rangle (a_1, \ldots, a_n) \text{ is in } R'$$

if and only if

$$A \to \textbf{if } b \textbf{ then } a_1 \cdots a_n \text{ is in } R.$$

Obviously, determinism is preserved.

It is straightforward to show, by induction on the length of the derivations, that for every $A(c) \in N(C)$ and every string $w \in \Delta^*$, $A(c) \Rightarrow^*_{\mathcal{A}} w$ if and only if there is a tree $t \in T_\Gamma$ such that $A(c) \Rightarrow^*_{\mathcal{B}} t$ and $pr_\Delta(t) = w$. □

Lemma 3.2. *Let S be an arbitrary storage type. For every IO look-ahead test on S there is an equivalent CF look-ahead test on S. Determinism is preserved.*

Proof. Let $\mathcal{A} = (N, e, \Delta, A_{\text{in}}, R)$ be an $IO(S)$ transducer. We define the $CF(S)$ transducer $\mathcal{B} = (N, e, \emptyset, A_{\text{in}}, R')$ as follows. For each rule

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ then } \gamma$$

in R, we put the rule

$$A \to \textbf{if } b \textbf{ then } pr_{N(F)}(\gamma)$$

in R'.

Obviously, determinism is preserved.

We show that for any tree $p \in T_{N(C)\cup\Delta}$ and integer $l \geqslant 0$ the following two statements are equivalent:

(i) $p \Rightarrow^{l}_{\mathcal{A}} t$ for some $t \in T_{\Delta}$,

(ii) $pr_{N(C)}(p) \Rightarrow^{l}_{\mathcal{B}} \lambda$.

((i)$\Rightarrow$ (ii)) Let us assume that Condition (i) holds. We show (ii) by induction on l.

Base case: $l = 0$. Then $p = t \in T_{\Delta}$. Hence $pr_{N(C)}(p) = \lambda$.

Induction step: $l > 0$. Then $p = p_0[A(c)(t_1, \ldots, t_n)] \Rightarrow_{\mathcal{A}} p_0[\gamma_c[t_1, \ldots, t_n]] \Rightarrow^{l-1}_{\mathcal{A}} t$, where $n \geqslant 0$, $A(c) \in N_n(C)$, $t_1, \ldots, t_n \in T_{\Delta}$, $p_0 \in \bar{T}_{N(C)\cup\Delta}(X_1)$,

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ then } \gamma \in R,$$

$m(b)(c) = \textbf{true}$ and for every instruction $f \in F$ occurring in γ, $c \in dom(m(f))$. Hence

$$A \to \textbf{if } b \textbf{ then } pr_{N(F)}(\gamma) \in R'. \tag{8}$$

By the induction hypothesis,

$$pr_{N(C)}(p_0[\gamma_c[t_1, \ldots, t_n]]) \Rightarrow^{l-1}_{\mathcal{B}} \lambda. \tag{9}$$

Let $pr_{N(C)\cup X_1}(p_0) = ux_1v$ for some strings $u, v \in N(C)^*$. Then

$$pr_{N(C)}(p_0[\gamma_c[t_1, \ldots, t_n]]) = u(pr_{N(C)}(\gamma_c))v. \tag{10}$$

Hence $pr_{N(C)}(p) = uA(c)v \Rightarrow_{\mathcal{B}} u(pr_{N(C)}(\gamma_c))v$ by (8)
$= pr_{N(C)}(p_0[\gamma_c[t_1, \ldots, t_n]])$ by (10)
$\Rightarrow^{l-1}_{\mathcal{B}} \lambda$ by (9).

((ii)$\Rightarrow$ (i)) Let us assume that Condition (ii) holds. We now show (i) by induction on l.

Base case: $l = 0$. Then $pr_{N(C)}(p) = \lambda$. Hence $p \in T_{\Delta}$. Let $t = p$. Then (i) holds true.

Induction step: $l > 0$. Condition (ii) implies that

$$pr_{N(C)}(p) = uA(c) \underset{\mathcal{B}}{\Rightarrow} u(pr_{N(C)}(\gamma_c)) \Rightarrow^{l-1}_{\mathcal{B}} \lambda, \tag{11}$$

where $u \in N(C)^*$, $A(c) \in N_n(C)$ for some $n \geqslant 0$,

$$A \to \textbf{if } b \textbf{ then } pr_{N(F)}(\gamma) \in R',$$

$m(b)(c) = \textbf{true}$ and for every instruction $f \in F$ occurring in γ, $c \in dom(m(f))$. Hence

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ then } \gamma \in R. \tag{12}$$

Then it follows from $pr_{N(C)}(p) = uA(c)$ that

$$p = p_0[A(c)(t_1, \ldots, t_n)] \tag{13}$$

for some $p_0 \in \bar{T}_{N(C)\cup\Delta}(X_1)$, $t_1, \ldots, t_n \in T_{\Delta}$. Furthermore $pr_{N(C)\cup X_1}(p_0) = ux_1$.

We observe that $pr_{N(C)}(p_0[\gamma_c[t_1, \ldots, t_n]]) = u(pr_{N(C)}(\gamma_c))$. Hence by (11),

$pr_{N(C)}(p_0[\gamma_c[t_1, \ldots, t_n]]) \Rightarrow_{\mathcal{B}}^{l-1} \lambda$. By the induction hypothesis, there is a tree $t \in T_\Delta$ such that

$$p_0[\gamma_c[t_1, \ldots, t_n]] \Rightarrow_{\mathcal{A}}^{l-1} t. \tag{14}$$

Thus $p = p_0[A(c)(t_1, \ldots, t_n)]$ by (13)
$\Rightarrow_{\mathcal{A}} p_0[\gamma_c[t_1, \ldots, t_n]]$ by (12)
$\Rightarrow_{\mathcal{A}}^{l-1} t$ by (14).

Now that we have shown that Conditions (i) and (ii) are equivalent, we finish the proof of the lemma by taking $p = A_{\mathrm{in}}(c)$. Observe that $pr_{N(C)}(A_{\mathrm{in}}(c)) = A_{\mathrm{in}}(c)$, and that the rules of R' do not include any terminal symbol. Hence for each $c \in C$, $m_{IO}(\langle\mathcal{A}\rangle)(c) = m_{CF}(\langle\mathcal{B}\rangle)(c)$. □

The following result is essentially shown in the proof of Lemma 8.8 of [8], but we repeat the proof for completeness sake.

Lemma 3.3. *Let S be an arbitrary storage type. For every OI look-ahead test on S there is an equivalent CF look-ahead test on S.*

Proof. Let $S = (C, P, F, m, I, E)$ be an arbitrary storage type. As in the proof of Lemma 8.8 of [8], we define the storage type $S' = (C, P, F, m, C, \{id_C\})$.

To every $OI(S)$ transducer $\mathcal{A} = (N, e, \Delta, A_{\mathrm{in}}, R)$ we assign the $OI(S')$ transducer $\mathcal{A}' = (N, id_C, \Delta, A_{\mathrm{in}}, R)$. We obtain by direct inspection that for each configuration $c \in C$, $m_{OI}(\langle\mathcal{A}\rangle)(c) = \mathbf{true}$ if and only if $c \in dom(\tau(\mathcal{A}'))$.

Now let $\mathcal{A} = (N, e, \Delta, A_{\mathrm{in}}, R)$ be an arbitrary $OI(S)$ transducer. By Definition 5.22 of [8] and Lemma 6.11 of [8], $dom(OI(S')) \subseteq dom(CF(S'))$. Hence there is a $CF(S')$ transducer $\mathcal{C} = (N_1, id_C, \Delta, C_{\mathrm{in}}, R_1)$ such that $dom(\tau(\mathcal{C})) = dom(\tau(\mathcal{A}'))$. Consider the $CF(S)$ transducer $\mathcal{B} = (N_1, e, \Delta, C_{\mathrm{in}}, R_1)$. Then $\mathcal{B}' = \mathcal{C}$. Hence for each configuration $c \in C$, $m_{OI}(\langle\mathcal{B}\rangle)(c) = \mathbf{true}$ if and only if $c \in dom(\tau(\mathcal{C}))$. Thus for every $c \in C$, $m_{OI}(\langle\mathcal{A}\rangle)(c) = m_{CF}(\langle\mathcal{B}\rangle)(c)$. □

We now show that the deterministic version of Lemma 3.3 does not hold. We adopt the notion of a deterministic *OI* macro tree transducer from [7]. We denote the class of all tree transformations induced by deterministic *OI* macro tree transducers by DMT_{OI}. It is well known that $dom(DMT_{OI}) = RECOG$, see Theorem 6.18 of [7]. By Theorem 3.19 in [8], $DOI(TR) = DMT_{OI}$. Hence

$$dom(DOI(TR)) = RECOG. \tag{15}$$

By Lemma 3.1, the domain of any $DCF(TR)$ transformation is the domain of a $DRT(TR)$ transformation. Theorem 3.1 of [4] states that the domains of all $DRT(TR)$ transformations are the tree languages recognized by deterministic top–down tree automata. It is well known that there exist recognizable tree languages that cannot be recognized by a deterministic top–down tree automaton. Hence there is a $DOI(TR)$ transducer $\mathcal{A}$ such that look-ahead test $\langle\mathcal{A}\rangle$ on *S* is not equivalent with any *DCF* look-ahead test.

Theorem 3.4. *Let S be an arbitrary storage type.*
(i) *The RT, IO, OI, and CF look-ahead tests on S are pairwise equivalent.*
(ii) *The DRT, DIO, and DCF look-ahead tests on S are pairwise equivalent.*

Proof. Observe that every *RT* look-ahead test on *S* is also an *IO* and an *OI* look-ahead test on *S*. Similarly, every *DRT* look-ahead test on *S* is also a *DIO* look-ahead test on *S*. Hence our result follows from Lemmas 3.1–3.3. □

We now give a storage type *S* such that there is no *DOI* look-ahead test on *S* equivalent to some *OI* look-ahead tests on *S*. Let $S = (C, \emptyset, \{ f_1, f_2 \}, m, C, \{ id_C \})$, where $C = \{ c_1, c_2, c_3 \}$ and $m(f_i) = id_{\{ c_i \}}$ for $i = 1, 2$. It is not hard to see that $\{ c_1, c_2 \} \in dom(OI(S))$ and $\{ c_1, c_2 \} \notin dom(DOI(S))$.

From Theorem 3.4 and the fact that every *DRT* look-ahead test on *S* is also a *DOI* look-ahead test on *S*, we obtain the next corollary.

Corollary 3.5. *Let S be an arbitrary storage type, and let* $K \in MOD \cup DMOD$. *Then* $K(S_{DRT}) = K(S_{DIO}) = K(S_{DCF}) \subseteq K(S_{DOI}) \subseteq K(S_{RT}) = K(S_{IO}) = K(S_{OI}) = K(S_{CF})$.

Definition 3.6. Let $K, M \in MOD \cup DMOD$. Let $S = (C, P, F, m, I, E)$ be a storage type. Let $\mathcal{A}$ be a $K(S_M)$ transducer.

(i) $\mathcal{A}$ has *positive look-ahead on S* (is a $K^+(S_M)$ transducer for short) if the test of each rule of $\mathcal{A}$ is of the form

$$b \textbf{ and } \langle \mathcal{L}_1 \rangle \textbf{ and} \cdots \textbf{and } \langle \mathcal{L}_n \rangle, \tag{16}$$

where $b \in BE(P)$, $n \geqslant 0$, and for each $1 \leqslant i \leqslant n$, $\langle \mathcal{L}_i \rangle$ is an *M* look-ahead test on *S*.

(ii) $\mathcal{A}$ has *1-positive look-ahead on S* (is a $K^{1+}(S_M)$ transducer for short) if the test of each rule of $\mathcal{A}$ is of the form (16) with $n = 1$.

The class of transformations induced by $K^+(S_M)$ transducers is denoted by $K^+(S_M)$. The class of transformations induced by $K^{1+}(S_M)$ transducers is denoted by $K^{1+}(S_M)$.

We now show that these two classes are the same, because look-ahead tests on *S* are closed under **and**.

Theorem 3.7. *Let S be an arbitrary storage type. Let* $K, M \in MOD \cup DMOD$ *be arbitrary. Then* $K^+(S_M) = K^{1+}(S_M)$.

Proof. By Definition 3.6, $K^+(S_M) \supseteq K^{1+}(S_M)$.

We now show that $K^+(S_M) \subseteq K^{1+}(S_M)$. First we consider the case that $M \notin \{CF, DCF\}$.

Let $\mathcal{A} = (N, e, \Delta, A_{\text{in}}, R_1)$ be a $K^+(S_M)$ transducer. We construct a $K^{1+}(S_M)$ transducer $\mathcal{B} = (N, e, \Delta, A_{\text{in}}, R_2)$ equivalent to $\mathcal{A}$. We define the rule set R_2 of $\mathcal{B}$ as follows. By

Definition 3.6, each test appearing in R_1 is of the form (16). For each rule

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ and } \langle \mathcal{L}_1 \rangle \textbf{ and } \cdots \textbf{ and } \langle \mathcal{L}_n \rangle \textbf{ then } \gamma$$

in R_1, we put the rule

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ and } \langle \mathcal{L} \rangle \textbf{ then } \gamma$$

in R_2. Here the $M(S)$ transducer $\mathcal{L} = (N^0, e, \Delta^0, A_{\text{in}}, R)$ is defined as follows:

If $n = 0$, then let $\mathcal{L} = (\{ A_{\text{in}} \}, e, \{ \omega^{(0)} \}, A_{\text{in}}, \{ A_{\text{in}} \to \textbf{ if true then } \omega \})$.

If $n = 1$, then let $\mathcal{L} = \mathcal{L}_1$.

If $n \geqslant 2$, then assume that $\mathcal{L}_i = (N^i, e^i, \Delta^i, A^i_{\text{in}}, R^i)$ for $1 \leqslant i \leqslant n$, and that the sets N^i are pairwise disjoint. Then let

(i) $N^0 = \bigcup_{i=1}^n N^i \cup \{ A_{\text{in}} \}$, where $A_{\text{in}} \notin \bigcup_{i=1}^n N^i$.

(ii) $\Delta^0 = \bigcup_{i=1}^n \Delta^i \cup \{ \omega^{(n)} \}$, where ω is a new terminal symbol.

We put the rule

$$A_{\text{in}} \to \textbf{ if } b_1 \textbf{ and} \cdots \textbf{and } b_n \textbf{ then } \omega(\gamma_1, \ldots, \gamma_n)$$

in R, where for each $1 \leqslant i \leqslant n$, $A^i_{\text{in}} \to \textbf{ if } b_i \textbf{ then } \gamma_i \in R^i$. Moreover, we put all elements of the set $\bigcup_{i=1}^n R^i$ in R. It should be clear that transducer $\mathcal{B}$ is equivalent to $\mathcal{A}$.

The construction for $M \in \{ CF, DCF \}$ is exactly the same, except that we take $\gamma_1 \cdots \gamma_n$ instead of $\omega(\gamma_1, \ldots, \gamma_n)$. □

The next corollary is obtained in the same way as Corollary 3.5.

Corollary 3.8. *Let S be an arbitrary storage type, and let* $K \in MOD \cup DMOD$. *Then* $K^+(S_{DRT}) = K^+(S_{DIO}) = K^+(S_{DCF}) \subseteq K^+(S_{DOI}) \subseteq K^+(S_{RT}) = K^+(S_{IO}) = K^+(S_{OI}) = K^+(S_{CF})$.

Theorem 3.9. $DRT^+(TR_{DRT}) \subset DRT^+(TR_{DOI})$.

Proof. By Corollary 3.8, $DRT^+(TR_{DRT}) \subseteq DRT^+(TR_{DOI})$. We are going to show that $DRT^+(TR_{DRT}) \neq DRT^+(TR_{DOI})$. Fülöp and Vágvölgyi [11] introduced the deterministic top–down tree transducer with deterministic top–down check denoted by DT^{DTRC}. We obtain by direct inspection that the deterministic top–down tree transducer with deterministic top–down check is the same as the $DRT^+(TR_{DRT})$ transducer. Hence

$$DT^{DTRC} = DRT^+(TR_{DRT}). \tag{17}$$

Fülöp and Vágvölgyi [12] have shown that there is a recognizable tree language K_0 not in $dom(DRT^+(TR_{DRT}))$. By (15), $K_0 \in dom(DOI(TR))$. Hence $K_0 \in dom(DRT^+(TR_{DOI}))$. □

We now give an additional normal form for transducers with positive look-ahead.

Lemma 3.10. *Let S be an arbitrary storage type. Let* $K \in MOD$ *and* $M \in MOD \cup DMOD$ *be arbitrary. Let* $\mathcal{A} = (N, e, \Delta, A_{\text{in}}, R_{\mathcal{A}})$ *be a* $K^{1+}(S_M)$ *transducer. Then there is an equivalent* $K^{1+}(S_M)$ *transducer* $\mathcal{B} = (N, e, \Delta, A_{\text{in}}, R_{\mathcal{B}})$ *such that the following condition holds. Let* $A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ and } \langle \mathcal{C} \rangle \textbf{ then } \gamma$ *be an arbitrary rule of* $R_{\mathcal{B}}$. *Then the M(S) transducer* $\mathcal{C} = (N_{\mathcal{C}}, e_{\mathcal{C}}, \Delta_{\mathcal{C}}, C_{\text{in}}, R_{\mathcal{C}})$ *has exactly one rule with* C_{in} *appearing in the left-hand side, and that rule has the form* $C_{\text{in}} \to \textbf{if true then } \delta$.

Proof. Let $\mathcal{A} = (N, e, \Delta, A_{\text{in}}, R_{\mathcal{A}})$ be a $K^{1+}(S_M)$ transducer. We define $K^{1+}(S_M)$ transducer $\mathcal{B} = (N, e, \Delta, A_{\text{in}}, R_{\mathcal{B}})$ as follows. Let $A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ and } \langle \mathcal{C} \rangle \textbf{ then } \gamma$ be an arbitrary rule of $R_{\mathcal{A}}$. Let $\mathcal{C} = (N_{\mathcal{C}}, e_{\mathcal{C}}, \Delta_{\mathcal{C}}, C_{\text{in}}, R_{\mathcal{C}})$. Let $k \geqslant 0$ and let $C_{\text{in}} \to \textbf{if } b_i \textbf{ then } \delta_i$, $1 \leqslant i \leqslant k$, be all rules of $\mathcal{C}$ with left-hand side C_{in}. For each $1 \leqslant i \leqslant k$, we define $\mathcal{C}_i = (N_{\mathcal{C}} \cup \{ C^i_{\text{in}} \}, e_{\mathcal{C}}, \Delta_{\mathcal{C}}, C^i_{\text{in}}, R_i)$ as follows. C^i_{in} is a new nonterminal with rank 0. We put the rule $C^i_{\text{in}} \to \textbf{if true then } \delta_i$ in R_i. We put each rule of $\mathcal{C}$ in R_i. For each $1 \leqslant i \leqslant k$, we put the rule $A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ and } b_i \textbf{ and } \langle \mathcal{C}_i \rangle \textbf{ then } \gamma$ in $R_{\mathcal{B}}$.

It is left to the reader to show that $\tau(\mathcal{A}) = \tau(\mathcal{B})$. □

4. *RT* transducers

By the decomposition theorem (6) of Engelfriet, $RT^{+}(TR_{RT}) = RT(TR) \circ LH$. We now generalize this composition result for an arbitrary storage type S.

Lemma 4.1. *For every storage type S,* $RT^{+}(S_{RT}) \supseteq RT(S) \circ LH$ *and* $DRT^{+}(S_{DRT}) \supseteq DRT(S) \circ LH$.

Proof. Let $S = (C, P, F, m, I, E)$. Let $\mathcal{A} = (N, e, \Sigma, A_{\text{in}}, R_1)$ be an $RT(S)$ transducer and let $\mathcal{H} = (\{ H_{\text{in}} \}, id_{T_\Gamma}, \Delta, H_{\text{in}}, R_2)$ be a linear tree homomorphism. Without loss of generality we may assume that $\Sigma = \Gamma$.

We define the $RT^{+}(S_{RT})$ transducer $\mathcal{A} \circ \mathcal{H} = (N, e, \Delta, A_{\text{in}}, R_3)$ as follows. We put the rule

$$A \to \textbf{if } b \textbf{ and } \langle \mathcal{L} \rangle \textbf{ then } \mathcal{H}_{N(F)}(\gamma) \tag{18}$$

in R_3, where Conditions (A) and (B) hold.

(A) The rule

$$A \to \textbf{if } b \textbf{ then } \gamma \tag{19}$$

is in R_1.

(B) $\mathcal{L} = (N \cup \{ B_{\text{in}} \}, e, \Sigma, B_{\text{in}}, R'_1)$ is an $RT(S)$ transducer, where B_{in} is a new nonterminal with rank 0 and $R'_1 = R_1 \cup \{ B_{\text{in}} \to \textbf{if true then } \gamma \}$.

We say that rule (18) is the *image* of rule (19).

If $\mathcal{A}$ is deterministic, then $\mathcal{A} \circ \mathcal{H}$ is an $DRT^{+}(S_{DRT})$ transducer.

Claim 4.2. *Let $\mathcal{L}$ be as in Condition* (B). *For each configuration $c \in C$, $m_{RT}(\langle \mathcal{L} \rangle)(c) =$* **true** *if and only if for every instruction $f \in F$ occurring in γ, $c \in dom(m(f))$, and there is a tree $w \in T_\Sigma$ such that $\gamma_c \Rightarrow^*_{\mathcal{A}} w$.* (*For the definition of γ_c, see Definition* 2.2.)

Proof. By the construction of the transducer $\mathcal{L}$. □

Intuitively, Claim 4.2 states that the look-ahead test $\mathcal{L}$ is true on an arbitrary configuration c if and only if γ_c is defined and the transducer $\mathcal{A}$ can derive a terminal tree from $\gamma_c \in T_{N(C)\cup\Delta}$.

It is sufficient to show that

$$\tau(\mathcal{A} \circ \mathcal{H}) = \tau(\mathcal{A}) \circ \tau(\mathcal{H}). \tag{20}$$

To this end we show the following result.

Claim 4.3. *For each $\alpha \in T_{N(C)\cup\Sigma}$ and $t \in T_\Delta$, Conditions* (I) *and* (II) *are equivalent.*

(I) *$\alpha \Rightarrow^*_{\mathcal{A}} p$ for some $p \in T_\Sigma$ and $\mathcal{H}_{N(C)}(\alpha) \Rightarrow^*_{\mathcal{A}\circ\mathcal{H}} t$,*

(II) *$\alpha \Rightarrow^*_{\mathcal{A}} s$ and $\mathcal{H}(s) = t$ for some $s \in T_\Sigma$.*

Proof. First we show that Condition (I) implies Condition (II). Let $\alpha \in T_{N(C)\cup\Sigma}$ and $t \in T_\Delta$ be arbitrary. Let $\alpha \Rightarrow^*_{\mathcal{A}} p$ for some $p \in T_\Sigma$ and

$$\mathcal{H}_{N(C)}(\alpha) \Rightarrow^l_{\mathcal{A}\circ\mathcal{H}} t \tag{21}$$

for some $l \geqslant 0$. We show by induction on l that (II) holds.

Base case of the proof of (II): If $l = 0$ then

$$\mathcal{H}_{N(C)}(\alpha) = t. \tag{22}$$

We show by tree induction on α that

$$\mathcal{H}(p) = t. \tag{23}$$

Base case of the proof of (23): $\alpha \in N(C) \cup \Sigma_0$. In this case by (22) and $t \in T_\Delta$, $\alpha \in \Sigma_0$. By (I) $p = \alpha$, hence by (22) $\mathcal{H}(p) = \mathcal{H}(\alpha) = t$.

Induction step of the proof of (23): As $t \in T_\Delta$, by (22) $\mathcal{H}_{N(C)}(\alpha) \in T_\Delta$. Thus $root(\alpha) \notin N(C)$. Hence $\alpha = \sigma(\alpha_1, \ldots, \alpha_n)$ for some $n \geqslant 1$, $\sigma \in \Sigma_n$, and $\alpha_1, \ldots, \alpha_n \in T_{N(C)\cup\Sigma}$. By (22),

$$t = \bar{t}[t_1, \ldots, t_n], \tag{24}$$

where $\bar{t} = \mathcal{H}(\sigma(x_1, \ldots, x_n)) \in T_\Delta(X_n)$ and

$$t_i = \mathcal{H}_{N(C)}(\alpha_i) \in T_{N(C)\cup\Delta} \quad \text{for } 1 \leqslant i \leqslant n. \tag{25}$$

Let

$$APP = \{ i \mid 1 \leqslant i \leqslant n \text{ and } x_i \text{ appears in } \bar{t} \}.$$

By (I), for each $i = 1, \ldots, n$, $\alpha_i \Rightarrow^*_{\mathcal{A}} p_i$ for some $p_i \in T_\Sigma$, and $p = \sigma(p_1, \ldots, p_n)$. Since $t \in T_\Delta$, for each $i \in APP$, $t_i \in T_\Delta$. By the induction hypothesis, for each $i \in APP$, $\mathcal{H}(p_i) = t_i$. Then $\mathcal{H}(p) = \bar{t}[\mathcal{H}(p_1), \ldots, \mathcal{H}(p_n)] = \bar{t}[t_1, \ldots, t_n] = t$, by (24). Hence the proof of (23) is complete.

Let $s = p$. Then by (23), Condition (II) trivially holds. The base of the proof of (II) is complete.

Induction step of the proof of (II): Let $l > 0$. The first step of derivation (21) is the result of applying rule (18). From that it follows that Conditions (A) and (B) hold. Reordering some of its steps, we can rewrite derivation (21) in the following way:

(a) $\mathcal{H}_{N(C)}(\alpha) = \bar{\beta}[A(c)] \Rightarrow_{\mathcal{A}\circ\mathcal{H}} \bar{\beta}[\delta] \Rightarrow^j_{\mathcal{A}\circ\mathcal{H}} \bar{\beta}[q] \Rightarrow^k_{\mathcal{A}\circ\mathcal{H}} t$ for some $\bar{\beta} \in \bar{T}_{N(C)\cup\Delta}(X_1)$, $A(c) \in N(C)$, and $\delta \in T_{N(C)\cup\Delta}$, $q \in T_\Delta$, $j, k \geqslant 0$ with $j + k = l - 1$.

Furthermore, Conditions (b)–(d) hold:

(b) $m_{RT}(b$ **and** $\langle\mathcal{L}\rangle)(c) =$ **true**.

(c) $\mathcal{H}_{N(F)}(\gamma)_c = \delta$.

(d) $\delta \Rightarrow^j_{\mathcal{A}\circ\mathcal{H}} q$.

Moreover, as H is linear, the following two conditions hold:

(e) $\alpha = \bar{\alpha}[A(c)]$, for some $\bar{\alpha} \in \bar{T}_{N(C)\cup\Sigma}(X_1)$.

(f) $\mathcal{H}_{N(C)}(\bar{\alpha}) = \bar{\beta}$.

Now, the derivation $\alpha \Rightarrow^*_{\mathcal{A}} p$ can be written as follows:

(g) $\alpha = \bar{\alpha}[A(c)] \Rightarrow^*_{\mathcal{A}} \bar{\alpha}[p'] \Rightarrow^*_{\mathcal{A}} \bar{p}[p'] = p$ for some $p' \in T_\Sigma$ and $\bar{p} \in T_\Sigma(X_1)$.

(h) $\bar{\alpha} \Rightarrow^*_{\mathcal{A}} \bar{p}$.

By (b) and Claim 4.2, there is a tree $w \in T_\Sigma$ such that $\gamma_c \Rightarrow^*_{\mathcal{A}} w$. Observe that $\mathcal{H}_{N(C)}(\gamma_c) = \mathcal{H}_{N(F)}(\gamma)_c$. By (c) and (d), we have $\mathcal{H}_{N(C)}(\gamma_c) = \delta \Rightarrow^j_{\mathcal{A}\circ\mathcal{H}} q$. By the induction hypothesis, there is a tree $s' \in T_\Sigma$ such that

$$\gamma_c \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} s' \tag{26}$$

and

$$\mathcal{H}(s') = q. \tag{27}$$

Then by (f), (27), (a), and the fact that the tree homomorphism $\mathcal{H}_{N(C)}$ distributes over substitution, $\mathcal{H}_{N(C)}(\bar{\alpha}[s']) = \mathcal{H}_{N(C)}(\bar{\alpha})[\mathcal{H}(s')] = \bar{\beta}[q] \Rightarrow^k_{\mathcal{A}\circ\mathcal{H}} t$. By (h), $\bar{\alpha}[s'] \Rightarrow^*_{\mathcal{A}} \bar{p}[s'] \in T_\Sigma$. By the induction hypothesis, there is a tree $s \in T_\Sigma$ such that $\bar{\alpha}[s'] \Rightarrow^*_{\mathcal{A}} s$ and $\mathcal{H}(s) = t$. Hence by (e), (A), (b), Claim 4.2, and (26), $\alpha = \bar{\alpha}[A(c)] \Rightarrow_{\mathcal{A}} \bar{\alpha}[\gamma_c] \Rightarrow^*_{\mathcal{A}} \bar{\alpha}[s'] \Rightarrow^*_{\mathcal{A}} s$ and $\mathcal{H}(s) = t$.

Second we show that Condition (II) implies Condition (I). Let $\alpha \in T_{N(C)\cup\Sigma}$ and $t \in T_\Delta$. Assume that

$$\alpha \Rightarrow^l_{\mathcal{A}} s \tag{28}$$

and $\mathcal{H}(s) = t$ for some $l \geqslant 0$ and $s \in T_\Sigma$. We show by induction on l that $\mathcal{H}_{N(C)}(\alpha) \Rightarrow^*_{\mathcal{A}\circ\mathcal{H}} t$.

Base case: Let $l = 0$. Then $\alpha = s$ and $\mathcal{H}(\alpha) = t$. Then $\mathcal{H}_{N(C)}(\alpha) = t$, hence $\mathcal{H}_{N(C)}(\alpha) \Rightarrow^*_{\mathcal{A}\circ\mathcal{H}} t$.

Induction step: Let $l > 0$. Then the first step of derivation (28) is the result of applying rule

$$A \to \textbf{if } b \textbf{ then } \gamma \tag{29}$$

in R_1. Let

$$\alpha = \bar{\alpha}[A(c)], \tag{30}$$

where $\bar{\alpha} \in \bar{T}_{N(C)\cup\Sigma}(X_1)$ and $c \in C$. Furthermore, there is a tree $s' \in T_\Sigma$ such that

$$\alpha = \bar{\alpha}[A(c)] \underset{\mathcal{A}}{\Rightarrow} \bar{\alpha}[\gamma_c] \Rightarrow^j_{\mathcal{A}} \bar{\alpha}[s'] \Rightarrow^k_{\mathcal{A}} s, \tag{31}$$

with $j + k = l - 1$ and

(i) $\gamma_c \Rightarrow^j_{\mathcal{A}} s'$.

Let

(ii) $t' = \mathcal{H}(s') \in T_\Delta$,

(iii) $\bar{\beta} = \mathcal{H}_{N(C)}(\bar{\alpha}) \in T_{N(C)\cup\Delta}(X_1)$, and

(iv) $\delta = \mathcal{H}_{N(F)}(\gamma)$.

Note that since tree homomorphism $\mathcal{H}_{N(C)}$ is linear, $\bar{\beta}$ contains in fact at most one occurrence of the variable x_1 but we will not make use of this fact.

By (30), (iii) and the distribution of $\mathcal{H}_{N(C)}$ over substitution, $\mathcal{H}_{N(C)}(\alpha) = \bar{\beta}[A(c)]$. Recall that rule (29) is in R_1. By the definition of $\mathcal{A} \circ \mathcal{H}$, and (iv), the rule

$$A \to \textbf{if } b \textbf{ and } \langle\mathcal{L}\rangle \textbf{ then } \delta \tag{32}$$

is in R_3, where $\mathcal{L}$ is as in Condition (B). Recall that $s' \in T_\Sigma$. Since γ_c is defined, by (i) and Claim 4.2, $m_{RT}(\langle\mathcal{L}\rangle)(c) = \textbf{true}$. Hence we can apply rule (32) in the following derivation as many times as x_1 occurs in $\bar{\beta}$:

$$\mathcal{H}_{N(C)}(\alpha) = \bar{\beta}[A(c)] \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \bar{\beta}[\delta_c]. \tag{33}$$

By (iv), $\mathcal{H}_{N(C)}(\gamma_c) = \delta_c$. By (i), (ii), and the induction hypothesis, $\mathcal{H}_{N(C)}(\gamma_c) = \delta_c \Rightarrow^*_{\mathcal{A}\circ\mathcal{H}} t'$. Hence

$$\bar{\beta}[\delta_c] \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \bar{\beta}[t']. \tag{34}$$

By the last part of (31), the fact that $\mathcal{H}(s) = t$, and the induction hypothesis,

$$\mathcal{H}_{N(C)}(\bar{\alpha}[s']) \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} t. \tag{35}$$

By (iii) and (ii),

$$\mathcal{H}_{N(C)}(\bar{\alpha}[s']) = \bar{\beta}[t'].$$

Hence by (35),

$$\bar{\beta}[t'] \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} t. \tag{36}$$

By (33), (34), and (36), we obtain that

$$\mathcal{H}_{N(C)}(\alpha) \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \bar{\beta}[\delta_c] \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \bar{\beta}[t'] \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} t.$$

Hence Condition (I) holds in this case, too. □

We now continue with the proof of Eq. (20). Let $u \in I$ be arbitrary. We now distinguish two cases.

Case 1: $u \in dom(\tau(\mathcal{A}))$. In this case by Claim 4.3 for every $t \in T_\Delta$,

$$A_{\text{in}}(e(u)) = \mathcal{H}_{N(C)}(A_{\text{in}}(e(u))) \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} t$$

if and only if

$$A_{\text{in}}(e(u)) \underset{\mathcal{A}}{\overset{*}{\Rightarrow}} s \text{ and } \mathcal{H}(s) = t \quad \text{for some } s \in T_\Sigma.$$

Case 2: $u \notin dom(\tau(\mathcal{A}))$. In this case by Claim 4.2 for every rule

$$A_{\text{in}} \to \textbf{if } b \textbf{ and } \langle\mathcal{L}\rangle \textbf{ then } \mathcal{H}_{N(C)}(\gamma)$$

in R_3, $m_{RT}(b \textbf{ and } \langle\mathcal{L}\rangle)(e(u)) = \textbf{false}$. Thus $u \notin dom(\tau(\mathcal{A} \circ \mathcal{H}))$.

These two cases prove Eq. (20). □

To generalize Engelfriet's decomposition result (6) for an arbitrary storage type *S*, we have to show the following.

Lemma 4.4. *For every storage type* S, $RT^+(S_{RT}) \subseteq RT(S) \circ LH$.

Proof. Let $S = (C, P, F, m, I, E)$ be an arbitrary storage type. Let $\mathcal{B} = (N^0, e, \Delta^0, A_{\text{in}}, R^0)$ be an $RT^+(S_{RT})$ transducer. Without loss of generality, we may assume that $\mathcal{B}$ is an $RT^{1+}(S_{RT})$ transducer, see Theorem 3.7. We construct an $RT(S)$ transducer $\mathcal{A}$ and a linear tree homomorphism $\mathcal{H}$. Then for the $RT(S)$ transducer $\mathcal{A}$ and the linear tree homomorphism $\mathcal{H}$, we construct the $RT^+(S_{RT})$ transducer $\mathcal{A} \circ \mathcal{H}$ as in the proof of Lemma 4.1. Then we show that $\tau(\mathcal{B}) = \tau(\mathcal{A} \circ \mathcal{H})$. By the proof of Lemma 4.1, $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \tau(\mathcal{A} \circ \mathcal{H})$. Hence $\tau(\mathcal{B}) = \tau(\mathcal{A}) \circ \tau(\mathcal{H})$.

We construct the $RT(S)$ transducer $\mathcal{A} = (N, e, \Sigma, A_{\text{in}}, R_1)$ in the following way. Let us number the rules of R^0 by the numbers $1, \ldots, r$, for some $r \geqslant 0$. Let us assume that the ith rule is of the form

$$A_i \to \textbf{if } b_i \textbf{ and } \langle\mathcal{L}^i\rangle \textbf{ then } \gamma_i \tag{37}$$

where $1 \leqslant i \leqslant r$, $A_i \in N^0$, $b_i \in BE(P)$, $\mathcal{L}^i = (N^i, e, \Delta^i, A^i_{\text{in}}, R^i)$ is an $RT(S)$ transducer, $\gamma_i \in T_{N^0(F)\cup\Delta^0}$. By Lemma 3.10, we may assume that the $RT(S)$ transducer $\mathcal{L}^i$ has only one rule with A^i_{in} appearing in the left-hand side. That rule is of the form

$$A^i_{\text{in}} \to \textbf{ if true then } \gamma^i. \tag{38}$$

We may assume that the sets $N^0, N^1, \ldots, N^r$ and $\Delta^0, \Delta^1, \ldots, \Delta^r$ are pairwise disjoint. Let ω be a new terminal symbol of arity 2, and let

- $N = \bigcup_{i=0}^r N^i$ and
- $\Sigma = \bigcup_{i=0}^r \Delta^i \cup \{\omega^{(2)}\}$.
- For each $1 \leqslant i \leqslant r$, we put the rule

$$A_i \to \textbf{if } b_i \textbf{ then } \omega(\gamma_i, \gamma^i), \tag{39}$$

in R_1, where rule (37) is the ith rule of R^0, and the rule (38) is in R^i.

We put each rule of the set $\bigcup_{i=1}^r R^i$ in R_1.

Claim 4.5. *For any $A(c) \in N^0(C)$ and $w \in T_{\Delta^0}$, if $A(c) \Rightarrow^*_{\mathcal{B}} w$, then $A(c) \Rightarrow^*_{\mathcal{A}} s$ for some $s \in T_\Sigma$.*

Proof. Let $A(c) \Rightarrow^l_{\mathcal{B}} w$ for some $l \geqslant 1$. We can proceed by induction on l. □

Let $\mathcal{H} = (\{H_{\text{in}}\}, id_{T_\Sigma}, \Delta^0, H_{\text{in}}, R_2)$ and the rule set R_2 consists of the following rules:

- $H_{\text{in}} \to$ **if** $root = \omega$ **then** $H_{\text{in}}(sel_1)$
- $H_{\text{in}} \to$ **if** $root = \sigma$ **then** $\sigma(H_{\text{in}}(sel_1), \ldots, H_{\text{in}}(sel_n))$, where $n \geqslant 0$, $\sigma \in \Sigma_n - \{\omega\}$.

For the $RT(S)$ transducer $\mathcal{A}$ and the linear tree homomorphism $\mathcal{H}$, we construct the $RT(S_{RT})$ transducer $\mathcal{A} \circ \mathcal{H}$ as in the proof of Lemma 4.1. Observe that, broadly speaking, the nonterminals in the set $\bigcup_{i=1}^r R^i$ are not reachable from the initial nonterminal of $\mathcal{A} \circ \mathcal{H}$. Let us define the transducer $\mathcal{D}$ from $\mathcal{A} \circ \mathcal{H}$ by dropping the images of the rules in $\bigcup_{i=1}^r R^i$. By our observation

$$\tau(\mathcal{D}) = \tau(\mathcal{A} \circ \mathcal{H}). \tag{40}$$

For each $1 \leqslant i \leqslant r$, the image of rule (39) of R_1 is a rule of $\mathcal{A} \circ \mathcal{H}$, and is of the form

$$A_i \to \textbf{if } b_i \textbf{ and } \langle \mathcal{C}_i \rangle \textbf{ then } \gamma_i, \tag{41}$$

where

$$\mathcal{C}_i = (N \cup \{C_{\text{in}}\}, e, \Sigma, C_{\text{in}}, R_1 \cup \{C_{\text{in}} \to \textbf{ if true then } \omega(\gamma_i, \gamma^i)\})$$

and C_{in} is a new nonterminal symbol. For $1 \leqslant i \leqslant r$, rules (41) are the rules of transducer $\mathcal{D}$. For each $1 \leqslant i \leqslant r$, condition b_i **and** $\langle \mathcal{C}_i \rangle$ in rule (41) implies the condition b_i **and** $\langle \mathcal{L}^i \rangle$ in rule (37). Hence $\tau(\mathcal{D}) \subseteq \tau(\mathcal{B})$. On the other hand, let $c \in C$ be arbitrary. Assume that there is a derivation $A_i(c) \Rightarrow_{\mathcal{B}} (\gamma_i)_c \Rightarrow^*_{\mathcal{B}} w$ for some $w \in T_{\Delta^0}$, where we apply rule (37) in the first step. Then by Claim 4.5, the condition b_i **and** $\langle \mathcal{C}_i \rangle$ in rule (41) is true for c. Hence we can apply rule (41) for $A_i(c)$. Hence $\tau(\mathcal{B}) \subseteq \tau(\mathcal{D})$. By (40), $\tau(\mathcal{B}) = \tau(\mathcal{A} \circ \mathcal{H})$. By the proof of Lemma 4.1, $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \tau(\mathcal{A} \circ \mathcal{H})$. Hence $\tau(\mathcal{B}) = \tau(\mathcal{A}) \circ \tau(\mathcal{H})$. □

Theorem 4.6. *For every storage type S*, $RT^+(S_{RT}) = RT(S) \circ LH$.

Proof. By Lemmas 4.1 and 4.4 we are done. □

By Lemma 4.1,

$$DRT^+(TR_{DRT}) \supseteq DRT(TR) \circ LH. \tag{42}$$

Inclusion diagram of Fig. 1 in [11], and Theorem 5 in [13] imply the proper inclusion

$$DT^{DTRC} \supset DT \circ DT \supseteq DT \circ LH, \tag{43}$$

where DT^{DTRC} has been introduced in the proof of Theorem 3.9, and *DT* denotes the class of tree transformations induced by all deterministic top–down tree transducers. Hence by (2) and (17),

$$DRT^+(TR_{DRT}) \supset DRT(TR) \circ LH. \tag{44}$$

This shows that the deterministic version of Theorem 4.6 does not hold.

In Theorem 4.6 we have generalized Engelfriet's decomposition result for $S = TR$. We wish to observe here that Theorem 4.6 also generalizes the well-known fact that the class *RECOG* of recognizable tree languages is closed under linear tree homomorphisms (see, e.g., Theorem II.4.16 of [14]). In fact, this is Theorem 4.6 for $S = S_0$, the trivial storage type. To see this, note that it is easy to show that $RT(S_0)$ is closed under look-ahead, i.e., that, for $S = S_0$, $RT^+(S_{RT}) = RT(S)$ (see, e.g., Lemma 2.6 of [9]). And it is easy to see (cf. Lemma 3.9 of [8]) that $RT(S_0)$ is essentially the class of tree languages generated by regular tree grammars, i.e., *RECOG*.

5. *IO* transducers

We show that for every storage type *S*, $IO(S)$ is closed under positive look-ahead and is closed under composition with tree homomorphisms. That is, for every storage type *S*, $IO^+(S_{IO}) = IO(S) = IO(S) \circ LH = IO(S) \circ H$.

In order to prove that $IO(S)$ is closed under composition with tree homomorphisms, we need the special case that the tree homomorphism is the identity on T_Δ where Δ is a subalphabet of the terminal alphabet of the $IO(S)$ transducer. The proof of this result is nontrivial and of the same complexity as that of the next, more general, fact: $IO(S)$ is closed under composition with the identity on a recognizable tree language. The proof is standard. It generalizes, for $S = S_0$, the fact that the *IO* context-free tree languages are closed under tree homomorphisms (see Corollary 6.2 of [6]). Let ID_{RECOG} denote the class of all mappings id_L with $L \in RECOG$.

Lemma 5.1. *For every storage type S*, $IO(S) \circ ID_{RECOG} \subseteq IO(S)$.

Proof. Let $\mathcal{A} = (N, e_1, \Sigma, A_{\text{in}}, R_1)$ be an $IO(S)$ transducer. Let $\mathcal{B} = (Q, \Sigma, Q_f, \delta)$ be a deterministic bottom–up tree automaton with $L(\mathcal{B}) = L$. We construct an $IO(S)$ transducer $\mathcal{D}$ such that $\tau(\mathcal{D}) = \tau(\mathcal{A}) \circ id_L$.

To this end we generalize the notion of an $IO(S)$ transducer. We now construct an $IO(S)$ transducer $\mathcal{C}$ with finitely many initial states. Let $\mathcal{C} = (N_{\mathcal{C}}, e_1, \Sigma, \{ A_{\text{in}}^{\phi} \mid \phi \in Q_f \}, R_2)$, where $N_{\mathcal{C}} = \{ A^{\phi} \mid A \in N_n, n \geqslant 0, \text{ and } \phi : Q^n \to Q \}$ and every $A^{\phi} \in N_{\mathcal{C}}$ has the same rank as A.

In order to define R_2, first, we extend the deterministic bottom–up tree automaton $\mathcal{B}$ for the set of symbols $\Sigma' = \Sigma \cup N_{\mathcal{C}}(C \cup F)$. Let $\mathcal{B}' = (Q, \Sigma', Q_f, \delta')$ be the deterministic bottom–up tree automaton where $\delta'_{\sigma} = \delta_{\sigma}$ for $\sigma \in \Sigma$, $\delta'_{A^{\phi}(c)} = \phi$ for $A^{\phi}(c) \in N_{\mathcal{C}}(C)$ and $\delta'_{A^{\phi}(f)} = \phi$ for $A^{\phi}(f) \in N_{\mathcal{C}}(F)$. Note that, for each $t \in T_{\Sigma}$, $t^{\mathcal{B}'} = t^{\mathcal{B}}$. Furthermore, for each $\alpha \in T_{\Sigma'}(X_n)$, $n \geqslant 0$, we define $\bar{\alpha} \in T_{\Sigma \cup N(C \cup F)}(X_n)$ from α by replacing every $A^{\phi} \in N_{\mathcal{C}}$ by $A \in N$. Note that, for $t \in T_{\Sigma}$, $\bar{t} = t$.

Now, R_2 consists of all the rules

$$A^{\phi}(x_1, \dots, x_n) \to \textbf{if } b \textbf{ then } \gamma, \tag{45}$$

where $A(x_1, \dots, x_n) \to \textbf{if } b \textbf{ then } \bar{\gamma}$ is in R_1 and $\phi = \gamma^{\mathcal{B}'}$.

We define $\Rightarrow_{\mathcal{C}}$ in the same way as for an $IO(S)$ transducer. The *transformation induced* by $\mathcal{C}$ is

$$\tau(\mathcal{C}) = \{ (u, v) \in I \times T_{\Sigma} \mid A_{\text{in}}^{\phi}(e(u)) \overset{*}{\underset{\mathcal{C}}{\Rightarrow}} v \text{ for some } \phi \in Q_f \}. \tag{46}$$

The following statement holds:

(a) For any $\alpha, \beta \in T_{\Sigma \cup N_{\mathcal{C}}(C)}$, if $\alpha \Rightarrow_{\mathcal{C}} \beta$ then $\bar{\alpha} \Rightarrow_{\mathcal{A}} \bar{\beta}$ and $\alpha^{\mathcal{B}'} = \beta^{\mathcal{B}'}$.

Indeed, if $\alpha = \alpha_0[A^{\phi}(c)(\alpha_1, \dots, \alpha_n)] \Rightarrow_{\mathcal{C}} \alpha_0[\gamma_c[\alpha_1, \dots, \alpha_n]] = \beta$ for rule (45) in R_2, with $\alpha_0 \in T_{\Sigma \cup N_{\mathcal{C}}(C)}(X_1)$ and $\alpha_1, \dots, \alpha_n \in T_{\Sigma \cup N_{\mathcal{C}}(C)}$, then, by the definition of R_2,

$$\bar{\alpha} = \bar{\alpha}_0[A(c)(\bar{\alpha}_1, \dots, \bar{\alpha}_n)] \underset{\mathcal{A}}{\Rightarrow} \bar{\alpha}_0[\bar{\gamma}_c[\bar{\alpha}_1, \dots, \bar{\alpha}_n]] = \bar{\beta}$$

and, by (7) and $\phi = \gamma^{\mathcal{B}'} = \gamma_c^{\mathcal{B}'}$,

$$\alpha^{\mathcal{B}'} = \alpha_0^{\mathcal{B}'}(\phi(\alpha_1^{\mathcal{B}'}, \dots, \alpha_n^{\mathcal{B}'})) = \alpha_0^{\mathcal{B}'}[\gamma_c^{\mathcal{B}'}[\alpha_1^{\mathcal{B}'}, \dots, \alpha_n^{\mathcal{B}'}]] = \beta^{\mathcal{B}'}.$$

By (a) we have

(b) for any $\alpha, \beta \in T_{\Sigma \cup N_{\mathcal{C}}(C)}$, if $\alpha \Rightarrow_{\mathcal{C}}^* \beta$ then $\bar{\alpha} \Rightarrow_{\mathcal{A}}^* \bar{\beta}$ and $\alpha^{\mathcal{B}'} = \beta^{\mathcal{B}'}$.

Thus

(c) for any $\phi \in Q$, $c \in C$ and $t \in T_{\Sigma}$, if $A_{\text{in}}^{\phi}(c) \Rightarrow_{\mathcal{C}}^* t$ then $A_{\text{in}}(c) \Rightarrow_{\mathcal{A}}^* t$ and $\phi = t^{\mathcal{B}}$.

Conversely, we show the following statement.

(d) For any $\xi, \zeta \in T_{\Sigma \cup N(C)}$ and $\beta \in T_{\Sigma \cup N_{\mathcal{C}}(C)}$, if $\xi \Rightarrow_{\mathcal{A}} \zeta$ and $\bar{\beta} = \zeta$, then there is a tree $\alpha \in T_{\Sigma \cup N_{\mathcal{C}}(C)}$ such that $\bar{\alpha} = \xi$, and $\alpha \Rightarrow_{\mathcal{C}} \beta$.

To prove (d), assume that $\xi = \xi_0[A(c)(\xi_1, \dots \xi_n)] \Rightarrow_{\mathcal{A}} \xi_0[\eta_c(\xi_1, \dots \xi_n)] = \zeta$ for the rule $A(x_1, \dots, x_n) \to \textbf{if } b \textbf{ then } \eta$ in $R_1, c \in C, \xi_0 \in T_{\Sigma \cup N(C)}(X_1), \xi_1, \dots, \xi_n \in T_{\Sigma \cup N(C)}$. Since $\bar{\beta} = \zeta$, $\beta = \beta_0[\gamma_c(\beta_1, \dots, \beta_n)]$ for some $\beta_0 \in T_{\Sigma \cup N_{\mathcal{C}}(C)}(X_1)$, $\gamma \in T_{\Sigma \cup N_{\mathcal{C}}(F)}$, and

$\beta_1, \ldots, \beta_n \in T_{\Sigma \cup N_{\mathcal{C}}(C)}$ with $\bar{\beta_i} = \xi_i$ for $0 \leqslant i \leqslant n$, and $\bar{\gamma} = \eta$. Let $\phi = \gamma^{\mathcal{B}'}$ and $\alpha = \beta_0[A^{\phi}(c)(\beta_1, \ldots, \beta_n)]$. Then

$$\bar{\alpha} = \bar{\beta}_0[A(c)(\bar{\beta}_1, \ldots, \bar{\beta}_n)] = \xi_0[A(c)(\xi_1, \ldots \xi_n)] = \xi$$

and, by the definition of R_2, (45) is in R_2 and so

$$\alpha = \beta_0[A^{\phi}(c)(\beta_1, \ldots, \beta_n)] \underset{\mathcal{C}}{\Rightarrow} \beta_0[\gamma_c[\beta_1, \ldots, \beta_n]] = \beta.$$

By (d) we have

(e) for any $\xi, \zeta \in T_{\Sigma \cup N(C)}$ and $\beta \in T_{\Sigma \cup N_{\mathcal{C}}(C)}$, if $\xi \Rightarrow^*_{\mathcal{A}} \zeta$ and $\bar{\beta} = \zeta$ then there is a tree $\alpha \in T_{\Sigma \cup N_{\mathcal{C}}(C)}$ such that $\bar{\alpha} = \xi$ and $\alpha \Rightarrow^*_{\mathcal{C}} \beta$.

By (e) and (c), we have

(f) for any $c \in C$ and $t \in T_\Sigma$, if $A_{\text{in}}(c) \Rightarrow^*_{\mathcal{A}} t$ then there is $\phi \in Q$ such that $A^{\phi}_{\text{in}}(c) \Rightarrow^*_{\mathcal{C}} t$ and $\phi = t^{\mathcal{B}}$.

By (46) and statements (c) and (f)

$$\tau(\mathcal{C}) = \tau(\mathcal{A}) \circ id_L. \tag{47}$$

Let $\mathcal{D} = (N_{\mathcal{C}} \cup \{ A_{\text{in}} \}, e_1, \Sigma, A_{\text{in}}, R_3)$, where R_3 is defined as follows. We put all elements of R_2 in R_3. Furthermore, for any $\phi \in Q_f$, and any rule $A^{\phi}_{\text{in}} \to r$ in R_2, we put the rule $A_{\text{in}} \to r$ in R_3. We obtain by direct inspection that $\tau(\mathcal{D}) = \tau(\mathcal{C})$. By (47), $\tau(\mathcal{D}) = \tau(\mathcal{A}) \circ id_L$. □

Lemma 5.2. *Let $\mathcal{A} = (N, e, \Sigma, A_{\text{in}}, R_1)$ be an IO(S) transducer and $\mathcal{H} = (\{ H_{\text{in}} \}, id_{T_\Gamma}, \Delta, H_{\text{in}}, R_{\mathcal{H}})$ be a tree homomorphism. Then we can effectively construct an IO(S) transducer $\mathcal{B} = (N_{\mathcal{B}}, e, \Sigma, A_{\text{in}}, R_2)$ and a tree homomorphism $\mathcal{J} = (\{ H_{\text{in}} \}, id_{T_\Sigma}, \Delta, H_{\text{in}}, R_{\mathcal{J}})$ such that $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \tau(\mathcal{B}) \circ \tau(\mathcal{J})$.*

Proof. Without loss of generality, we may assume that $\Gamma \subseteq \Sigma$. Let $\omega \in \Delta_0$ be arbitrary. We put all rules of $R_{\mathcal{H}}$ in $R_{\mathcal{J}}$. Furthermore, for each $\sigma \in \Sigma - \Gamma$, we put the rule

$$H_{\text{in}} \to \textbf{if } root = \sigma \textbf{ then } \omega$$

in $R_{\mathcal{J}}$. By Lemma 5.1, we take $IO(S)$ transducer $\mathcal{B} = (N_{\mathcal{B}}, e, \Sigma, A_{\text{in}}, R_2)$ such that $\tau(B) = \tau(\mathcal{A}) \circ id_{T_\Gamma}$. Hence $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \tau(\mathcal{B}) \circ \tau(\mathcal{J})$. □

Theorem 5.3. *For every storage type S, $IO(S) \circ H \subseteq IO(S)$.*

Proof. Let $S = (C, P, F, m, I, E)$ be an arbitrary storage type. Let $\mathcal{A} = (N_{\mathcal{A}}, e, \Sigma, A_{\text{in}}, R_1)$ be an $IO(S)$ transducer and $\mathcal{H} = (\{ H_{\text{in}} \}, id_{T_\Gamma}, \Delta, H_{\text{in}}, R)$ be a tree homomorphism. By Lemma 5.2, we may assume that $\Gamma = \Sigma$. We introduce $IO(S)$ transducer $\mathcal{B} = (N_{\mathcal{B}}, e, \Delta, A_{\text{in}}, R_2)$ as follows:

- $N_{\mathcal{A}} \subseteq N_{\mathcal{B}}$,

- For each $\sigma \in \Sigma$, we introduce the nonterminal $A_\sigma \in N_{\mathcal{B}}$ with the same rank as that of σ.

In order to define R_2, for each $\gamma \in T_{N_{\mathcal{A}}(F) \cup \Sigma}(X)$, we define a tree $\gamma' \in T_{N_{\mathcal{B}}(F) \cup \Delta}(X)$ as follows. Intuitively, if $\gamma \in T_\Sigma(X)$, then $\gamma' = \mathcal{H}(\gamma)$, otherwise, γ' is obtained from γ by replacing every maximal subtree $t \in T_\Sigma(X)$ by $\mathcal{H}(t)$ and then replacing every remaining $\sigma \in \Sigma$ by $A_\sigma(f)$, where f is an arbitrary instruction symbol occurring in γ.

Formally, let $\gamma = \alpha[t_1, \ldots, t_m]$, where $\alpha \in T_{N_{\mathcal{A}}(F) \cup \Sigma}(X_m)$, $m \geqslant 0$, α has no subtree $t \in T_\Sigma(X_m) - X_m$, and $t_1, \ldots, t_m \in T_\Sigma(X)$. Then $\gamma' = \bar{\alpha}[\mathcal{H}(t_1), \ldots, \mathcal{H}(t_m)]$ where $\bar{\alpha}$ is obtained from α by replacing every $\sigma \in \Sigma$ by $A_\sigma(f)$, where f is an arbitrary instruction symbol occurring in γ.

We define R_2 in the following way. For each $\sigma \in \Sigma_n$, $n \geqslant 0$, we put the rule

$$A_\sigma(x_1, \ldots, x_n) \to \textbf{if true then } \mathcal{H}(\sigma(x_1, \ldots, x_n)) \tag{48}$$

in R_2. Then for every rule

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ then } \gamma$$

in R_1, we put the rule

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ then } \gamma'$$

in R_2.

One can show the following result in a straightforward but tedious way by induction on the length of the derivations $\Rightarrow^*_{\mathcal{A}}$ and $\Rightarrow^*_{\mathcal{B}}$.

(a) For all $c \in C$ and $\alpha \in T_{N_{\mathcal{A}}(C) \cup \Sigma}$, if (i) then there is a $\beta \in T_{N_{\mathcal{B}}(C) \cup \Sigma}$ such that (ii) and (iii) and (iv), and

(b) for all $c \in C$ and $\beta \in T_{N_{\mathcal{B}}(C) \cup \Sigma}$, if (ii) then there is a $\alpha \in T_{N_{\mathcal{A}}(C) \cup \Sigma}$ such that (i) and (iii) and (iv).

Here

(i) $A_{\text{in}}(c) \Rightarrow^*_{\mathcal{A}} \alpha$.

(ii) $A_{\text{in}}(c) \Rightarrow^*_{\mathcal{B}} \beta$.

(iii) $\alpha = \bar{\alpha}[t_1, \ldots, t_m]$, where $\bar{\alpha} \in T_{N_{\mathcal{A}}(C) \cup \Sigma}(X_m)$, $m \geqslant 0$, $\bar{\alpha}$ has no subtree $t \in T_\Sigma(X_m) - X_m$, and $t_1, \ldots, t_m \in T_\Sigma$.

(iv) $\tilde{\beta} = \bar{\beta}[\mathcal{H}(t_1), \ldots, \mathcal{H}(t_m)]$, where $\bar{\beta}$ is obtained from $\bar{\alpha}$ by replacing every symbol $\sigma \in \Sigma$ by $A_\sigma(c')$ for some configuration c'. Moreover, $\tilde{\beta} \in T_{N_{\mathcal{B}}(C) \cup \Delta}$ is obtained from β by a derivation $\beta \Rightarrow^*_{\mathcal{B}} \tilde{\beta}$ where each rule applied by $\mathcal{B}$ is of the form (48) for some $\sigma \in \Sigma_n$, $n \geqslant 0$, and no rule of the form (48) is applicable for $\tilde{\beta}$.

We now show that $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \tau(\mathcal{B})$. Let $(u, w) \in \tau(\mathcal{A}) \circ \tau(\mathcal{H})$. Then there is a tree $v \in T_\Sigma$ such that $(u, v) \in \tau(\mathcal{A})$ and $(v, w) \in \tau(\mathcal{H})$. Then $A_{\text{in}}(e(u)) \Rightarrow^*_{\mathcal{A}} v$ and $\mathcal{H}(v) = w$. For $c = e(u)$ and $\alpha = v$, Condition (i) holds. By (a), there is a $\beta \in T_{N_{\mathcal{B}}(C) \cup \Sigma}$ such that Conditions (ii)–(iv) hold. By (iii), we may take $\bar{\alpha} = x_1$, $m = 1$, and $t_1 = v$. By (iv), $\bar{\beta} = x_1$ implying that $\tilde{\beta} = \mathcal{H}(v) = w$. By (iv), $\tilde{\beta} \in T_{N_{\mathcal{B}}(C) \cup \Delta}$ is obtained from β by a derivation $\beta \Rightarrow^*_{\mathcal{B}} \tilde{\beta}$. Hence

$$A_{\text{in}}(e(u)) \underset{\mathcal{B}}{\overset{*}{\Rightarrow}} \beta \underset{\mathcal{B}}{\overset{*}{\Rightarrow}} \tilde{\beta} = w.$$

Thus $(u, w) \in \tau(\mathcal{B})$.

Conversely, let $(u, w) \in \tau(\mathcal{B})$. Then $A_{\text{in}}(e(u)) \Rightarrow^*_{\mathcal{B}} w$. Let $c = e(u)$ and $\beta = w$. By (b), there is a tree $\alpha \in T_{N_{\mathcal{A}}(C)\cup\Sigma}$ such that Conditions (i), (iii) and (iv) hold. By (iv), $\tilde{\beta} = w$. By (iv) and (iii), we may take $\bar{\beta} = x_1$, $m = 1$,

$$\mathcal{H}(t_1) = w,$$

$\bar{\alpha} = x_1$, and $\alpha = t_1$. By (i)

$$A_{\text{in}}(e(u)) \underset{\mathcal{A}}{\overset{*}{\Rightarrow}} t_1.$$

Then $(u, t_1) \in \tau(\mathcal{A})$ and $(t_1, w) \in \tau(\mathcal{H})$. $\square$

Now we prove that $IO(S)$ is closed under positive look-ahead. The proof is a variation of the one of Lemma 4.4.

Theorem 5.4. *For every storage type S, $IO^+(S_{IO}) \subseteq IO(S)$.*

Proof. Let $S = (C, P, F, m, I, E)$ be an arbitrary storage type. Let $\mathcal{A} = (N^0, e, \Delta^0, A_{\text{in}}, R^0)$ be an $IO^+(S_{IO})$ transducer. Without loss of generality, we may assume that $\mathcal{A}$ is an $IO^{1+}(S_{IO})$ transducer, see Theorem 3.7. We construct an $IO(S)$ transducer $\mathcal{B}$. Then we show that $\tau(\mathcal{A}) = \tau(\mathcal{B})$.

We construct the $IO(S)$ transducer $\mathcal{B} = (N, e, \Sigma, A_{\text{in}}, R_1)$ in the following way. Let us number the rules of R^0 by the numbers $1, \ldots, r$, for some $r \geqslant 0$. Let us assume that the ith rule is of the form

$$A_i(x_1, \ldots, x_{n_i}) \rightarrow \textbf{if } b_i \textbf{ and } \langle \mathcal{L}^i \rangle \textbf{ then } \gamma_i \tag{49}$$

where $1 \leqslant i \leqslant r$, $n_i \geqslant 0$, $A_i \in N^0_{n_i}$, $b_i \in BE(P)$, $\mathcal{L}^i = (N^i, e, \Delta^i, A^i_{\text{in}}, R^i)$ is an $IO(S)$ transducer, and $\gamma_i \in T_{N^0(F)\cup\Delta^0}(X_{n_i})$. By Lemma 3.10, we may assume that the $IO(S)$ transducer $\mathcal{L}^i$ has only one rule with A^i_{in} appearing in the left-hand side, and that rule has the form

$$A^i_{\text{in}} \rightarrow \textbf{ if true then } \gamma^i. \tag{50}$$

We may assume that the sets $N^0, N^1, \ldots, N^r$ and $\Delta^0, \Delta^1, \ldots, \Delta^r$ are pairwise disjoint. Let B be a new nonterminal symbol of arity 2.

- Let $N = \bigcup_{i=0}^r N^i \cup \{ B \}$.
- Let $\Sigma = \bigcup_{i=0}^r \Delta^i$.
- We put the rule $B(x_1, x_2) \rightarrow$ **if true then** x_1 in R_1.
- For each $1 \leqslant i \leqslant r$, let the rule (49) be the ith rule of R^0, and let the rule (50) be in R^i. Then we put the rule

$$A_i(x_1, \ldots, x_{n_i}) \rightarrow \textbf{if } b_i \textbf{ then } B(f)(\gamma_i, \gamma^i) \tag{51}$$

in R_1, if $\gamma^i \notin T_{\Delta^i}$ and $f \in F$ occurs in γ^i and we put the rule

$$A_i(x_1, \ldots, x_{n_i}) \to \textbf{if } b_i \textbf{ then } \gamma_i \tag{52}$$

in R_1, if $\gamma^i \in T_{\Delta^i}$.

- We put each rule of the set $\bigcup_{i=1}^{r} R^i$ in R_1.

Note that for each $c \in C$ and each $1 \leqslant i \leqslant r$, condition b_i **and** $\langle \mathcal{L}^i \rangle$ in rule (49) is equivalent to the condition (b_i **and** there is a tree $w \in T_{\Delta^i}$ such that $\gamma_c^i \Rightarrow^*_{\mathcal{L}^i} w$) by rule (50), where condition b_i and tree γ^i also appear in rule (51). Thus the application of rule (49) of $\mathcal{A}$ is equivalent to the application of rule (52) or is equivalent to the subsequent application of rule (51), some rules in R^i, and rule $B(x_1, x_2) \to$ **if true then** x_1 of $\mathcal{B}$. Hence $\tau(\mathcal{A}) = \tau(\mathcal{B})$. We now give a formal proof.

Claim 5.5. *For all* $A \in N_n^0$, $n \geqslant 0$, $c \in C$, $t_1, \ldots, t_n \in T_{\Delta^0}$, *and* $s \in T_{\Delta^0}$, $A(c)(t_1, \ldots, t_n) \Rightarrow^*_{\mathcal{A}} s$ *if and only if* $A(c)(t_1, \ldots, t_n) \Rightarrow^*_{\mathcal{B}} s$.

Proof. ($\Rightarrow$) Let $A(c)(t_1, \ldots, t_n) \Rightarrow^l_{\mathcal{A}} s$ for some $l \geqslant 1$. We show that $A(c)(t_1, \ldots, t_n) \Rightarrow^*_{\mathcal{B}} s$ by induction on l.

Base case: Let $l = 1$. Then $A = A_i$ for some $1 \leqslant i \leqslant r$ and we apply rule (49). Hence $m(\langle \mathcal{L}^i \rangle)(c) = \textbf{true}$. That is, $A^i_{\text{in}}(c) \Rightarrow_{\mathcal{L}^i} \gamma_c^i \Rightarrow^*_{\mathcal{L}^i} w$ for some $w \in T_{\Delta^i}$. By the definition of R_1, if $\gamma^i \notin T_{\Delta^i}$, then rule (51) is in R_1, and

$A(c)(t_1, \ldots, t_n) \Rightarrow_{\mathcal{B}} B(m(f)(c))((\gamma_i[t_1, \ldots, t_n])_c, \gamma_c^i) \Rightarrow^*_{\mathcal{B}}$
$B(m(f)(c))((\gamma_i[t_1, \ldots, t_n])_c, w) \Rightarrow_{\mathcal{B}} (\gamma_i[t_1, \ldots, t_n])_c$.

If $\gamma^i \in T_{\Delta^i}$, then rule (52) is in R_1 and $A(c)(t_1, \ldots, t_n) \Rightarrow_{\mathcal{B}} (\gamma_i[t_1, \ldots, t_n])_c$.

Induction step: Let $l > 1$. Then $A = A_i$ for some $1 \leqslant i \leqslant r$, $n = n_i$, and we apply rule (49) in the first step of $A(c)(t_1, \ldots, t_n) \Rightarrow^l_{\mathcal{A}} s$. Hence

$$m(b_i)(c) = \textbf{true} \tag{53}$$

and $m(\langle \mathcal{L}^i \rangle)(c) = \textbf{true}$. That is, $A^i_{\text{in}}(c) \Rightarrow_{\mathcal{L}^i} \gamma_c^i \Rightarrow^*_{\mathcal{L}^i} w$ for some $w \in T_{\Delta^i}$. Furthermore,

$$A(c)(t_1, \ldots, t_n) \underset{\mathcal{A}}{\Rightarrow} (\gamma_i[t_1, \ldots, t_n])_c \Rightarrow^{l-1}_{\mathcal{A}} s. \tag{54}$$

Then, there are $\delta_1, \ldots, \delta_v \in T_{\Delta^0 \cup N^0(C)}$, $w_1, \ldots, w_v \in T_{\Delta^0}$, $v \geqslant 1$, such that

(a) $(\gamma_i[t_1, \ldots, t_n])_c = \delta_1$ and $w_1 = s$,

(b) for each $j = 1, \ldots, v$, $\delta_j \Rightarrow^*_{\mathcal{A}} w_j$ where

- $\delta_j = u_j[A_{j1}(c_{j1})(\delta_{\mu_{j11}}, \ldots, \delta_{\mu_{j1\kappa_{j1}}}), \ldots, A_{jk_j}(c_{jk_j})(\delta_{\mu_{jk_j1}}, \ldots, \delta_{\mu_{jk_j\kappa_{jk_j}}})]$ for some $u_j \in T_{\Delta^0}(X_{k_j})$, $k_j \geqslant 1$, $A_{j1}, \ldots, A_{jk_j} \in N^0$, $c_{j1}, \ldots, c_{jk_j} \in C$, and $\mu_{j11}, \ldots, \mu_{j1\kappa_{j1}}, \ldots, \mu_{jk_j1}, \ldots, \mu_{jk_j\kappa_{jk_j}} \in \{j+1, \ldots, v\}$,
- there are $\eta_{j1}, \ldots, \eta_{jk_j} \in T_{\Delta^0}$ such that for each $j = 1, \ldots, v$,

  $A_{j1}(c_{j1})(w_{\mu_{j11}}, \ldots, w_{\mu_{j1\kappa_{j1}}}) \Rightarrow^{l_{j1}}_{\mathcal{A}} \eta_{j1} \in T_{\Delta^0}$, where $l_{j1} \leqslant l - 1$,

  ...

  $A_{jk_j}(c_{jk_j})(w_{\mu_{jk_j1}}, \ldots, w_{\mu_{jk_j\kappa_{jk_j}}}) \Rightarrow^{l_{jk_j}}_{\mathcal{A}} \eta_{jk_j} \in T_{\Delta^0}$, where $l_{jk_j} \leqslant l - 1$, and

- $w_j = u_j[\eta_{j1}, \ldots, \eta_{jk_j}]$ for $j = 1, \ldots, v$.

By the induction hypothesis, for each $j = 1, \ldots, v$,

$A_{j1}(c_{j1})(w_{\mu_{j11}}, \ldots, w_{\mu_{j1\kappa_{j1}}}) \Rightarrow^*_{\mathcal{B}} \eta_{j1}$,

...

$A_{jk_j}(c_{jk_j})(w_{\mu_{jk_j1}}, \ldots, w_{\mu_{jk_j\kappa_{jk_j}}}) \Rightarrow^*_{\mathcal{B}} \eta_{jk_j}$.

Hence, using an obvious induction on $v - j$, $\delta_j \Rightarrow^*_{\mathcal{B}} w_j$ for $j = 1, \ldots, v$. By (a)

$$(\gamma_i[t_1, \ldots, t_n])_c \overset{*}{\underset{\mathcal{B}}{\Rightarrow}} s. \tag{55}$$

By the definition of R_1, we now distinguish two cases.

Case 1: $\gamma^i \notin T_{\Delta^i}$. Then rule (51) is in R_1. Thus by (53) and (55) we have $A(c)(t_1, \ldots, t_n) \Rightarrow_{\mathcal{B}} B(m(f)(c))((\gamma_i[t_1, \ldots, t_n])_c, \gamma^i_c) \Rightarrow^*_{\mathcal{B}} B(m(f)(c))(s, w) \Rightarrow_{\mathcal{B}} s$.

Case 2: $\gamma^i \in T_{\Delta^i}$. Then rule (52) is in R_1 and $A(c)(t_1, \ldots, t_n) \Rightarrow_{\mathcal{B}} (\gamma_i[t_1, \ldots, t_n])_c \Rightarrow^*_{\mathcal{B}} s$.

($\Leftarrow$) Let $A(c)(t_1, \ldots, t_n) \Rightarrow^l_{\mathcal{B}} s$ for some $l \geqslant 1$. We show that $A(c)(t_1, \ldots, t_n) \Rightarrow^*_{\mathcal{A}} s$ by induction on l.

Base case: Let $l = 1$. Then $A = A_i$ for some $1 \leqslant i \leqslant r$, $n = n_i$, and $\mathcal{B}$ applies rule (52). Hence $m(b_i)(c) = \mathbf{true}$. By the definition of R_1, $\gamma^i \in T_{\Delta^i}$. Thus $m(\langle \mathcal{L}^i \rangle)(c) = \mathbf{true}$. The definition of R_1 also implies that rule (49) is in R^0. Hence $A(c)(t_1, \ldots, t_n) \Rightarrow_{\mathcal{A}} (\gamma_i[t_1, \ldots, t_n])_c$.

Induction step: Let $l > 1$. Then $A = A_i$ for some $1 \leqslant i \leqslant r$, and $n = n_i$. By the definition of R_1, we distinguish two cases.

Case 1: $\mathcal{B}$ applies rule (51) in the first step of $A(c)(t_1, \ldots, t_n) \Rightarrow^l_{\mathcal{B}} s$. That is,

$$A(c)(t_1, \ldots, t_n) \underset{\mathcal{B}}{\Rightarrow} B(m(f)(c))((\gamma_i[t_1, \ldots, t_n])_c, \gamma^i_c) \Rightarrow^{l-1}_{\mathcal{B}} s. \tag{56}$$

Then

(i) rule (49) is in R^0,
(ii) $m(b_i)(c) = \mathbf{true}$, and
(iii) $A^i_{\text{in}}(c) \Rightarrow_{\mathcal{L}^i} \gamma^i_c \Rightarrow^*_{\mathcal{L}^i} w$ for some $w \in T_{\Delta^i}$.
(iv) $(\gamma_i[t_1, \ldots, t_n])_c \Rightarrow^{\mu}_{\mathcal{B}} s$ for some $\mu \leqslant l - 1$.

By (iii), $m(\langle \mathcal{L}^i \rangle)(c) = \mathbf{true}$. Thus, by (i) and (ii),

$$A(c)(t_1, \ldots, t_n) \underset{\mathcal{A}}{\Rightarrow} (\gamma_i[t_1, \ldots, t_n])_c. \tag{57}$$

From (iv) we can conclude by induction, in exactly the same way as in points (a) and (b) in the ($\Rightarrow$)-part of this proof, that $(\gamma_i[t_1, \ldots, t_n])_c \Rightarrow^*_{\mathcal{A}} s$. By (57) we get that

$$A(c)(t_1, \ldots, t_n) \underset{\mathcal{A}}{\Rightarrow} (\gamma_i[t_1, \ldots, t_n])_c \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} s.$$

Case 2: $\mathcal{B}$ applies rule (52) in the first step of $A(c)(t_1, \ldots, t_n) \Rightarrow^l_{\mathcal{B}} s$. This case is similar to Case 1. □

From Claim 5.5 it follows, taking $A = A_{\rm in}$ and $c = e(u)$, that $\tau(\mathcal{A}) = \tau(\mathcal{B})$. □

We note that the deterministic version of Theorem 5.4 is not true for $S = TR$ (see Corollary 5.20 of [7]). In fact, it is not even true that $DIO^+(S_{IO}) = DIO(S) \circ LH$ because $DIO(S)$ is closed under composition with LH, for $S = TR$ (see Theorem 7.6(2) of [7]).

From Theorems 5.3 and 5.4 (and the obvious facts that $IO(S) \subseteq IO^+(S_{IO})$ and $IO(S) \subseteq IO(S) \circ H$) we immediately obtain the following result.

Corollary 5.6. *For every storage type S,* $IO^+(S_{IO}) = IO(S) = IO(S) \circ LH = IO(S) \circ H$.

6. *OI* transducers

We generalize the nondeterministic part of Lemma 4.1 to *OI* transducers. That is, we show that for every storage type S, $OI^+(S_{OI}) \supseteq OI(S) \circ LH$. We also show that for every storage type S, $OI(S)$ is closed under positive look-ahead, and hence is closed under composition with linear tree homomorphisms. That is, for every storage type S, $OI^+(S_{OI}) = OI(S) = OI(S) \circ LH$.

Note that for the trivial storage type S_0, the closure of $OI(S_0)$ under linear tree homomorphisms is shown in [17]. However, $OI(S_0)$ is not closed under tree homomorphisms (see Example 6.7 in [6]), in contrast with $IO(S_0)$.

First we show that for every storage type S, $OI^+(S_{OI}) \supseteq OI(S) \circ LH$. We intuitively discuss the main difference between the *RT* and *OI* cases. We illustrate by an example why the straightforward generalization of the construction used in the proof of Lemma 4.1 does not work for *OI*.

Example 6.1. Let $S_0 = (\{c\}, \emptyset, \{id\}, m, \{c\}, \{id_{\{c\}}\})$ be the trivial storage type. Consider the $OI(S_0)$ transducer $\mathcal{A} = (N, e, \Sigma, A_{\rm in}, R_1)$, with $e = id_{\{c\}}$ and

- $N = \{A_{\rm in}, A, B, C\}$, $A_{\rm in}, B, C$ have rank 0 and A has rank 2,
- $\Sigma = \Sigma_1 \cup \Sigma_0$, $\Sigma_1 = \{\sigma_1, \sigma_2\}$, $\Sigma_0 = \{b\}$, and
- R_1 consists of the following rules.
  $A_{\rm in} \to$ **if true then** $A(id)(\sigma_1(B(id)), \sigma_2(C(id)))$,
  $A(x_1, x_2) \to$ **if true then** x_1,
  $A(x_1, x_2) \to$ **if true then** x_2, and
  $B \to$ **if true then** b.

The one and only successful derivation of $\mathcal{A}$ is the following:

$$A_{\rm in}(c) \underset{\mathcal{A}}{\Rightarrow} A(c)(\sigma_1(B(c)), \sigma_2(C(c))) \underset{\mathcal{A}}{\Rightarrow} \sigma_1(B(c)) \underset{\mathcal{A}}{\Rightarrow} \sigma_1(b). \tag{58}$$

Thus, transducer $\mathcal{A}$ induces the transformation $\tau(\mathcal{A}) = \{(c, \sigma_1(b))\}$.

Let $\mathcal{H} = (\{H_{\rm in}\}, id_{T_\Sigma}, \Delta, H_{\rm in}, R_2)$ be a linear tree homomorphism, where

- $\Delta = \Delta_0 = \{a_1, a_2\}$, and
- R_2 consists of the following rules.
  $H_{\rm in} \to$ **if** *root* $= \sigma_1$ **then** a_1,

$H_{\text{in}} \to$ **if** *root* $= \sigma_2$ **then** a_2, and
$H_{\text{in}} \to$ **if** *root* $= b$ **then** a_1.
Obviously, $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \{ (c, a_1) \}$.

In order to construct $\mathcal{A} \circ \mathcal{H}$ as in the proof of Lemma 4.1, we now define an $OI(S)$ transducer $\mathcal{L}$. It will appear as a look-ahead transducer in a rule of $\mathcal{A} \circ \mathcal{H}$. Let $\mathcal{L} = (N \cup \{ B_{\text{in}} \}, e, \Sigma, B_{\text{in}}, R_1')$ be an $OI(S)$ transducer, where B_{in} is a new nonterminal and R_1' consists of the following rules:
$B_{\text{in}} \to$ **if true then** $A(id)(\sigma_1(B(id)), \sigma_2(C(id)))$,
$A_{\text{in}} \to$ **if true then** $A(id)(\sigma_1(B(id)), \sigma_2(C(id)))$,
$A(x_1, x_2) \to$ **if true then** x_1,
$A(x_1, x_2) \to$ **if true then** x_2, and
$B \to$ **if true then** b.
Observe that $B_{\text{in}}(c) \Rightarrow_{\mathcal{L}} A(c)(\sigma_1(B(c)), \sigma_2(C(c))) \Rightarrow_{\mathcal{L}} \sigma_1(B(c)) \Rightarrow_{\mathcal{L}} \sigma_1(b)$. Hence

$$m(\langle \mathcal{L} \rangle)(c) = \textbf{true}. \tag{59}$$

Now the straightforward generalization of the construction used in the proof of Lemma 4.1 gives the $OI^+(S_{OI})$ transducer $\mathcal{A} \circ \mathcal{H} = (N, e, \Delta, A_{\text{in}}, R_3)$, where R_3 consists of the following rules.
$A_{\text{in}} \to$ **if true and** $\langle \mathcal{L} \rangle$ **then** $A(id)(a_1, a_2)$,
$A(x_1, x_2) \to$ **if true then** x_1,
$A(x_1, x_2) \to$ **if true then** x_2, and
$B \to$ **if true then** a_1.
By (59), we have $A_{\text{in}}(c) \Rightarrow_{\mathcal{A} \circ \mathcal{H}} A(c)(a_1, a_2)$. Hence we have the following derivations:
$A_{\text{in}}(c) \Rightarrow_{\mathcal{A} \circ \mathcal{H}} A(c)(a_1, a_2) \Rightarrow_{\mathcal{A} \circ \mathcal{H}} a_1$,
$A_{\text{in}}(c) \Rightarrow_{\mathcal{A} \circ \mathcal{H}} A(c)(a_1, a_2) \Rightarrow_{\mathcal{A} \circ \mathcal{H}} a_2$.
Thus $\tau(\mathcal{A} \circ \mathcal{H}) = \{ (c, a_1), (c, a_2) \}$. Hence $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) \neq \tau(\mathcal{A} \circ \mathcal{H})$. This ends Example 6.1.

Let $\mathcal{A} = (N, e, \Sigma, A_{\text{in}}, R_1)$ be an $OI(S)$ transducer. Let $t \in T_{N(C) \cup \Sigma}$ be a tree, and let

$$t = u_1 \underset{\mathcal{A}}{\Rightarrow} u_2 \underset{\mathcal{A}}{\Rightarrow} \cdots \underset{\mathcal{A}}{\Rightarrow} u_k = u, \;\; k \geqslant 1 \tag{60}$$

for some $u_1, \ldots, u_{k-1} \in T_{N(C) \cup \Sigma}$ and $u_k \in T_\Sigma$. Transducer $\mathcal{A}$ might delete some subtree p of t along the derivation (60) such that there is no tree $q \in T_\Sigma$ with $p \Rightarrow_{\mathcal{A}}^* q$. An example of this phenomenon is derivation (58) because the subtree $\sigma_2(C(c))$ is deleted. Assume that along (60), we derive a terminal tree from the subtree

$$A(c)(\alpha_1, \ldots, \alpha_n) \tag{61}$$

of u_j, $1 \leqslant j \leqslant k$. Then there is a set $V \subseteq X_n$ of variables such that along (60),
(a) from subtree (61), we derive a tree $s[\alpha_1, \ldots, \alpha_n]$ for some $s \in T_\Sigma(V)$, and
(b) for each $x_i \in V$, from the subtree α_i we derive a tree $s_i \in T_\Sigma$ (or several such trees).

In the light of this observation, for any set $V \subseteq X_n$, and nonterminal $A \in N$, we introduce a new nonterminal A^V. Moreover, we modify derivation (60). We substitute A^V for A in subtree (61) of u_j, $1 \leqslant j \leqslant k$. Then A^V has the following "meaning". For any $1 \leqslant j \leqslant k$ and subtree

$$A^V(c)(\alpha_1, \ldots, \alpha_n)$$

of u_j and for any variable $x_i \in V$, there is a tree $s_i \in T_\Sigma$ such that $\alpha_i \Rightarrow^*_{\mathcal{A}} s_i$. Furthermore, there is a tree $s \in T_\Sigma(V)$ such that $A(c)(x_1, \ldots, x_n) \Rightarrow^*_{\mathcal{A}} s$.

Let $\mathcal{H}$ be a linear tree homomorphism. In the proof of Theorem 6.3 we will define the $OI^+(S_{OI})$ transducer $\mathcal{A} \circ \mathcal{H}$ by modifying the construction of Lemma 4.1. When constructing the rules of $\mathcal{A} \circ \mathcal{H}$, we replace the nonterminals A of $\mathcal{A}$ by nonterminals A^V. The "meaning" of A^V will be forced by the look-ahead tests of $\mathcal{A} \circ \mathcal{H}$.

Example 6.1, continued.

For transducers $\mathcal{A}$ and $\mathcal{H}$ of Example 6.1 we will construct the transducer $\mathcal{A} \circ \mathcal{H} = (J, e, \Delta, A^{\emptyset}_{\text{in}}, R_3)$, where $J_0 = \{ A^{\emptyset}_{\text{in}}, B^{\emptyset}, C^{\emptyset} \}$ and $J_2 = \{ A^{\{x_1\}}, A^{\{x_2\}}, A^{\{x_1,x_2\}} \}$. In order to construct the rules of $\mathcal{A} \circ \mathcal{H}$, we now define $OI(S)$ transducers $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$, $\mathcal{L}_4$, and $\mathcal{L}_5$. They will appear as look-ahead transducers in the rules of $\mathcal{A} \circ \mathcal{H}$. Intuitively, $\mathcal{L}_1$ corresponds to $A^{\{x_1\}}(id)$, where $A(id)$ appears in the right-hand side of the first rule of R_1. Similarly, $\mathcal{L}_2$ and $\mathcal{L}_3$ correspond to $A^{\{x_2\}}(id)$ and $A^{\{x_1,x_2\}}(id)$, respectively, and $\mathcal{L}_4$ and $\mathcal{L}_5$ to $B^{\emptyset}(id)$ and $C^{\emptyset}(id)$, respectively. Transducer $\mathcal{L}_1$ is defined in such a way that the test $\langle \mathcal{L}_1 \rangle$ is true on c if and only if there is a tree $s \in T_\Sigma(\{x_1\})$ such that $A(c)(x_1, x_2) \Rightarrow^*_{\mathcal{A}} s$. Similarly, $\langle \mathcal{L}_2 \rangle$ and $\langle \mathcal{L}_3 \rangle$ test whether there is tree s in $T_\Sigma(\{x_2\})$ and $s \in T_\Sigma(\{x_1, x_2\})$, respectively, such that $A(c)(x_1, x_2) \Rightarrow^*_{\mathcal{A}} s$. The meanings of $\langle \mathcal{L}_4 \rangle$ and $\langle \mathcal{L}_5 \rangle$ are analogous.

Let ω be a new 0-ary terminal symbol. Let $\mathcal{L}_1 = (N \cup \{ D_{\text{in}}, D_1, D_2 \}, e, \Sigma \cup \{\omega\}, D_{\text{in}}, R'_1)$ be an $OI(S)$ transducer, where D_{in}, D_1, D_2 are new nonterminals of rank 0, and R'_1 consists of the following rules:

$D_{\text{in}} \to$ **if true then** $A(id)(D_1(id), D_2(id))$,
$A(x_1, x_2) \to$ **if true then** x_1,
$A(x_1, x_2) \to$ **if true then** x_2, and
$D_1 \to$ **if true then** ω.

Let $\mathcal{L}_2 = (N \cup \{ D_{\text{in}}, D_1, D_2 \}, e, \Sigma \cup \{ \omega \}, D_{\text{in}}, R'_2)$ be an $OI(S)$ transducer, where R'_2 consists of the following rules:

$D_{\text{in}} \to$ **if true then** $A(id)(D_1(id), D_2(id))$,
$A(x_1, x_2) \to$ **if true then** x_1,
$A(x_1, x_2) \to$ **if true then** x_2, and
$D_2 \to$ **if true then** ω.

Let $\mathcal{L}_3 = (N \cup \{ D_{\text{in}}, D_1, D_2 \}, e, \Sigma \cup \{ \omega \}, D_{\text{in}}, R'_3)$ be an $OI(S)$ transducer, where R'_3 consists of the following rules:

$D_{\text{in}} \to$ **if true then** $A(id)(D_1(id), D_2(id))$,
$A(x_1, x_2) \to$ **if true then** x_1,
$A(x_1, x_2) \to$ **if true then** x_2,
$D_1 \to$ **if true then** ω and
$D_2 \to$ **if true then** ω.

Let $\mathcal{L}_4 = (N \cup \{ D_{\text{in}} \}, e, \Sigma, D_{\text{in}}, R'_4)$ be an $OI(S)$ transducer, where R'_4 consists of the following rules:

$D_{\text{in}} \to$ **if true then** $B(id)$ and
$B \to$ **if true then** b.

Let $\mathcal{L}_5 = (N \cup \{ D_{\text{in}} \}, e, \Sigma, D_{\text{in}}, R'_5)$ be an $OI(S)$ transducer, where R'_5 consists of the only rule $D_{\text{in}} \to$ **if true then** $C(id)$.

Observe that $m(\langle \mathcal{L}_1 \rangle)(c) = m(\langle \mathcal{L}_2 \rangle)(c) = m(\langle \mathcal{L}_3 \rangle)(c) = m(\langle \mathcal{L}_4 \rangle)(c) =$ **true** and $m(\langle \mathcal{L}_5 \rangle)(c) =$ **false**.

Now a modified generalization of the construction used in the proof of Lemma 4.1 gives the $OI^+(S_{OI})$ transducer $\mathcal{A} \circ \mathcal{H} = (J, e, \Delta, A^{\emptyset}_{\text{in}}, R_3)$, where R_3 consists of the following rules:

$A^{\emptyset}_{\text{in}} \to$ **if** $\langle \mathcal{L}_1 \rangle$ **and** $\langle \mathcal{L}_4 \rangle$ **then** $A^{\{x_1\}}(id)(a_1, a_2)$,
$A^{\emptyset}_{\text{in}} \to$ **if** $\langle \mathcal{L}_2 \rangle$ **and** $\langle \mathcal{L}_5 \rangle$ **then** $A^{\{x_2\}}(id)(a_1, a_2)$,
$A^{\emptyset}_{\text{in}} \to$ **if** $\langle \mathcal{L}_3 \rangle$ **and** $\langle \mathcal{L}_4 \rangle$ **and** $\langle \mathcal{L}_5 \rangle$ **then** $A^{\{x_1,x_2\}}(id)(a_1, a_2)$,
$A^{\{x_1\}}(x_1, x_2) \to$ **if true then** x_1,
$A^{\{x_2\}}(x_1, x_2) \to$ **if true then** x_2,
$A^{\{x_1,x_2\}}(x_1, x_2) \to$ **if true then** x_1, and
$A^{\{x_1,x_2\}}(x_1, x_2) \to$ **if true then** x_2.

Since $\langle \mathcal{L}_5 \rangle$ is false and the other look-ahead tests are true, we have the only $\Rightarrow_{\mathcal{A}}$-derivation resulting in a tree over the terminal alphabet Δ:

$A^{\emptyset}_{\text{in}}(c) \Rightarrow_{\mathcal{A}\circ\mathcal{H}} A^{\{x_1\}}(c)(a_1, a_2) \Rightarrow_{\mathcal{A}\circ\mathcal{H}} a_1$.

Thus $\tau(\mathcal{A} \circ \mathcal{H}) = \{ (c, a_1) \}$. Hence $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \tau(\mathcal{A} \circ \mathcal{H})$.

In order to prove $OI^+(S_{OI}) \supseteq OI(S) \circ LH$, we need the counterpart of Lemma 5.2.

Lemma 6.2. *Let $\mathcal{A} = (N, e, \Sigma, A_{\text{in}}, R_1)$ be an $OI(S)$ transducer and $\mathcal{H} = (\{H_{\text{in}}\}, id_{T_\Gamma}, \Delta, H_{\text{in}}, R_2)$ be a linear tree homomorphism. Then we can effectively construct an $OI(S)$ transducer $\mathcal{B} = (N', e, \Gamma, A_{\text{in}}, R_{\mathcal{B}})$ such that $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \tau(\mathcal{B}) \circ \tau(\mathcal{H})$.*

Proof. First, we construct an $OI(S)$ transducer $\mathcal{A}' = (N', e, \Sigma, A_{\text{in}}, R'_1)$ equivalent to $\mathcal{A}$ such that for every rule $A(x_1, \ldots, x_n) \to$ **if** b **then** γ in R'_1, there is no terminal symbol in any subtree of γ with nonterminal root (see the first step of the proof of Lemma 5.3 in [8]). Let us observe that a terminal symbol occurs in a tree derived by $\mathcal{A}'$ from $A_{\text{in}}(e(u))$ (where u is an arbitrary input element) if and only if this terminal symbol occurs in a rule of R'_1 applied along the derivation.

Then, let us remove all the rules from R'_1 in which occurs one of the terminal symbols in $\Sigma - \Gamma$ and in this way, we obtain the set of rules $R_{\mathcal{B}}$. By the previous observation $\tau(\mathcal{A}) \circ \tau(\mathcal{H}) = \tau(\mathcal{B}) \circ \tau(\mathcal{H})$. □

Theorem 6.3. *For every storage type S, $OI^+(S_{OI}) \supseteq OI(S) \circ LH$.*

Proof. Let $S = (C, P, F, m, I, E)$ be an arbitrary storage type, $\mathcal{A} = (N, e, \Sigma, A_{\text{in}}, R_1)$ be an $OI(S)$ transducer and $\mathcal{H} = (\{H_{\text{in}}\}, id_{T_\Gamma}, \Delta, H_{\text{in}}, R_2)$ be a linear tree homomorphism. According to Lemma 6.2, we may assume $\Sigma = \Gamma$. We introduce the ranked alphabet J, where $J_n = \{ A^V \mid A \in N_n, V \subseteq X_n \}$ for $n \geqslant 0$.

Definition 6.4. Let $k \geqslant 0$, $\beta \in T_{J(F)\cup\Sigma}(X_k)$, and $\gamma \in T_{J(C)\cup\Sigma}$. We define the tree $\bar{\beta}$ from β by replacing each symbol $A^V(f)$ in $J(F)$ by the symbol $A(f)$ in $N(F)$. Similarly, we define the tree $\bar{\gamma}$ from γ by replacing each symbol $A^V(c)$ in $J(C)$ by the symbol $A(c)$ in $N(C)$.

Before defining the $OI^+(S_{OI})$ transducer $\mathcal{A} \circ \mathcal{H}$ that induces $\tau(\mathcal{A}) \circ \tau(\mathcal{H})$, we define the look-ahead tests that are needed in the rules of $\mathcal{A} \circ \mathcal{H}$. Let $f \in F$ be an arbitrary instruction. We now introduce the look-ahead test $\langle \mathcal{L}_f \rangle \in P_{OI}$ so that for each configuration $c \in C$, $\langle \mathcal{L}_f \rangle$ is true on c if and only if f is defined on c. We define $OI(S)$ transducer $\mathcal{L}_f = (N', e, \Sigma', B_{\text{in}}, R_1')$ as follows.

(i) $N' = \{ B_{\text{in}}, B \}$, where B_{in} and B are 0-ary nonterminals.
(ii) $\Sigma' = \{ \omega \}$, where ω is a new 0-ary terminal symbol.
(iii) R_1' consists of the following rules:

$$B_{\text{in}} \to \textbf{if true then } B(f)$$

and

$$B \to \textbf{if true then } \omega.$$

Claim 6.5. *For each configuration* $c \in C$, $m_{OI}(\langle \mathcal{L}_f \rangle)(c) = \textbf{true}$ *if and only if* $c \in dom(m(f))$.

Proof. It follows directly from the definition of $\mathcal{L}_f$. □

Let $k \geqslant 0$ and $\beta \in T_{J(F)\cup\Sigma}(X_k)$. We now introduce the test $d_F(\beta) \in BE(P_{OI})$ so that the following holds. For each configuration c, the test $d_F(\beta)$ is true on c if and only if all instructions f occurring in β are defined on c.

Definition 6.6. Let $k \geqslant 0$ and $\beta \in T_{J(F)\cup\Sigma}(X_k)$. We define the test $d_F(\beta) \in BE(P_{OI})$ as

$$d_F(\beta) = \bigwedge(\langle \mathcal{L}_f \rangle \mid f \in F \text{ occurs in } \beta).$$

Definition 6.6 implies the following result.

Claim 6.7. *Let* $k \geqslant 0$ *and* $\beta \in T_{J(F)\cup\Sigma}(X_k)$. *For each configuration* $c \in C$, $m_{OI}(d_F(\beta))(c) = \textbf{true}$ *if and only if* β_c *is defined.*

Let $n \geqslant 0$, $A \in N_n$, $V \subseteq X_n$, $f \in F$. We now introduce a look-ahead test so that for each configuration $c \in C$, the look-ahead test is true on the configuration c if and only if there is a tree $s_0 \in T_\Sigma(V)$ such that

$$A(m(f)(c))(x_1, \ldots, x_n) \underset{\mathcal{A}}{\overset{*}{\Rightarrow}} s_0.$$

To this end, let the $OI(S)$ transducer $\mathcal{L}_{A,V,f} = (N', e, \Sigma', B_{\text{in}}, R_1')$ be defined as follows:

(i) $N' = N \cup \{ B_{\text{in}}, B_1, \ldots, B_n \}$, where $B_{\text{in}}, B_1, \ldots, B_n$ are new 0-ary nonterminals.

(ii) $\Sigma' = \Sigma \cup \{\omega\}$, where ω is a 0-ary terminal symbol, as before.
(iii) We define R_1' from R_1 by adding the rules

$$B_{\text{in}} \to \textbf{if true then } A(f)(B_1(f), \ldots, B_n(f))$$

and

$$B_i \to \textbf{if true then } \omega \quad \text{for each } x_i \in V.$$

Claim 6.8. *For each configuration $c \in C$, $m_{OI}(\langle \mathcal{L}_{A,V,f} \rangle)(c) =$ **true** if and only if there is a tree $s_0 \in T_\Sigma(V)$ such that $A(m(f(c)))(x_1, \ldots, x_n) \Rightarrow^*_{\mathcal{A}} s_0$.*

Proof. It is straightfoward to show that for all $t_0 \in T_{\Sigma'}$ and $c \in C$, $B_{\text{in}}(c) \Rightarrow^*_{\mathcal{L}_{A,V,f}} t_0$ if and only if $t_0 = s_0[\omega, \ldots, \omega]$ for some $s_0 \in T_\Sigma(V)$ such that $A(m(f)(c))(x_1, \ldots, x_n) \Rightarrow^*_{\mathcal{A}} s_0$. This proves the claim. □

Let $k \geqslant 0$, $W \subseteq X_k$, and $\beta \in T_{J(F)\cup\Sigma}(X_k)$. We now introduce the test $b_F(\beta, W) \in BE(P_{OI})$ so that intuitively the following holds. For each configuration c, the test $b_F(\beta, W)$ is true on c if and only if there is a derivation $\bar{\beta}_c \Rightarrow^*_{\mathcal{A}} s \in T_\Sigma(W)$ in which the "meaning" of each A^V occurring in β is respected.

Definition 6.9. Let $k \geqslant 0$, $W \subseteq X_k$, and $\beta \in T_{J(F)\cup\Sigma}(X_k)$. We define the test $b_F(\beta, W) \in BE(P_{OI})$ by tree induction on β.
(i) Assume that $\beta \in X_k$. If $\beta \in W$ then $b_F(\beta, W) =$ **true** else $b_F(\beta, W) =$ **false**.
(ii) Assume that $\beta = \sigma(\beta_1, \ldots, \beta_n)$ for some $n \geqslant 0$, $\sigma \in \Sigma_n$, $\beta_1, \ldots, \beta_n \in T_{J(F)\cup\Sigma}(X_k)$. Then

$$b_F(\beta, W) = \bigwedge (b_F(\beta_i, W) \mid 1 \leqslant i \leqslant n).$$

(iii) Assume that $\beta = A^V(f)(\beta_1, \ldots, \beta_n)$ for some $n \geqslant 0$, $A \in N_n$, $V \subseteq X_n$, $f \in F$, and $\beta_1, \ldots, \beta_n \in T_{J(F)\cup\Sigma}(X_k)$. Then

$$b_F(\beta, W) = \langle \mathcal{L}_{A,V,f} \rangle \textbf{ and } \bigwedge (b_F(\beta_i, W) \mid x_i \in V).$$

Definition 6.9 implies the following result.

Claim 6.10. *Let $k \geqslant 0$, $W \subseteq X_k$, and $\beta \in T_{J(F)\cup\Sigma}(X_k)$. Either $m_{OI}(b_F(\beta, W))(c) =$ **false** for all $c \in C$ or*

$$m_{OI}(b_F(\beta, W)) = m_{OI}\left(\bigwedge (\langle \mathcal{L}_{A_j,V_j,f_j} \rangle \mid j = 1, \ldots, n)\right)$$

for some $n \geqslant 0$, and $A_j \in N$, $V_j \subseteq X$, and $f_j \in F$ for $1 \leqslant j \leqslant n$.

Let $k \geqslant 0$, $W \subseteq X_k$, and $\beta \in T_{J(C)\cup\Sigma}(X_k)$. We now introduce the Boolean value $b_C(\beta, W) \in \{$**true**, **false**$\}$ so that it is true if and only if there is a derivation $\bar{\beta} \Rightarrow^*_{\mathcal{A}} s \in T_\Sigma(W)$ in which the "meaning" of each A^V occurring in β is respected.

Definition 6.11. Let $k \geqslant 0$, $W \subseteq X_k$, and $\beta \in T_{J(C)\cup\Sigma}(X_k)$. We define the Boolean value $b_C(\beta, W)$ by tree induction on β.

(i) Assume that $\beta \in X_k$. If $\beta \in W$, then $b_C(\beta, W) =$ **true** else $b_C(\beta, W) =$ **false**.

(ii) Assume that $\beta = \sigma(\beta_1, \ldots, \beta_n)$ for some $n \geqslant 0$, $\sigma \in \Sigma_n$, $\beta_1, \ldots, \beta_n \in T_{J(C)\cup\Sigma}(X_k)$. Then

$$b_C(\beta, W) = \bigwedge(b_C(\beta_i, W) \mid 1 \leqslant i \leqslant n).$$

(iii) Assume that $\beta = A^V(c)(\beta_1, \ldots, \beta_n)$ for some $n \geqslant 0$, $A \in N_n$, $V \subseteq X_n$, $c \in C$, and $\beta_1, \ldots, \beta_n \in T_{J(C)\cup\Sigma}(X_k)$. Then $b_C(\beta, W) =$ **true** if and only if there is a tree $s_0 \in T_\Sigma(V)$ such that $A(c)(x_1, \ldots, x_n) \Rightarrow^*_{\mathcal{A}} s_0$ and $\bigwedge(b_C(\beta_i, W) \mid x_i \in V) =$ **true**.

Claim 6.12. *Let $k \geqslant 0$, $W \subseteq X_k$, $\beta \in T_{J(F)\cup\Sigma}(X_k)$. For each configuration $c \in C$, if β_c is defined, then $m_{OI}(b_F(\beta, W))(c) = b_C(\beta_c, W)$.*

Proof. The claim can be shown by tree induction on the tree β applying Claim 6.8. $\square$

Claim 6.13. *For arbitrary $k \geqslant 0$, $W \subseteq X_k$, and $\alpha \in T_{N(C)\cup\Sigma}(X_k)$, the following two statements are equivalent*:

(a) *There is a tree $\beta \in T_{J(C)\cup\Sigma}(X_k)$ such that $\bar{\beta} = \alpha$ and $b_C(\beta, W) =$* **true**.

(b) *There is a tree $s \in T_\Sigma(W)$ such that $\alpha \Rightarrow^*_{\mathcal{A}} s$.*

Proof. The claim can be shown by tree induction on the tree α. In the case that $\alpha = A(c)(\alpha_1, \ldots, \alpha_n)$, one can use the fact that α generates a tree in $T_\Sigma(W)$ if and only if there exists $V \subseteq X_n$ such that $A(c)(x_1, \ldots, x_n)$ generates a tree in $T_\Sigma(V)$, and for each $x_i \in V$, α_i generates a tree in $T_\Sigma(W)$. $\square$

By Claims 6.12 and 6.13, we get the following observation.

For arbitrary $k \geqslant 0$, $W \subseteq X_k$, $\alpha \in T_{N(F)\cup\Sigma}(X_k)$, and $c \in C$, the following two statements are equivalent:

(a) There is a tree $\beta \in T_{J(F)\cup\Sigma}(X_k)$ such that $\bar{\beta} = \alpha$ and $m_{OI}(b_F(\beta, W))(c) =$ **true**.

(b) There is a tree $s \in T_\Sigma(W)$ such that $\alpha_c \Rightarrow^*_{\mathcal{A}} s$.

We will need the following elementary property of b_C.

Claim 6.14. *Let $k \geqslant 0$, $W \subseteq X_k$. Let $\beta_0 \in T_{J(C)\cup\Sigma}(X_{k+1})$, where x_{k+1} appears exactly once in β_0, and β_0 has no subtree p such that p contains the variable x_{k+1} and the root of p is in $J(C)$. Let $\beta_1 \in T_{J(C)\cup\Sigma}(X_k)$. Then*

$$b_C(\beta_0[x_{k+1} \leftarrow \beta_1], W) = b_C(\beta_0, W \cup \{x_{k+1}\}) \text{ **and** } b_C(\beta_1, W).$$

Proof. The claim can be shown by tree induction on the tree β_0. $\square$

We now define the $OI^+(S_{OI})$ transducer $\mathcal{A} \circ \mathcal{H} = (J, e, \Delta, A_{\text{in}}^{\emptyset}, R_3)$ as follows. We put the rule

$$A^V(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ and } d_F(\delta) \textbf{ and } b_F(\delta, V) \textbf{ then } \mathcal{H}_{J(F)}(\delta) \tag{62}$$

in R_3 if Conditions $\langle 1\rangle$–$\langle 3\rangle$ hold.

$\langle 1\rangle$ the rule $A(x_1, \ldots, x_n) \to$ **if** b **then** γ is in R_1 for some $n \geqslant 0$, $A \in N_n$, $b \in BE(P)$, $\gamma \in T_{N(F)\cup\Sigma}(X_n)$.

$\langle 2\rangle$ $V \subseteq X_n$.

$\langle 3\rangle$ $\delta \in T_{J(F)\cup\Sigma}(X_n)$ and $\bar{\delta} = \gamma$.

By Definitions 6.6 and 6.9, $\mathcal{A} \circ \mathcal{H}$ is an $OI^+(S_{OI})$ transducer or it becomes one after changing the $b_F(\delta, V)$ tests by logically equivalent ones, as shown in Claim 6.10 (and, of course, such a change does not alter the induced transformation). Our aim is to show that $\tau(\mathcal{A} \circ \mathcal{H}) = \tau(\mathcal{A}) \circ \tau(\mathcal{H})$. To this end, we need the following concept.

Definition 6.15. Let $k \geqslant 0$, $\alpha \in T_{N(C)\cup\Sigma}(X_k)$, and $W \subseteq X_k$ be arbitrary. We define tree $\phi(\alpha, W) \in T_{J(C)\cup\Sigma}(X_k)$ by tree induction on α.

- If $\alpha \in X_k$, then $\phi(\alpha, W) = \alpha$,
- if $\alpha = \sigma(\alpha_1, \ldots, \alpha_n)$ with $\sigma \in \Sigma_n$, $n \geqslant 0$, then $\phi(\alpha, W) = \sigma(\phi(\alpha_1, W), \ldots, \phi(\alpha_n, W))$, and
- if $\alpha = A(c)(\alpha_1, \ldots, \alpha_n)$ with $A(c) \in N_n(C)$, $n \geqslant 0$, then $\phi(\alpha, W) = A^V(c)(\phi(\alpha_1, W), \ldots, \phi(\alpha_n, W))$, where $V = \{x_i \mid \exists s \in T_\Sigma(W) : \alpha_i \Rightarrow^*_{\mathcal{A}} s\}$.

We now show that $\phi(\alpha, W)$ is one of the β's that satisfy (a) of Claim 6.13, provided (b) of that claim holds.

Claim 6.16. *Let* $k \geqslant 0$, $\alpha \in T_{N(C)\cup\Sigma}(X_k)$, $W \subseteq X_k$, *and* $\beta = \phi(\alpha, W)$. *Then*

(i) $\bar{\beta} = \alpha$, *and*

(ii) *for every* $s \in T_\Sigma(W)$, if $\alpha \Rightarrow^*_{\mathcal{A}} s$, then $b_C(\beta, W) = \textbf{true}$.

Proof. Obviously $\bar{\beta} = \alpha$. We now show that (ii) holds. Assume that $s \in T_\Sigma(W)$ and $\alpha \Rightarrow^*_{\mathcal{A}} s$. We proceed by tree induction on α.

Base case: Let $\alpha = x_i \in X_k$. Then $\alpha = \beta = s$. Since $s \in T_\Sigma(W)$, $\beta \in W$. Hence $b_C(\beta, W) = \textbf{true}$.

Induction step: First, let $\alpha = \sigma(\alpha_1, \ldots, \alpha_n)$, where $n \geqslant 0$, $\sigma \in \Sigma_n$, $\alpha_1, \ldots, \alpha_n \in T_{N(C)\cup\Sigma}(X_k)$. Then $\beta = \sigma(\beta_1, \ldots, \beta_n)$, where $\beta_i = \phi(\alpha_i, W)$ for $1 \leqslant i \leqslant n$. Derivation $\alpha \Rightarrow^*_{\mathcal{A}} s$ implies that for every $1 \leqslant i \leqslant n$, there is a tree $s_i \in T_\Sigma(W)$ such that $\alpha_i \Rightarrow^*_{\mathcal{A}} s_i$. By the induction hypothesis $b_C(\beta_i, W) = \textbf{true}$ for $1 \leqslant i \leqslant n$. Hence $b_C(\beta, W) = \textbf{true}$.

Second, let $\alpha = A(c)(\alpha_1, \ldots, \alpha_n)$ for some $n \geqslant 0$, $A(c) \in N(C)_n$, and $\alpha_1, \ldots, \alpha_n \in T_{N(C)\cup\Sigma}(X_k)$. Then $\beta = A^V(c)(\beta_1, \ldots, \beta_n)$, where $V = \{x_i \mid \exists p \in T_\Sigma(W) : \alpha_i \Rightarrow^*_{\mathcal{A}} p\}$ and $\beta_i = \phi(\alpha_i, W)$ for $1 \leqslant i \leqslant n$. The definition of V and the derivation $\alpha \Rightarrow^*_{\mathcal{A}} s$ implies that there is a tree $s_0 \in T_\Sigma(V)$ such that $A(c)(x_1, \ldots, x_n) \Rightarrow^*_{\mathcal{A}} s_0$. By the induction hypothesis, for each $x_i \in V$, $b_C(\beta_i, W) = \textbf{true}$. Hence $b_C(\beta, W) = \textbf{true}$. □

Claim 6.17. *Let* $k \geqslant 0$, $\alpha \in T_{N(C)\cup\Sigma}(X_k)$, $W \subseteq X_k$, *and* $\beta = \phi(\alpha, W)$. *For every* $s \in T_\Sigma(W)$, *if* $\alpha \Rightarrow^*_{\mathcal{A}} s$, *then* $\mathcal{H}_{J(C)}(\beta) \Rightarrow^*_{\mathcal{A}\circ\mathcal{H}} \mathcal{H}(s)$.

Proof. Assume that $s \in T_\Sigma(W)$ and $\alpha \Rightarrow^l_{\mathcal{A}} s$ for some $l \geqslant 0$. We proceed by induction on l.

Base case: Let $l = 0$. Then $\alpha = s$. Thus $\bar{\beta} = \alpha = s$. Since $s \in T_\Sigma(W)$, $\beta = s$ and $\mathcal{H}_{J(C)}(\beta) = \mathcal{H}_{J(C)}(s) = \mathcal{H}(s)$.

Induction step: Let $l > 0$, and let the rule

$$A(x_1, \ldots, x_n) \to \textbf{ if } b \textbf{ then } \gamma \tag{63}$$

in R_1 be applied in the first step of the derivation. Then $\alpha = \alpha_0[x_{k+1} \leftarrow A(c)(\alpha_1, \ldots, \alpha_n)]$, where

- $\alpha_0 \in T_{N(C)\cup\Sigma}(X_{k+1})$,
- the variable x_{k+1} appears exactly once in α_0,
- α_0 has no subtree p such that p contains the variable x_{k+1} and the root of p is in $N(C)$, and
- $n \geqslant 0$, $A(c) \in N_n(C)$, $\alpha_1, \ldots, \alpha_n \in T_{N(C)\cup\Sigma}(X_k)$.

Moreover,

$$m(b)(c) = \textbf{true} \text{ and } \gamma_c \text{ is defined.} \tag{64}$$

Let $\alpha = \alpha_0[x_{k+1} \leftarrow A(c)(\alpha_1, \ldots, \alpha_n)] \Rightarrow_{\mathcal{A}}$
$\alpha_0[x_{k+1} \leftarrow \gamma_c[\alpha_1, \ldots, \alpha_n]] \Rightarrow^{l_1}_{\mathcal{A}}$
$\alpha_0[x_{k+1} \leftarrow s_1[\alpha_{j_1}, \ldots, \alpha_{j_m}]] \Rightarrow^{l_2}_{\mathcal{A}}$
$\alpha_0[x_{k+1} \leftarrow s_1[s'_1, \ldots, s'_m]] \Rightarrow^{l_3}_{\mathcal{A}} s_0[x_{k+1} \leftarrow s_1[s'_1, \ldots, s'_m]]$.
Here

$$s_0[x_{k+1} \leftarrow s_1[s'_1, \ldots, s'_m]] = s. \tag{65}$$

Furthermore, Conditions (A)–(E) hold.
(A) $l_1 + l_2 + l_3 = l - 1$.
(B) $s_1 \in \bar{T}_\Sigma(X_m)$, $m \geqslant 0$, $s'_1, \ldots, s'_m \in T_\Sigma(W)$, and $s_0 \in T_\Sigma(W \cup \{x_{k+1}\})$.
(C) $\gamma_c \Rightarrow^{l_1}_{\mathcal{A}} s_1[x_{j_1}, \ldots, x_{j_m}]$, $j_1, \ldots, j_m \in \{1, \ldots, n\}$.
(D) $\alpha_{j_i} \Rightarrow^{l_{i2}}_{\mathcal{A}} s'_i$ for $1 \leqslant i \leqslant m$ with $\sum_{i=1}^m l_{i2} = l_2$.
(E) $\alpha_0 \Rightarrow^{l_3}_{\mathcal{A}} s_0$.
By (65),

$$\mathcal{H}(s) = \mathcal{H}(s_0)[x_{k+1} \leftarrow \mathcal{H}(s_1)[\mathcal{H}(s'_1), \ldots, \mathcal{H}(s'_m)]], \tag{66}$$

where $\mathcal{H}(s_0) \in T_\Delta(X_{k+1})$, $\mathcal{H}(s_1) \in T_\Delta(X_m)$, and $\mathcal{H}(s'_i) \in T_\Delta(X_k)$ for $1 \leqslant i \leqslant m$. By the definition of β (Definition 6.15),

$$\beta = \beta_0[x_{k+1} \leftarrow A^V(c)(\beta_1, \ldots, \beta_n)] \in T_{J(C)\cup\Sigma}(X_k), \tag{67}$$

where $V = \{x_i \mid \exists p \in T_\Sigma(W) : \alpha_i \Rightarrow^*_{\mathcal{A}} p\}$, $\phi(\alpha_0, W) = \beta_0 \in T_{J(C)\cup\Sigma}(X_{k+1})$, and $\phi(\alpha_i, W) = \beta_i \in T_{J(C)\cup\Sigma}(X_k)$ for $1 \leqslant i \leqslant n$. Let $\eta = \phi(\gamma_c, V)$. By (D) and the definition of V, $\{ j_1, \ldots, j_m \} \subset V$. Hence by (C) and Claim 6.16, $b_C(\eta, V) = \mathbf{true}$. By (A), (C), and the induction hypothesis, $\mathcal{H}_{J(C)}(\eta) \Rightarrow^*_{\mathcal{A}\circ\mathcal{H}} \mathcal{H}(s_1[x_{j_1}, \ldots, x_{j_m}])$. Obviously, there exists $\delta \in T_{J(F)\cup\Sigma}(X_n)$ such that $\bar{\delta} = \gamma$ and $\delta_c = \eta$. Hence

$$b_C(\delta_c, V) = \mathbf{true} \tag{68}$$

and

$$\mathcal{H}_{J(C)}(\delta_c) \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \mathcal{H}(s_1[x_{j_1}, \ldots, x_{j_m}]). \tag{69}$$

Since the rule (63) is in R_1, the rule

$$A^V(x_1, \ldots, x_n) \to \ \mathbf{if}\ b\ \mathbf{and}\ d_F(\delta)\ \mathbf{and}\ b_F(\delta, V)\ \mathbf{then}\ \mathcal{H}_{J(F)}(\delta)$$

is in R_3. By Claim 6.12 and Condition (68), $m_{OI}(b_F(\delta, V))(c) = b_C(\delta_c, V) = \mathbf{true}$. By (64), $m(b)(c) = \mathbf{true}$ and, by Claim 6.7, $m_{OI}(d_F(\delta))(c) = \mathbf{true}$. Hence

$$A^V(c)(x_1, \ldots, x_n) \underset{\mathcal{A}\circ\mathcal{H}}{\Rightarrow} \mathcal{H}_{J(C)}(\delta_c).$$

By (69),

$$A^V(c)(x_1, \ldots, x_n) \underset{\mathcal{A}\circ\mathcal{H}}{\Rightarrow} \mathcal{H}_{J(C)}(\delta_c) \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \mathcal{H}(s_1[x_{j_1}, \ldots, x_{j_m}]). \tag{70}$$

By (D) and the induction hypothesis,

$$\mathcal{H}_{J(C)}(\beta_{j_i}) \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \mathcal{H}(s'_i) \quad \text{for } 1 \leqslant i \leqslant m. \tag{71}$$

It is easy to see that $\beta_0 = \phi(\alpha_0, W \cup \{ x_{k+1} \})$. By (B), $s_0 \in T_\Sigma(W \cup \{ x_{k+1} \})$. By (E) and the induction hypothesis, for tree $\beta_0 \in T_{J(C)\cup\Sigma}(X_{k+1})$ we have

$$\mathcal{H}_{J(C)}(\beta_0) \underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \mathcal{H}(s_0). \tag{72}$$

Hence

$$\begin{aligned}
&\mathcal{H}_{J(C)}(\beta) = \mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow A^V(c)(\mathcal{H}_{J(C)}(\beta_1), \ldots, \mathcal{H}_{J(C)}(\beta_m))] \text{ by (67)}\\
&\underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow \mathcal{H}(s_1)[\mathcal{H}_{J(C)}(\beta_{j_1}), \ldots, \mathcal{H}_{J(C)}(\beta_{j_m})]] \text{ by (70)}\\
&\underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow \mathcal{H}(s_1)[\mathcal{H}(s'_1), \ldots, \mathcal{H}(s'_m)]] \text{ by (71)}\\
&\underset{\mathcal{A}\circ\mathcal{H}}{\overset{*}{\Rightarrow}} \mathcal{H}(s_0)[x_{k+1} \leftarrow \mathcal{H}(s_1)[\mathcal{H}(s'_1), \ldots, \mathcal{H}(s'_m)]] \text{ by (72)}\\
&= \mathcal{H}(s) \text{ by (66)}. \qquad \square
\end{aligned}$$

Claim 6.18. *For arbitrary $k \geqslant 0$, $W \subseteq X_k$, $\alpha \in T_{N(C)\cup\Sigma}(X_k)$, and $t \in T_\Delta(X_k)$, the following two statements are equivalent.*

(a) *There is a tree $\beta \in T_{J(C)\cup\Sigma}(X_k)$ such that $\bar{\beta} = \alpha$, $b_C(\beta, W) = \mathbf{true}$, and $\mathcal{H}_{J(C)}(\beta) \Rightarrow^*_{\mathcal{A}\circ\mathcal{H}} t$.*

(b) *There is a tree* $s \in T_\Sigma(W)$ *such that* $\alpha \Rightarrow^*_{\mathcal{A}} s$ *and* $\mathcal{H}(s) = t$.

Proof. Let $k \geqslant 0$, $W \subseteq X_k$, $\alpha \in T_{N(C)\cup\Sigma}(X_k)$, and $t \in T_\Delta(X_k)$ be arbitrary. By Claims 6.16 and 6.17, Condition (b) implies Condition (a). We now show that Condition (a) implies Condition (b). Let us assume that (a) holds and that

$$\mathcal{H}_{J(C)}(\beta) \Rightarrow^l_{\mathcal{A}\circ\mathcal{H}} t \tag{73}$$

for some $l \geqslant 0$. We show (b) by induction on l.

Base case: Let $l = 0$. Then

$$\mathcal{H}_{J(C)}(\beta) = t. \tag{74}$$

By Claim 6.13, $b_C(\beta, W) = \textbf{true}$ implies that

$$\text{there is a tree } s \in T_\Sigma(W) \text{ such that } \bar{\beta} \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} s. \tag{75}$$

Using (74) and (75), we now show by tree induction on β that $\mathcal{H}(s) = t$.

Assume that $\beta = x_i$ with $1 \leqslant i \leqslant k$. Then $\bar{\beta} = x_i = s$. Hence $\mathcal{H}(s) = \mathcal{H}(\beta) = t$.

Assume that $\beta = \sigma(\beta_1, \ldots, \beta_n)$ for some $n \geqslant 0$, $\sigma \in J(C)_n \cup \Sigma_n$, $\beta_1, \ldots, \beta_n \in T_{J(C)\cup\Sigma}(X_k)$. As $\mathcal{H}_{J(C)}(\beta) = t \in T_\Delta(X_k)$, $\sigma = root(\beta) \notin J(C)$. Hence $\sigma = root(\beta) \in \Sigma$. Thus

$$\beta = \sigma(\beta_1, \ldots, \beta_n) \text{ for some } n \geqslant 0,\ \sigma \in \Sigma_n,\ \beta_1, \ldots, \beta_n \in T_{J(C)\cup\Sigma}(X_k). \tag{76}$$

By (74), $t = t_0[t_1, \ldots, t_n]$, where $t_0 = \mathcal{H}(\sigma(x_1, \ldots, x_n))$, and

$$t_i = \mathcal{H}_{J(C)}(\beta_i) \quad \text{for } 1 \leqslant i \leqslant n. \tag{77}$$

By (75) and (76), $s = \sigma(s_1, \ldots, s_n)$ for some trees $s_1, \ldots, s_n \in T_\Sigma(W)$ such that

$$\bar{\beta}_i \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} s_i \quad \text{for } 1 \leqslant i \leqslant n. \tag{78}$$

Let $1 \leqslant i \leqslant n$, and assume that x_i appears in the tree t_0. Then $t_i \in T_\Delta(X_k)$. Hence by (77), (78), and the induction hypothesis, $\mathcal{H}(s_i) = t_i$. Thus

$$\mathcal{H}(s) = t_0[\mathcal{H}(s_1), \ldots, \mathcal{H}(s_n)] = t_0[t_1, \ldots, t_n] = t.$$

Induction step: Let $l > 0$. In the first step of derivation (73) the rule (62) is applied to $\mathcal{H}_{J(C)}(\beta)$. Hence Conditions ⟨1⟩–⟨3⟩ hold, and $\mathcal{H}_{J(C)}(\beta)$ is of the following form:

$$\mathcal{H}_{J(C)}(\beta) = \eta_0[x_{k+1} \leftarrow A^V(c)(\eta_1, \ldots, \eta_n)], \tag{79}$$

where

- $\mathcal{H}_{J(C)}(\beta), \eta_1, \ldots, \eta_n \in T_{J(C)\cup\Delta}(X_k)$, $n \geqslant 0$,
- $\eta_0 \in T_{J(C)\cup\Delta}(X_{k+1})$, the variable x_{k+1} appears exactly once in η_0,
- η_0 has no subtree p such that p contains the variable x_{k+1} and the root of p is in $J(C)$, and
- $c \in C$.

Moreover, by Claim 6.7,

⟨4⟩ $m(b)(c) = \mathbf{true}$, $m_{OI}(b_F(\delta, V))(c) = \mathbf{true}$, and δ_c is defined.
Furthermore, derivation (73) looks as follows:

$\mathcal{H}_{J(C)}(\beta) = \eta_0[x_{k+1} \leftarrow A^V(c)(\eta_1, \dots, \eta_n)] \Rightarrow_{\mathcal{A}\circ\mathcal{H}}$
$\eta_0[x_{k+1} \leftarrow (\mathcal{H}_{J(F)}(\delta))_c[\eta_1, \dots, \eta_n]] \Rightarrow^{l-1}_{\mathcal{A}\circ\mathcal{H}} t$.

Since $\mathcal{H}$ is linear, there are trees $\beta_0 \in T_{J(C)\cup\Sigma}(X_{k+1})$, $\beta_1, \dots, \beta_n \in T_{J(C)\cup\Sigma}(X_k)$ such that Conditions ⟨5⟩–⟨8⟩ hold.

⟨5⟩ $\beta = \beta_0[x_{k+1} \leftarrow A^V(c)(\beta_1, \dots, \beta_n)]$,
⟨6⟩ $\mathcal{H}_{J(C)}(\beta_i) = \eta_i$ for $0 \leqslant i \leqslant n$,
⟨7⟩ the variable x_{k+1} appears exactly once in β_0.
⟨8⟩ β_0 has no subtree p such that p contains the variable x_{k+1} and the root of p is in $J(C)$.

By ⟨6⟩ and (79), we have

$$\mathcal{H}_{J(C)}(\beta) = \mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow A^V(c)(\mathcal{H}_{J(C)}(\beta_1), \dots, \mathcal{H}_{J(C)}(\beta_n))]. \tag{80}$$

The first step of (73) is of the form

$\mathcal{H}_{J(C)}(\beta) = \mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow A^V(c)(\mathcal{H}_{J(C)}(\beta_1), \dots, \mathcal{H}_{J(C)}(\beta_n))]$
$\Rightarrow_{\mathcal{A}\circ\mathcal{H}} \mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow (\mathcal{H}_{J(F)}(\delta))_c[\mathcal{H}_{J(C)}(\beta_1), \dots, \mathcal{H}_{J(C)}(\beta_n)]]$

Derivation

$\mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow (\mathcal{H}_{J(F)}(\delta))_c[\mathcal{H}_{J(C)}(\beta_1), \dots, \mathcal{H}_{J(C)}(\beta_n)]] \Rightarrow^{l-1}_{\mathcal{A}\circ\mathcal{H}} t$

can be split into three parts:

⟨9⟩ $\mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow (\mathcal{H}_{J(F)}(\delta))_c[\mathcal{H}_{J(C)}(\beta_1), \dots, \mathcal{H}_{J(C)}(\beta_n)]]$
$\Rightarrow^{l_1}_{\mathcal{A}\circ\mathcal{H}} \mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow t_1[\mathcal{H}_{J(C)}(\beta_1), \dots, \mathcal{H}_{J(C)}(\beta_n)]]$
$\Rightarrow^{l_2}_{\mathcal{A}\circ\mathcal{H}} \mathcal{H}_{J(C)}(\beta_0)[x_{k+1} \leftarrow t_2]$
$\Rightarrow^{l_3}_{\mathcal{A}\circ\mathcal{H}} t_0[x_{k+1} \leftarrow t_2] = t$,

where $l_1 + l_2 + l_3 = l - 1$, $t_1 \in T_\Delta(X_n)$, $t_2 \in T_\Delta(X_k)$, $t_0 \in T_\Delta(X_{k+1})$.

Here

⟨10⟩ $(\mathcal{H}_{J(F)}(\delta))_c \Rightarrow^{l_1}_{\mathcal{A}\circ\mathcal{H}} t_1$.
⟨11⟩ $t_1[\mathcal{H}_{J(C)}(\beta_1), \dots, \mathcal{H}_{J(C)}(\beta_n)] \Rightarrow^{l_2}_{\mathcal{A}\circ\mathcal{H}} t_2$.
⟨12⟩ $\mathcal{H}_{J(C)}(\beta_0) \Rightarrow^{l_3}_{\mathcal{A}\circ\mathcal{H}} t_0$.

By (4), δ_c is defined. By Claim 6.12 and ⟨4⟩, $b_C(\delta_c, V) = \mathbf{true}$. Furthermore, $(\mathcal{H}_{J(F)}(\delta))_c = \mathcal{H}_{J(C)}(\delta_c)$ and $\bar{\delta}_c = \gamma_c$. Hence by ⟨10⟩ and by the induction hypothesis, there is a tree $s_1 \in T_\Sigma(V)$ such that

$$\gamma_c \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} s_1 \tag{81}$$

and

$$\mathcal{H}(s_1) = t_1. \tag{82}$$

Recall that Condition (a) states that $b_C(\beta, W) = \mathbf{true}$. By ⟨5⟩ and Claim 6.14, we have

$$b_C(\beta_0, W \cup \{x_{k+1}\}) = \mathbf{true} \tag{83}$$

and

$$b_C(A^V(c)(\beta_1, \ldots, \beta_n), W) = \mathbf{true}. \tag{84}$$

By ⟨12⟩, (83), and the induction hypothesis, there is a tree $s_0 \in T_\Sigma(W \cup \{x_{k+1}\})$ such that

$$\bar{\beta}_0 \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} s_0, \text{ and } \mathcal{H}(s_0) = t_0. \tag{85}$$

By (84) and Definition 6.11, for each $x_i \in V$, $b_C(\beta_i, W) = \mathbf{true}$. Recall that $s_1 \in T_\Sigma(V)$. Using Definition 6.11 one can show by tree induction on s_1 that $b_C(s_1[\beta_1, \ldots, \beta_n], W) = \mathbf{true}$. Hence by (82), ⟨11⟩, and the induction hypothesis there is a tree $s_2 \in T_\Sigma(W)$ such that

$$s_1[\bar{\beta}_1, \ldots, \bar{\beta}_n] \overset{*}{\underset{\mathcal{A}}{\Rightarrow}} s_2 \text{ and } \mathcal{H}(s_2) = t_2. \tag{86}$$

Let

$$s = s_0[x_{k+1} \leftarrow s_2]. \tag{87}$$

Recall that $\bar{\beta} = \alpha$. By ⟨5⟩, $\alpha = \bar{\beta}_0[x_{k+1} \leftarrow A(c)(\bar{\beta}_1, \ldots, \bar{\beta}_n)]$. By ⟨1⟩ the rule $A(x_1, \ldots, x_n) \to$ **if** b **then** γ is in R_1. By ⟨4⟩,
$\alpha = \bar{\beta}_0[x_{k+1} \leftarrow A(c)(\bar{\beta}_1, \ldots, \bar{\beta}_n)]$
$\Rightarrow_{\mathcal{A}} \bar{\beta}_0[x_{k+1} \leftarrow \gamma_c[\bar{\beta}_1, \ldots, \bar{\beta}_n]]$
$\Rightarrow^*_{\mathcal{A}} \bar{\beta}_0[x_{k+1} \leftarrow s_1[\bar{\beta}_1, \ldots, \bar{\beta}_n]]$ by (81)
$\Rightarrow^*_{\mathcal{A}} \bar{\beta}_0[x_{k+1} \leftarrow s_2]$ by (86) (By ⟨7⟩ and ⟨8⟩ these derivations are *OI*.)
$\Rightarrow^*_{\mathcal{A}} s_0[x_{k+1} \leftarrow s_2]$ by (85)
$= s$ by (87).

By (85)–(87), and ⟨9⟩, $\mathcal{H}(s) = \mathcal{H}(s_0)[x_{k+1} \leftarrow \mathcal{H}(s_2)] = t_0[x_{k+1} \leftarrow t_2] = t$. In this way we have shown that Condition (a) implies Condition (b). □

We now show that

$$\tau(\mathcal{A} \circ \mathcal{H}) = \tau(\mathcal{A}) \circ \tau(\mathcal{H}). \tag{88}$$

Consider Claim 6.18 with $k = 0$, $W = \emptyset$, $\alpha = A_{\text{in}}(e(u))$ for some $u \in I$. Condition (a) is true if and only if $(u, t) \in \tau(\mathcal{A} \circ \mathcal{H})$ and $u \in dom(\tau(\mathcal{A}))$. Condition (b) is true if and only if $(u, t) \in \tau(\mathcal{A}) \circ \tau(\mathcal{H})$. Thus, it remains to show that if $(u, t) \in \tau(\mathcal{A} \circ \mathcal{H})$, then $u \in dom(\tau(\mathcal{A}))$, or equivalently, if $u \notin dom(\tau(\mathcal{A}))$, then $(u, t) \notin \tau(\mathcal{A} \circ \mathcal{H})$.

Assume that $u \notin dom(\tau(\mathcal{A}))$. Any rule of $\mathcal{A} \circ \mathcal{H}$ which can be applied to $A^{\emptyset}_{\text{in}}(e(u))$ is of the form

$$A^{\emptyset}_{\text{in}} \to \textbf{if } b \textbf{ and } d_F(\delta) \textbf{ and } b_F(\delta, \emptyset) \textbf{ then } \mathcal{H}_{J(F)}(\delta), \tag{89}$$

where the rule $A_{\text{in}} \to$ **if** b **then** $\bar{\delta}$ is in R_1. Since $u \notin dom(\tau(\mathcal{A}))$, $m(b)(e(u)) = \mathbf{false}$ or $\bar{\delta}_{e(u)}$ is not defined or $\bar{\delta}_{e(u)}$ does not generate a terminal tree. In the second case, by Claim 6.7, $d_F(\delta)(e(u)) = \mathbf{false}$. In the third case, by Claim 6.13, $b_C(\delta_{e(u)}, \emptyset) = \mathbf{false}$, and hence, by Claim 6.12, $m_{OI}(b_F(\delta, \emptyset))(e(u)) = \mathbf{false}$. Hence, in all three cases, $\mathcal{A} \circ \mathcal{H}$ cannot apply rule (89) to the configuration $A_{\text{in}}^{\emptyset}(e(u))$. Hence no rule of $\mathcal{A} \circ \mathcal{H}$ can be applied to the configuration $A_{\text{in}}^{\emptyset}(e(u))$. Thus $(u, t) \notin \tau(\mathcal{A} \circ \mathcal{H})$.

The theorem simply follows from Eq. (88). □

Theorem 6.19. *For every storage type* S, $OI^+(S_{OI}) \subseteq OI(S)$.

Proof. Let $S = (C, P, F, m, I, E)$ be an arbitrary storage type. By Corollary 3.8, it suffices to prove that $OI^+(S_{CF}) \subseteq OI(S)$. Let $\mathcal{A} = (N, e, \Delta, A_{\text{in}}, R)$ be an $OI^+(S_{CF})$ transducer. By Theorem 3.7, we may assume that $\mathcal{A}$ is an $OI^{1+}(S_{CF})$ transducer. Let $\mathcal{L}_i$, $1 \leqslant i \leqslant m$, $m \geqslant 0$, be all $CF(S)$ transducers appearing as look-ahead test in the rules of $\mathcal{A}$. We define $OI(S)$ transducer $\mathcal{B}$ such that $\tau(\mathcal{B}) = \tau(\mathcal{A})$. We define $\mathcal{B} = (N', e, \Delta, A_{\text{in}}, R')$ as follows:

- We put all elements of N in N'. Furthermore, we put each nonterminal A of $\mathcal{L}_i$, in N'_1 for $1 \leqslant i \leqslant m$.
- Let

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ and } \langle \mathcal{L}_i \rangle \textbf{ then } \gamma \tag{90}$$

be a rule in R. Let $\mathcal{L}_i = (N^i, e, \Delta^i, D_{\text{in}}^i, R^i)$. We may assume that the terminal alphabet Δ^i of $\mathcal{L}_i$ is empty. We may also assume that there is exactly one rule of $\mathcal{L}_i$ with D_{in}^i appearing in the left-hand side, cf. Lemma 3.10. Moreover, we may assume that this rule has the form $D_{\text{in}}^i \to \textbf{if true then } d_1 \cdots d_l$ with $l \geqslant 0$, $d_k \in N^i(F)$ for $1 \leqslant k \leqslant l$. Then we put the rule

$$A(x_1, \ldots, x_n) \to \textbf{if } b \textbf{ then } d_1(\cdots d_l(\gamma) \cdots) \tag{91}$$

in R'.

For each $1 \leqslant i \leqslant m$, for each rule $B \to \textbf{if } b \textbf{ then } d_1 \cdots d_j$ ($j \geqslant 0$, $d_1, \ldots, d_j \in N^i(F)$) of $\mathcal{L}_i$, we put the rule

$$B(x_1) \to \textbf{if } b \textbf{ then } d_1(\cdots d_j(x_1) \cdots) \tag{92}$$

in R'.

$\mathcal{B}$ mimics $\mathcal{A}$ in the following way. Let $\mathcal{A}$ apply rule (90). First $\mathcal{B}$ applies rule (91). Second $\mathcal{B}$ checks the look-ahead test $\langle \mathcal{L}_i \rangle$ by rewriting the subtree $d_1(\cdots d_l(\alpha))$ to α applying rules (92). A formal proof is left to the reader.

We note that the same construction also works in the *IO* case but then $\mathcal{B}$ checks the look-ahead test much later. In this way we get an alternative proof for Theorem 5.4. □

From Theorems 6.19 and 6.3 (and the obvious facts that $OI(S) \subseteq OI^+(S_{OI})$ and $OI(S) \subseteq OI(S) \circ LH$) we immediately obtain the following result.

Corollary 6.20. *For every storage type* S, $OI^+(S_{OI}) = OI(S) = OI(S) \circ LH$.

7. Conclusion

We generalized Engelfriet's decomposition result $T^R = T \circ LH$ by showing that for each storage type S, $RT^+(S_{RT}) = RT(S) \circ LH$.

We showed that for every storage type S, $IO(S)$ is closed under positive look-ahead, and is closed under composition with tree homomorphisms. That is, for every storage type S, $IO^+(S_{IO}) = IO(S) = IO(S) \circ LH = IO(S) \circ H$. We also showed that for every storage type S, $OI(S)$ is closed under positive look-ahead, and is closed under composition with linear tree homomorphisms. That is, for every storage type S, $OI^+(S_{OI}) = OI(S) = OI(S) \circ LH$.

Consider the proof of Theorem 6.3. In the light of Definition 6.15 and Claim 6.16, it is intuitively clear that it is possible to give an alternative definition of $\mathcal{A} \circ \mathcal{H}$ such that determinism is preserved: if $\mathcal{A}$ is a $DOI(S)$ transducer, then $\mathcal{A} \circ \mathcal{H}$ is a $DOI^+(S_{OI})$ transducer. Hence we conjecture that the deterministic version of Theorem 6.3 holds as well.

Conjecture 7.1. *For every storage type S, $DOI^+(S_{OI}) \supseteq DOI(S) \circ LH$.*

We raise the following problem. For a given storage type S, what is the inclusion diagram of the transformation classes $K^+(S_M)$ and $K(S_M)$ for $K \in \{RT, IO, OI, DRT, DIO, DOI\}$ and $M \in MOD \cup DMOD$?

References

[1] R. Bloem, J. Engelfriet, A comparison of tree transductions defined by monadic second order logic and by attribute grammars, J. Comput. System Sci. 61 (2000) 1–50.

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

[5] J. Engelfriet, Context-free grammars with storage, Techn. Rep. Nr 86-11, Department of Computer Science, University of Leiden, July 1986.

[6] J. Engelfriet, E.M. Schmidt, IO and OI, J. Comput. System Sci. 15 (1977) 328–353; J. Comput. System Sci. 16 (1978) 67–99.

[7] J. Engelfriet, H. Vogler, Macro tree transducers, J. Comput. System Sci. 31 (1985) 71–146.

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

[9] J. Engelfriet, H. Vogler, Look-ahead on pushdowns, Inform. and Comput. 3 (1987) 245–279.

[10] M.J. Fischer, Grammars with macro-like productions, Ph.D. Thesis, Harvard University, USA, 1968.

[11] Z. Fülöp, S. Vágvölgyi, Variants of top–down tree transducers with look-ahead, Math. Systems Theory 21 (1989) 125–145.

[12] Z. Fülöp, S. Vágvölgyi, Iterated deterministic top–down look-ahead, Fundamentals of Computation Theory, FCT '89, Lecture Notes in Computer Science, Vol. 380, Springer, Berlin, 1989, pp. 175–184.

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

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

[17] W.C. Rounds, Tree-oriented proofs of some theorems on context-free and indexed languages, Second Ann. Symp. on Theory of Computing, 1970, pp. 109–116.

[19] G. Slutzki, S. Vágvölgyi, Deterministic top–down tree transducers with iterated look-ahead, Theoret. Comput. Sci. 143 (1995) 285–308.

[20] S. Vágvölgyi, Top–down tree transducers with two-way tree walking look-ahead, Theoret. Comput. Sci. 93 (1992) 43–74.