Information and Computation **154**, 34–91 (1999)
Article ID inco.1999.2807, available online at http://www.idealibrary.com on IDEAL®

Macro Tree Transducers, Attribute Grammars, and MSO Definable Tree Translations[1]

Joost Engelfriet and Sebastian Maneth

Leiden University, LIACS, PO Box 9512,
2300 RA Leiden, The Netherlands
E-mail: engelfri@liacs.nl, maneth@liacs.nl

A characterization is given of the class of tree translations definable in monadic second-order logic (MSO), in terms of macro tree transducers. The first main result is that the MSO definable tree translations are exactly those tree translations realized by macro tree transducers (MTTs) with regular look-ahead that are single use restricted. For this the single use restriction known from attribute grammars is generalized to MTTs. Since MTTs are closed under regular look-ahead, this implies that every MSO definable tree translation can be realized by an MTT. The second main result is that the class of MSO definable tree translations can also be obtained by restricting MTTs with regular look-ahead to be finite copying, i.e., to require that each input subtree is processed only a bounded number of times. The single use restriction is a rather strong, static restriction on the rules of an MTT, whereas the finite copying restriction is a more liberal, dynamic restriction on the derivations of an MTT. © 1999 Academic Press

1. INTRODUCTION

Formulas in monadic second-order logic (MSO) can be used to define functions from graphs to graphs (cf. [Cou94]), called MSO (graph) transductions. MSO transductions have nice properties, comparable to those of finite state transductions on strings. In particular, they are closed under composition and they preserve the class of context-free graph languages. In fact, there are two large classes of context-free graph languages, namely, those generated by hyperedge replacement (HR; see, e.g., [Hab92, DKH97, Eng97]) and those generated by node replacement (NR; see, e.g., [ER97]). Both of them are preserved by (two different types of) MSO transductions. Moreover, both of them can be characterized in terms of MSO transductions; they are obtained by applying MSO transductions to regular tree languages [EvO97, CE95].

If, for an MSO transduction, we restrict the input and output graphs to be (node-labeled, ordered) trees, then we obtain a function from trees to trees, i.e., a

[1] This work was supported by the EC TMR Network GETGRATS.

0890-5401/99 $30.00

tree translation. In [BE98] the class of MSO definable tree translations was investigated and it was proved that it equals the class of tree translations realized by attributed tree transducers (ATTs) with look-ahead which are *single use restricted* (sur). ATTs are a variation of attribute grammars in which all attribute values are trees (see [Fül81, FV98]), and the sur property is a well-known restriction on the rules of an attribute grammar (introduced in [Gie88, Gan83]): each attribute is used at most once. Like for attribute grammars, the class of translations realized by sur ATTs is closed under composition, which does not hold for unrestricted ATTs (see also [Küh97]). This closure property remains when look-ahead is added. The look-ahead of an ATT can be understood as a preprocessing attribute grammar, all attributes of which are finite-valued and which merely relabels each node of the input tree. Note that ATTs are not closed under look-ahead [FV95] and, in particular, that not every translation realized by a sur ATT with look-ahead can be realized by an ATT without look-ahead.

In this paper we want to characterize the class of MSO definable tree translations in terms of macro tree transducers (MTTs). MTTs are a well-known model of syntax-directed semantics that combines the features of top-down tree transducers and of context-free tree grammars [Eng80, CF82, EV85, FV98]. Each translation realized by an ATT can also be realized by an MTT, but not vice versa [Eng80, Fra82]. Even if we add look-ahead to an ATT, the corresponding translation can still be realized by an MTT. In fact, after defining an appropriate sur property for MTTs (related to, but different from the one in [Küh97, Küh98]), we prove that the class of translations realized by sur ATTs with look-ahead is precisely the class of translations realized by sur MTTs with regular look-ahead. Unlike ATTs, MTTs are closed under regular look-ahead [EV85]. Hence, every MSO definable tree translation can be realized by an MTT.

Let us discuss this result in more detail. There is a close relationship between the states of an MTT and the synthesized attributes of an ATT and between the parameters of an MTT and the inherited attributes of an ATT. Through this relationship the single use restriction for ATTs can be generalized to MTTs and it can be shown that, in the presence of look-ahead, the classes of translations realized by single use restricted MTTs and ATTs are equal. This is our first main result. Given a tree translation τ defined in MSO we can, via the result of [BE98], construct an MTT with regular look-ahead which is single use restricted and which realizes τ. Conversely, given a single use restricted MTT with regular look-ahead we can construct a corresponding MSO transducer. However, the single use restriction is a rather strong restriction on the rules of an MTT. Thus, only for relatively few MTTs we can obtain equivalent MSO transducers using the above equivalence.

Our second main result tries to compensate this inconvenience. We give a much larger class of transducers for which the translations they realize are MSO definable. This class is obtained by restricting the MTTs to be *finite copying*. The notion of finite copying was introduced in [AU71] for generalized syntax-directed translation schemes, which are closely related to top-down tree transducers. For top-down tree transducers it was investigated in [ERS80]. Intuitively, an MTT is finite copying, if each input subtree and each parameter is copied only a bounded number of times. In contrast to the single use restriction, finite copying is a dynamic restriction which is not immediate from the rules of an MTT. We prove that finite copying MTTs with regular look-ahead realize exactly the same class of translations as single use restricted MTTs with regular look-ahead. Hence, we obtain another characterization of the MSO definable tree translations. Since, in terms of the transducers, finite copying is a much weaker restriction than single use, the class of transducers for which we can now obtain an equivalent MSO transducer is much larger.

As mentioned in the beginning of this introduction, the class of context-free graph languages (HR or NR) can be obtained by applying MSO graph transductions to regular tree languages. Thus, if we apply MSO *tree* transductions to regular tree languages, we obtain the class of tree languages which can be generated by context-free graph grammars (which turns out to be the same for HR and NR). By our results this is the class of output languages of sur (or, equivalently, finite-copying) MTTs taking regular tree languages as input. A related result has recently been proved in [Dre97]. In fact, the results of [Dre97] can be used to obtain, in a more direct way, the above characterization by MTTs of the tree languages generated by context-free graph grammars, as shown in [EM].

This paper is structured as follows. Section 2 contains basic notions concerning trees, tree translations, and tree languages. In Section 3 we recall the notions of macro tree transducer and attributed tree transducer, we introduce the concept of state sequences of MTTs, and we recall from [BE98] the notion of attributed relabeling (which defines the look-ahead of an attributed tree transducer). In Section 4 we prove the equivalence of attributed relabelings and top-down relabelings (a very restricted type of top-down tree transducer) with regular look-ahead. Section 5 concerns the single use property. This property is introduced for MTTs, together with a variant, called strongly single use. After investigating the relationship between the two variants, we are ready to prove our first main result, namely that sur ATTs and sur MTTs, both with look-ahead, realize the same class of translations. In Section 6 we define the notion of finite-copying for MTTs (based on state sequences) and prove our second main result, namely that finite copying MTTs with regular look-ahead and sur MTTs with regular look-ahead realize the same class of translations. In Section 7 we present some consequences of our results and mention some open problems.

The reader is assumed to be familiar with macro tree transducers and attribute grammars. Monadic second-order logic and MSO translations are not discussed in this paper, except in Section 7.

2. PRELIMINARIES

The set $\{0, 1, ...\}$ of natural numbers is denoted by $\mathbb{N}$. The empty set is denoted by $\varnothing$. For $k \in \mathbb{N}$, $[k]$ denotes the set $\{1, ..., k\}$; thus $[0] = \varnothing$. For a set A, $\mathscr{P}(A)$ is its powerset, $|A|$ is its cardinality, and A^* is the set of all strings over A. The empty string is denoted by ε, and the length of a string w is denoted $|w|$. For strings $v, w_1, ..., w_n \in A^*$ and distinct $a_1, ..., a_n \in A$, we denote by $v[a_1 \leftarrow w_1, ..., a_n \leftarrow w_n]$ the result of (simultaneously) substituting w_i for every occurrence of a_i in v. Note that $[a_1 \leftarrow w_1, ..., a_n \leftarrow w_n]$ is a homomorphism on strings. Let P be a condition on a and w such that $\{(a, w) \mid P\}$ is a partial function; then we use, similar to set notation, $[a \leftarrow w \mid P]$ to denote the substitution $[L]$, where L is the list of all replacements $a \leftarrow w$ for which condition P holds.

For functions $f\colon A \to B$ and $g\colon B \to C$ their composition is $(f \circ g)(x) = g(f(x))$; note that the order of f and g is nonstandard. For sets of functions F and G their composition is $F \circ G = \{f \circ g \mid f \in F, g \in G\}$.

Let $\Rightarrow \subseteq A \times A$ be a binary relation on A. Its transitive reflexive closure is denoted by $\Rightarrow^*$. If, for $a \in A$, there is exactly one $b \in A$ such that (i) $a \Rightarrow^* b$ and (ii) there is no $b' \in A$ such that $b \Rightarrow b'$, then b is said to be the *normal form of a* (*with respect to* $\Rightarrow$) and is denoted by $\mathrm{nf}(\Rightarrow, a)$.

In the remainder of this section we recall some basic notions concerning trees, tree translations, and tree languages (see, e.g., [GS84, GS97]).

2.1. *Ranked Alphabets and Trees*

A set Σ, together with a mapping $\mathrm{rank}_\Sigma\colon \Sigma \to \mathbb{N}$, is called a *ranked* set. For $k \geqslant 0$, $\Sigma^{(k)}$ is the set $\{\sigma \in \Sigma \mid \mathrm{rank}_\Sigma(\sigma) = k\}$; we also write $\sigma^{(k)}$ to indicate that $\mathrm{rank}_\Sigma(\sigma) = k$. For sets Σ and A, $\langle \Sigma, A \rangle = \Sigma \times A$; if Σ is ranked, then so is $\langle \Sigma, A \rangle$, with $\mathrm{rank}_{\langle \Sigma, A \rangle}(\langle \sigma, a \rangle) = \mathrm{rank}_\Sigma(\sigma)$ for every $\langle \sigma, a \rangle \in \langle \Sigma, A \rangle$. A *ranked alphabet* is a finite ranked set.

For the rest of this paper we choose the *set of input variables* to be $X = \{x_1, x_2, ...\}$ and the *set of parameters* to be $Y = \{y_1, y_2, ...\}$. For $k \geqslant 0$, $X_k = \{x_1, ..., x_k\}$ and $Y_k = \{y_1, ..., y_k\}$.

Let Σ be a ranked set. The *set of trees over* Σ, denoted by T_Σ, is the smallest set of strings $T \subseteq (\Sigma \cup \{(,), ,\})^*$ such that if $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $t_1, ..., t_k \in T$, then $\sigma(t_1, ..., t_k) \in T$. For $\alpha \in \Sigma^{(0)}$ we denote the tree $\alpha(\,)$ also by α. For a set A, the *set of trees over* Σ *indexed by* A, denoted by $T_\Sigma(A)$, is the set $T_{\Sigma \cup A}$, where for every $a \in A$, $\mathrm{rank}_A(a) = 0$. If $A = Y$, then $T_\Sigma(Y)$ is the set of trees (over Σ) with parameters. For every tree $t \in T_\Sigma$, the *set of occurrences* (or, *nodes*) *of* t, denoted by $\mathrm{Occ}(t)$, is a subset of $\mathbb{N}^*$ which is inductively defined as follows: if $t = \sigma(t_1, ..., t_k)$ with $\sigma \in \Sigma^{(k)}$ and $k \geqslant 0$, and for all $i \in [k]$, $t_i \in T_\Sigma$, then $\mathrm{Occ}(t) = \{\varepsilon\} \cup \bigcup_{i \in [k]} \{iu \mid u \in \mathrm{Occ}(t_i)\}$. Thus, the occurrence ε represents the root of a tree. For an occurrence u the ith child of u is represented by the occurrence ui, and for convenience we let $u0$ denote u. In particular this means that 0 denotes the occurrence ε. The usual preorder of the nodes of t (which, in fact, is the lexicographical order on $\mathbb{N}^*$) is denoted $<$; thus, $\varepsilon < iu$ (for $i > 0$). If $u < v$ then $iu < iv$, and if $i < j$ then $iu < jv$. For a tree $t \in T_\Sigma$, yt denotes the *yield of* t, i.e., the string in $(\Sigma^{(0)})^*$ obtained by reading the leaves of t in preorder; if $\Sigma^{(0)}$ contains the special symbol e, then e is interpreted as the empty string ε (thus, $y(\sigma(a, \sigma(e, b))) = ab$).

2.2. *Tree Substitution*

Let Σ be a ranked set. For every tree $t \in T_\Sigma$ and every occurrence u of t, the *label of t at occurrence u* is denoted by $t[u]$; we also say that $t[u]$ occurs in t at node u.

The *subtree of* t *at occurrence* u is denoted by t/u. The *substitution* of the tree $s \in T_\Sigma$ at occurrence u in t is denoted by $t[u \leftarrow s]$; it means that the subtree t/u is replaced by s. Formally, these notions can be defined as follows: $t[\varepsilon]$ is the first symbol of t (in Σ), $t/\varepsilon = t$, $t[\varepsilon \leftarrow s] = s$, and if $t = \sigma(t_1, ..., t_k)$, $i \in [k]$, and $u \in \mathrm{Occ}(t_i)$, then $t[iu] = t_i[u]$, $t/iu = t_i/u$, and $t[iu \leftarrow s] = \sigma(t_1, ..., t_i[u \leftarrow s], ..., t_k)$. Since trees are strings, we will also use string substitution for trees, taking care that the resulting string is a tree again. Thus, for $t, s \in T_\Sigma$ and $\sigma \in \Sigma$, $t[\sigma \leftarrow s]$ denotes the substitution of (every occurrence of) σ by s in t; if $\sigma \in \Sigma^{(k)}$ with $k \geqslant 1$, then s should also be a symbol in $\Sigma^{(k)}$.

Let $\sigma_1, ..., \sigma_n$ be distinct elements of Σ, $n \geqslant 1$, and for each $i \in [n]$ let s_i be a tree in $T_\Sigma(Y_k)$, where $k = \mathrm{rank}_\Sigma(\sigma_i)$. For $t \in T_\Sigma$, the *second-order substitution of* σ_i *by* s_i *in* t, denoted by $t[\![\sigma_1 \leftarrow s_1, ..., \sigma_n \leftarrow s_n]\!]$ is inductively defined as follows (abbreviating $[\![\sigma_1 \leftarrow s_1, ..., \sigma_n \leftarrow s_n]\!]$ by $[\![\cdots]\!]$). For $t = \sigma(t_1, ..., t_k)$ with $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $t_1, ..., t_k \in T_\Sigma$: (i) if $\sigma = \sigma_i$ for an $i \in [n]$, then $t[\![\cdots]\!] = s_i[y_j \leftarrow t_j[\![\cdots]\!] \mid j \in [k]]$ and (ii) otherwise, $t[\![\cdots]\!] = \sigma(t_1[\![\cdots]\!], ..., t_k[\![\cdots]\!])$. Let P be a condition on σ and s such that $\{(\sigma, s) \mid P\}$ is a partial function; then we use $[\![\sigma \leftarrow s \mid P]\!]$ to denote the substitution $[\![L]\!]$, where L is the list of all replacements $\sigma \leftarrow s$ for which condition P holds.

Note that (just as in ordinary substitution) second-order substitution is associative, i.e., that $t[\![\sigma \leftarrow s]\!][\![\sigma \leftarrow s']\!] = t[\![\sigma \leftarrow s[\![\sigma \leftarrow s']\!]]\!]$ and if $\sigma' \neq \sigma$ then $t[\![\sigma \leftarrow s]\!][\![\sigma' \leftarrow s']\!] = t[\![\sigma' \leftarrow s', \sigma \leftarrow s[\![\sigma' \leftarrow s']\!]]\!]$, and similarly for the general case (cf. Sections 3.4 and 3.7 of [Cou83]). The notion of second-order substitution is closely related to that of a tree homomorphism; associativity of second-order substitution corresponds to the fact that tree homomorphisms are closed under composition (cf. Theorem IV.3.7 of [GS84]).

2.3. *Tree Translations and Tree Languages*

Let Σ and Δ be ranked alphabets. A subset L of T_Σ is called a *tree language*. A (total) function $\tau: T_\Sigma \to T_\Delta$ is called a *tree translation* or simply translation. For a tree language $L \subseteq T_\Sigma$, $\tau(L)$ denotes the set $\{t \in T_\Delta \mid t = \tau(s)$ for some $s \in L\}$. For a class $\mathscr{T}$ of tree translations and a class $\mathscr{L}$ of tree languages, $\mathscr{T}(\mathscr{L})$ denotes the class of tree languages $\{\tau(L) \mid \tau \in \mathscr{T}, L \in \mathscr{L}\}$.

A *finite state tree automaton* is a tuple (P, Σ, h), where P is a finite set of *states*, Σ is a ranked alphabet of *input symbols* such that Σ is disjoint with P, and h is a collection of mappings such that for every $\sigma \in \Sigma^{(k)}$, h_σ is a mapping from P^k to P. The extension $\tilde{h}$ of h to a mapping from T_Σ to P is recursively defined as $\tilde{h}(\sigma(s_1, ..., s_k)) = h_\sigma(\tilde{h}(s_1), ..., \tilde{h}(s_k))$ for every $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_\Sigma$. Throughout this paper we simply write $h(s)$ to mean $\tilde{h}(s)$, for $s \in T_\Sigma$.

A tree language L is *regular* (or, recognizable) if there is a finite state tree automaton (P, Σ, h) and a subset F of P such that $L = \{s \in T_\Sigma \mid h(s) \in F\}$. The *class of regular tree languages* is denoted by *REGT*.

For a tree language L, $yL = \{yt \mid t \in L\}$ and for a class of tree languages $\mathscr{L}$, $y\mathscr{L} = \{yL \mid L \in \mathscr{L}\}$. For a tree translation τ, $y\tau = \{(s, yt) \mid (s, t) \in \tau\}$ and for a class of tree translations $\mathscr{T}$, $y\mathscr{T} = \{y\tau \mid \tau \in \mathscr{T}\}$.

3. TREE TRANSDUCERS

In this section we recall the basic notions of macro tree transducers [Eng80, CF82, EV85] and attributed tree transducers [Fül81, EF81, CF82] (an extensive survey of these two models of syntax-directed semantics is presented in the recent monograph [FV98]). Moreover, for macro tree transducers we introduce the notion of state sequence (Definition 3.7), and for attributed tree transducers we recall from [BE98] the notion of an attributed relabeling (Definition 3.16).

3.1. *Macro Tree Transducers*

A macro tree transducer is a syntax-directed translation device in which the translation of an input tree may not only depend on its subtrees but also on its context. The subtrees are represented by *input variables*, as usual. The context information is handled by *parameters*. Recall that for $k, m \geqslant 0$, X_k denotes the set $\{x_1, ..., x_k\}$ of input variables, and Y_m denotes the set $\{y_1, ..., y_m\}$ of parameters. In this paper we will only consider *total deterministic* macro tree transducers.

Definition 3.1 (Macro tree transducer, top-down tree transducer, top-down relabeling). A *macro tree transducer* (for short, MTT) is a tuple $M = (Q, \Sigma, \Delta, q_0, R)$, where Q is a ranked alphabet of *states*, Σ and Δ are ranked alphabets of *input* and *output symbols*, respectively, $q_0 \in Q^{(0)}$ is the *initial state*, and R is a finite set of *rules*; for every $q \in Q^{(m)}$ and $\sigma \in \Sigma^{(k)}$ with $m, k \geqslant 0$ there is exactly one rule of the form

$$\langle q, \sigma(x_1, ..., x_k)\rangle(y_1, ..., y_m) \rightarrow \zeta$$

in R, where $\zeta \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$. (Recall from Section 2.1 that $\langle Q, X_k\rangle$ is the ranked set $Q \times X_k$, where every $\langle q, x_i\rangle$ has the rank of q.)

A *top-down tree transducer* is a macro tree transducer all states of which are of rank zero. If all rules of a top-down tree transducer are of the form $\langle q, \sigma(x_1, ..., x_k)\rangle \rightarrow \delta(\langle q_1, x_1\rangle, ..., \langle q_k, x_k\rangle)$ with $\sigma \in \Sigma^{(k)}$, $\delta \in \Delta^{(k)}$, and $q, q_1, ..., q_k \in Q$, then M is a *top-down relabeling* (for short, T-REL).

A rule of the form $\langle q, \sigma(x_1, ..., x_k)\rangle(y_1, ..., y_m) \rightarrow \zeta$ is called the (q, σ)-rule and its right-hand side is denoted by $\text{rhs}_M(q, \sigma)$ (the index M is dropped if it is clear from the context); it is also called a q-rule or a σ-rule.

The derivation relation of an MTT works on trees in $T_{\langle Q, T_\Sigma\rangle \cup \Delta}$; for technical reasons (cf. Lemma 3.4 below) we extend it to trees with parameters, i.e., trees in $T_{\langle Q, T_\Sigma\rangle \cup \Delta}(Y)$.

Definition 3.2 (Derivation relation, translation). Let $M = (Q, \Sigma, \Delta, q_0, R)$ be an MTT. The *derivation relation induced by M*, denoted by $\Rightarrow_M$, is the binary relation on $T_{\langle Q, T_\Sigma\rangle \cup \Delta}(Y)$ such that, for every $\xi_1, \xi_2 \in T_{\langle Q, T_\Sigma\rangle \cup \Delta}(Y)$, $\xi_1 \Rightarrow_M \xi_2$ if and only if there exist $u \in \text{Occ}(\xi_1)$, $\sigma \in \Sigma^{(k)}$, $s_1, ..., s_k \in T_\Sigma$, $q \in Q^{(m)}$, and $t_1, ..., t_m \in T_{\langle Q, T_\Sigma\rangle \cup \Delta}(Y)$ such that $\xi_1/u = \langle q, \sigma(s_1, ..., s_k)\rangle(t_1, ..., t_m)$ and $\xi_2 = \xi_1[u \leftarrow \zeta]$ with $\zeta = \text{rhs}(q, \sigma)[\langle q', x_i\rangle \leftarrow \langle q', s_i\rangle \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle][y_j \leftarrow t_j \mid j \in [m]]$.

The *translation realized by M*, denoted by τ_M, is the total function

$$\{(s, t) \in T_\Sigma \times T_\Delta \mid \langle q_0, s\rangle \Rightarrow^*_M t\}.$$

The class of all translations which can be realized by macro tree transducers is denoted by *MTT*. The classes of translations realized by top-down tree transducers and by top-down relabelings are denoted by *T* and *T-REL*, respectively.

Let us now add regular look-ahead to a macro tree transducer [EV85, Eng77].

DEFINITION 3.3 (MTT with regular look-ahead). A *macro tree transducer with regular look-ahead* (for short, MTT^{R}) is a tuple $M = (Q, P, \Sigma, \Delta, q_0, R, h)$, where (P, Σ, h) is a finite state tree automaton, called the *look-ahead automaton of M*, the components Q, Σ, Δ, and q_0 are as in Definition 3.1, and R is a finite set of rules of the following form: For every $q \in Q^{(m)}$, $\sigma \in \Sigma^{(k)}$, and $p_1, ..., p_k \in P$ with $m, k \geqslant 0$ there is exactly one rule of the form

$$\langle q, \sigma(x_1, ..., x_k)\rangle(y_1, ..., y_m) \to \zeta \quad \langle p_1, ..., p_k\rangle \tag{$*$}$$

in R, where $\zeta \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$.

A rule of the form $(*)$ is called the $(q, \sigma, \langle p_1, ..., p_k\rangle)$-rule and its right-hand side ζ is denoted by $\text{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$ (M is dropped if it is clear from the context). The derivation relation $\Rightarrow_M$ of M is defined as in Definition 3.2, with $\text{rhs}(q, \sigma)$ replaced by $\text{rhs}(q, \sigma, \langle h(s_1), ..., h(s_k)\rangle)$ and the translation τ_M realized by M is defined as in Definition 3.2. The class of all translations which can be realized by MTT^{R}s is denoted by MTT^R; it is shown in Theorem 4.21 of [EV85] that $MTT^R = MTT$. The class of all translations which can be realized by MTT^{R}s with states of rank zero only, i.e., by top-down tree transducers with regular look-ahead (for short, T^{R}s), is denoted by T^R. The class of all translations realized by top-down relabelings with regular look-ahead (for short, T^{R}-RELs) is denoted by T^R-*REL*.

In Sections 5 and 6 several subclasses of MTT^R will be defined by putting restrictions on MTT^{R}s. We fix the following convention: If X is a restriction on MTT^{R}s, then an MTT $M = (Q, \Sigma, \Delta, q_0, R)$ satisfies X, if the MTT^{R} $(Q, \{p\}, \Sigma, \Delta, q_0, \{(r\langle p, ..., p\rangle) \mid r \in R\}, h)$ satisfies X, where $h_\sigma(p, ..., p) = p$ for every $\sigma \in \Sigma$. Moreover, if MTT^R_X is the class of translations realized by MTT^{R}s which satisfy X, then we denote by MTT_X the class of translations realized by MTTs which satisfy X (in the above sense). By T^{R}_X we denote the class of translations realized by T^{R}s which satisfy X, and by T_X we denote the class of translations realized by top-down tree transducers which satisfy X (defined as above for MTTs).

The next lemma will be used in proofs by induction on the structure of the input tree. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^{R}. For every $q \in Q^{(m)}$ and $s \in T_\Sigma$ let the *q-translation of s*, denoted by $M_q(s)$, be the unique tree $t \in T_\Delta(Y_m)$ such that $\langle q, s\rangle(y_1, ..., y_m) \Rightarrow^*_M t$. Note that, for $s \in T_\Sigma$, $\tau_M(s) = M_{q_0}(s)$. The q-translations of trees in T_Σ can be characterized inductively as follows.

LEMMA 3.4 (Lemma 4.8 of [EV94]). Let $M=(Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT$^{\mathrm{R}}$. For every $q \in Q$, $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_\Sigma$:

$$M_q(\sigma(s_1, ..., s_k)) = \mathrm{rhs}(q, \sigma, \langle h(s_1), ..., h(s_k)\rangle)[\![\langle q', x_i\rangle \leftarrow M_{q'}(s_i) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!].$$

State Sequences of MTTs. The notion of state sequence was introduced in [ERS80] for top-down tree transducers. We now generalize this notion to MTT$^{\mathrm{R}}$s. To motivate the definition, we first discuss and prove a generalization of Lemma 3.4.

Let $M=(Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT$^{\mathrm{R}}$. Lemma 3.4 shows how the translation of a tree can be expressed in terms of the translations of its direct subtrees. More generally, we wish to know how the translation of a tree s depends on the translations of a subtree s/u for a given node u of s. To see this, we have to know how M translates the "context" of u in s, i.e., the tree $s[u \leftarrow x_1]$. However, M cannot process $s[u \leftarrow x_1]$ unless it knows the look-ahead state p of the subtree s/u. Thus, more precisely, we define the *context of u in s* to be the tree $s[u \leftarrow p]$, where $p = h(s/u)$, viewed as a symbol of rank 0. Now, clearly, if we extend the look-ahead automaton of M by putting $h_p() = p$, M translates the context $s[u \leftarrow p]$ into a tree which still contains state calls $\langle q', p\rangle$. Then, the translation of s is obtained from this tree by the second-order substitution $[\![\langle q', p\rangle \leftarrow M_{q'}(s/u)]\!]$. We will now prove this formally, and we start by formalizing the translation by M of the context of u in s. To do this, it is technically convenient to view the state calls $\langle q', p\rangle$ as new output symbols, just as we viewed the look-ahead state as a new input symbol.

For a ranked alphabet $\langle Q, P\rangle$, let $\langle\!\langle Q, P\rangle\!\rangle$ be a fresh copy of $\langle Q, P\rangle$; i.e., $\langle\!\langle Q, P\rangle\!\rangle = \{\langle\!\langle q, p\rangle\!\rangle \mid q \in Q, p \in P\}$, where $\langle\!\langle q, p\rangle\!\rangle$ is a new symbol of the same rank as $\langle q, p\rangle$.

The q-translation by M of the context of u in s is now defined to be $\hat{M}_q(s[u \leftarrow h(s/u)])$, where $\hat{M}$ is the following extension of M.

DEFINITION 3.5 (Extension of M). Let $M=(Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT$^{\mathrm{R}}$. The *extension of* M, denoted by $\hat{M}$, is the MTT$^{\mathrm{R}}$ $(Q, P, \hat{\Sigma}, \hat{\Delta}, q_0, \hat{R}, \hat{h})$, where $\hat{\Sigma} = \Sigma \cup \{p^{(0)} \mid p \in P\}$, $\hat{\Delta} = \Delta \cup \langle\!\langle Q, P\rangle\!\rangle$, $\hat{R} = R \cup \{\langle q, p\rangle(y_1, ..., y_m) \rightarrow \langle\!\langle q, p\rangle\!\rangle (y_1, ..., y_m) \mid \langle q, p\rangle \in \langle Q, P\rangle^{(m)}\}$, $\hat{h}_p() = p$ for $p \in P$, and $\hat{h}_\sigma(p_1, ..., p_k) = h_\sigma(p_1, ..., p_k)$ for $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $p_1, ..., p_k \in P$.

Now, $\hat{M}_q(s[u \leftarrow p])$ is a tree (with parameters) over $\Delta \cup \langle\!\langle Q, \{p\}\rangle\!\rangle$ and if we replace in $\hat{M}_q(s[u \leftarrow p])$ each $\langle\!\langle q', p\rangle\!\rangle$ by $M_{q'}(s/u)$, then we obtain $M_q(s)$. This then generalizes the inductive characterization of $M_q(s)$ in Lemma 3.4 from the application of a rule at the root of s to an arbitrary node u of s. It is stated in the next lemma, for the slightly more general case that s/u may also contain look-ahead states.

LEMMA 3.6. Let $M=(Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT$^{\mathrm{R}}$ and $\hat{M}=(Q, P, \hat{\Sigma}, \hat{\Delta}, q_0, \hat{R}, \hat{h})$ its extension. Let $q \in Q$, $s \in T_{\hat{\Sigma}}$, $u \in \mathrm{Occ}(s)$, and $p = \hat{h}(s/u)$, such that $s[u \leftarrow p]$ contains exactly one occurrence of an element of P. Then

$$\hat{M}_q(s) = \hat{M}_q(s[u \leftarrow p])[\![\langle\!\langle q', p\rangle\!\rangle \leftarrow \hat{M}_{q'}(s/u) \mid q' \in Q]\!].$$

Proof. This lemma is proved by induction on the structure of s. Let $q \in Q^{(m)}$ and $s = \sigma(s_1, ..., s_k)$ with $\sigma \in \hat{\Sigma}^{(k)}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_{\hat{\Sigma}}$. Let $[\![\cdots]\!]$ denote the substitution $[\![\langle\!\langle q', p \rangle\!\rangle \leftarrow \hat{M}_{q'}(s/u) \mid q' \in Q]\!]$. If $u = \varepsilon$, then $\hat{M}_q(s[\varepsilon \leftarrow p])[\![\cdots]\!] = \hat{M}_q(p)[\![\cdots]\!] = \langle\!\langle q, p \rangle\!\rangle(y_1, ..., y_m)[\![\cdots]\!] = \hat{M}_q(s/\varepsilon) = \hat{M}_q(s)$. Otherwise, $u = iv$ with $i \in [k]$ and $v \in \mathrm{Occ}(s_i)$. Then $\hat{M}_q(s[iv \leftarrow p])[\![\cdots]\!] = \hat{M}_q(\sigma(s_1, ..., s_{i-1}, s_i[v \leftarrow p], s_{i+1}, ..., s_k))[\![\cdots]\!]$. Let $\zeta = \mathrm{rhs}_{\hat{M}}(q, \sigma, \langle \hat{h}(s_1), ..., \hat{h}(s_{i-1}), \hat{h}(s_i[v \leftarrow p]), \hat{h}(s_{i+1}), ..., \hat{h}(s_k) \rangle)$. By Lemma 3.4 the above equals $\zeta \Theta_1 \cdots \Theta_{i-1} [\![\langle r, x_i \rangle \leftarrow \hat{M}_r(s_i[v \leftarrow p]) \mid r \in Q]\!] \Theta_{i+1} \cdots \Theta_k [\![\cdots]\!]$, where $\Theta_j = [\![\langle r, x_j \rangle \leftarrow \hat{M}_r(s_j) \mid r \in Q]\!]$ for $j \in [k]$. By associativity of second-order substitution this equals $\zeta [\![\cdots]\!] \Theta'_1 \cdots \Theta'_{i-1} [\![\langle r, x_i \rangle \leftarrow \hat{M}_r(s_i[v \leftarrow p]) [\![\cdots]\!] \mid r \in Q]\!] \Theta'_{i+1} \cdots \Theta'_k$, where $\Theta'_j = [\![\langle r, x_j \rangle \leftarrow \hat{M}_r(s_j)[\![\cdots]\!] \mid r \in Q]\!]$ for $j \in [k]$. Since $s[u \leftarrow p]$ contains exactly one occurrence of an element in P, $\hat{M}_r(s_j)$ does not contain elements of $\langle\!\langle Q, \{p\} \rangle\!\rangle$ for $j \in [k] - \{i\}$. Also $\zeta \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$ and, hence, $\zeta [\![\cdots]\!] = \zeta$ and $\Theta'_j = \Theta_j$ for $j \in [k] - \{i\}$. Since $s/u = s_i/v$, $\hat{M}_r(s_i[v \leftarrow p])[\![\cdots]\!] = \hat{M}_r(s_i[v \leftarrow p])[\![\langle\!\langle q', p \rangle\!\rangle \leftarrow \hat{M}_{q'}(s_i/v) \mid q' \in Q]\!]$ which equals $\hat{M}_r(s_i)$ by induction. Therefore $[\![\langle r, x_i \rangle \leftarrow \hat{M}_r(s_i[v \leftarrow p])[\![\cdots]\!] \mid r \in Q]\!] = \Theta_i$ and we get $\zeta \Theta_1 \cdots \Theta_k$ which is equal to $\hat{M}_q(s)$ by Lemma 3.4. ∎

Obviously, $\hat{M}_q(s) = M_q(s)$ for every $s \in T_\Sigma$; thus, in this case, the first and third hat can be removed in the displayed formula of Lemma 3.6. In particular (since $\tau_M(s) = M_{q_0}(s)$), for every tree $s \in T_\Sigma$ and every node u of s, the translation $\tau_M(s)$ of the input tree s can be expressed in terms of the q-translations $M_q(s/u)$ of the subtree s/u. The states q that are used in this expression form the *state sequence of s at u*. In other words, the state sequence of s at u is the sequence of states that occur in $\hat{M}_{q_0}(s[u \leftarrow h(s/u)])$, the q_0-translation of the context of u in s.

Definition 3.7 (State sequence). Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\mathrm{MTT}^{\mathrm{R}}$, $s \in T_\Sigma$, and $u \in \mathrm{Occ}(s)$. Let $p = h(s/u)$ and $\xi = \hat{M}_{q_0}(s[u \leftarrow p]) \in T_{\langle\!\langle Q, \{p\} \rangle\!\rangle \cup \Delta}$, and let $\{v \in \mathrm{Occ}(\xi) \mid \xi[v] \in \langle\!\langle Q, \{p\} \rangle\!\rangle\} = \{v_1, ..., v_n\}$ with $v_1 < \cdots < v_n$. The *state sequence of s at u*, denoted by $\mathrm{sts}_M(s, u)$, is the sequence of states $q_1 \cdots q_n$ such that $\xi[v_i] = \langle\!\langle q_i, p \rangle\!\rangle$ for every $i \in [n]$.

The following small example illustrates Definition 3.7.

Example 3.8. Consider the $\mathrm{MTT}^{\mathrm{R}}$ $M = (Q, P, \Sigma, \Delta, q_0, R, h)$, where $Q = \{q_0^{(0)}, q_1^{(2)}, q_2^{(0)}, q_3^{(0)}, q_4^{(1)}\}$, $P = \{p, p'\}$, $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}, \beta^{(0)}\}$, and $\Delta = \Sigma$. Moreover, let $h_\sigma(p', p') = p$ and $h_\gamma(p) = p$, and let R contain, among others, the rules:

$$\langle q_0, \gamma(x_1) \rangle \to \langle q_1, x_1 \rangle(\langle q_2, x_1 \rangle, \langle q_3, x_1 \rangle) \qquad \langle p \rangle$$

$$\langle q_1, \sigma(x_1, x_2) \rangle(y_1, y_2) \to \sigma(\langle q_1, x_2 \rangle(y_2, y_1), \langle q_4, x_1 \rangle(y_1)) \qquad \langle p', p' \rangle$$

$$\langle q_2, \sigma(x_1, x_2) \rangle \to \langle q_2, x_2 \rangle \qquad \langle p', p' \rangle$$

$$\langle q_3, \sigma(x_1, x_2) \rangle \to \langle q_3, x_2 \rangle \qquad \langle p', p' \rangle$$

Consider $s = \gamma(\sigma(s_1, s_2))$ and assume that $h(s_1) = h(s_2) = p'$. Then a derivation by M looks like

$$\begin{aligned}\langle q_0, \gamma(\sigma(s_1, s_2))\rangle &\Rightarrow_M \langle q_1, \sigma(s_1, s_2)\rangle(\langle q_2, \sigma(s_1, s_2)\rangle, \langle q_3, \sigma(s_1, s_2)\rangle)\\ &\Rightarrow_M \sigma(\langle q_1, s_2\rangle(\langle q_3, \sigma(s_1, s_2)\rangle, \langle q_2, \sigma(s_1, s_2)\rangle),\\ &\qquad \langle q_4, s_1\rangle(\langle q_2, \sigma(s_1, s_2)\rangle))\\ &\Rightarrow^*_M \sigma(\langle q_1, s_2\rangle(\langle q_3, s_2\rangle, \langle q_2, s_2\rangle), \langle q_4, s_1\rangle(\langle q_2, s_2\rangle)).\end{aligned}$$

Then $\mathrm{sts}_M(s, \varepsilon) = q_0$, $\mathrm{sts}_M(s, 1) = q_1 q_2 q_3$, $\mathrm{sts}_M(s, 11) = q_4$, and $\mathrm{sts}_M(s, 12) = q_1 q_3 q_2 q_2$ because

$\hat{M}_{q_0}(s[\varepsilon \leftarrow p]) = \hat{M}_{q_0}(p) = \langle\!\langle q_0, p\rangle\!\rangle$,
$\hat{M}_{q_0}(s[1 \leftarrow p]) = \hat{M}_{q_0}(\gamma(p)) = \langle\!\langle q_1, p\rangle\!\rangle(\langle\!\langle q_2, p\rangle\!\rangle, \langle\!\langle q_3, p\rangle\!\rangle)$,
$\hat{M}_{q_0}(s[11 \leftarrow p']) = \hat{M}_{q_0}(\gamma(\sigma(p', s_2))) = \sigma(\sigma(\beta, \alpha), \langle\!\langle q_4, p'\rangle\!\rangle(\alpha))$, and
$\hat{M}_{q_0}(s[12 \leftarrow p']) = \hat{M}_{q_0}(\gamma(\sigma(s_1, p'))) = \sigma(\langle\!\langle q_1, p'\rangle\!\rangle(\langle\!\langle q_3, p'\rangle\!\rangle, \langle\!\langle q_2, p'\rangle\!\rangle), \langle\!\langle q_2, p'\rangle\!\rangle)$,

where we assume that $M_{q_1}(s_2) = \sigma(y_1, y_2)$, $M_{q_3}(s_2) = \beta$, $M_{q_2}(s_2) = \alpha$, and $M_{q_4}(s_1) = y_1$.

It should be noted here that the notion of state sequence for MTTs, in general, is less straightforward (and maybe less intuitive) than for top-down tree transducers. For a top-down tree transducer M, a state sequence $q_1 \cdots q_n$ (at node u of s) means that the trees $M_{q_1}(s/u)$, ..., $M_{q_n}(s/u)$ will be subtrees of $\tau_M(s)$, in the same order. This is because $\langle\!\langle q_1, p\rangle\!\rangle$, ..., $\langle\!\langle q_n, p\rangle\!\rangle$ label leaves of $\hat{M}_{q_0}(s[u \leftarrow p])$. For an arbitrary MTT M, however, $\langle\!\langle q_1, p\rangle\!\rangle$, ..., $\langle\!\langle q_n, p\rangle\!\rangle$ may occur nested. Let, for instance, $\hat{M}_{q_0}(s[u \leftarrow p])$ be $\langle\!\langle q_1, p\rangle\!\rangle(\langle\!\langle q_2, p\rangle\!\rangle, \langle\!\langle q_3, p\rangle\!\rangle)$; i.e., the state sequence is $q_1 q_2 q_3$. Then $M_{q_1}(s/u)$ is a part of $\tau_M(s)$, though not a subtree, of course. However, $\tau_M(s)$ does not contain $M_{q_2}(s/u)$ at all if the parameter y_1 does not occur in $M_{q_1}(s/u)$, and it contains $M_{q_2}(s/u)$ twice if y_1 occurs twice in $M_{q_1}(s/u)$. Also, even if $M_{q_1}(s/u)$ contains y_1 and y_2 exactly once, $M_{q_2}(s/u)$ and $M_{q_3}(s/u)$ appear in reversed order in $\tau_M(s)$ if $M_{q_1}(s/u)$ contains first y_2 and then y_1. Thus, in Example 3.8 (with $u = 1$ and $s/u = \sigma(s_1, s_2)$), $M_{q_1}(s/u) = \sigma(\sigma(y_2, y_1), y_1)$, $M_{q_2}(s/u) = \alpha$, $M_{q_3}(s/u) = \beta$, and $\tau_M(s) = \sigma(\sigma(\beta, \alpha), \alpha)$. Altogether this means that the deletion, copying, and permutation of parameters influences the relationship between the state sequence and the way in which $\tau_M(s)$ contains the q-translations of s/u.

3.2. Attributed Tree Transducers

Attributed tree transducers are attribute grammars, in which all attribute values are trees and the only operation in the semantic rules is the substitution of trees for the leaves of a given tree. Moreover, an attributed tree transducer takes as input the set T_Σ of trees over a ranked alphabet Σ, instead of the set of derivation trees of a context-free grammar, as is the case for an attribute grammar.

The symbols $\pi, \pi 1, \pi 2, \ldots$ are called *node variables* and are used in the rules of an attributed tree transducer to indicate nodes of an input tree. If π denotes a node u, then πi denotes its ith child ui. We also define $\pi 0 = \pi$ (recall from Section 2.1 that $u0 = u$).

Definition 3.9 (Attributed tree transducer). An *attributed tree transducer* (for short, ATT) is a tuple $A = (\text{Syn}, \text{Inh}, \Sigma, \Delta, \text{root}, a_0, R)$, where Syn and Inh are disjoint alphabets, the elements of which are called *synthesized* and *inherited attributes*, respectively, Σ and Δ are ranked alphabets of *input* and *output symbols*, respectively, root is a symbol of rank 1 with $\text{root} \notin \Sigma$, called the *root marker*, and a_0 is a synthesized attribute, called the *initial attribute*. Let Σ_{root} denote the ranked alphabet $\Sigma \cup \{\text{root}^{(1)}\}$ and let Att denote the set $\text{Syn} \cup \text{Inh}$. Before defining R we fix two auxiliary notions.

For every $\sigma \in \Sigma^{(k)}$, the *set of inside attributes of* σ, denoted by ins_σ, is the set $\{\langle a, \pi\rangle \mid a \in \text{Syn}\} \cup \{\langle b, \pi i\rangle \mid b \in \text{Inh},\ i \in [k]\}$ and the *set of outside attributes*, denoted by outs_σ, is the set $\{\langle b, \pi\rangle \mid b \in \text{Inh}\} \cup \{\langle a, \pi i\rangle \mid a \in \text{Syn}, i \in [k]\}$. For the root marker, $\text{ins}_{\text{root}} = \{\langle a_0, \pi\rangle\} \cup \{\langle b, \pi 1\rangle \mid b \in \text{Inh}\}$ and $\text{outs}_{\text{root}} = \{\langle a, \pi 1\rangle \mid a \in \text{Syn}\}$.

$R = (R_\sigma \mid \sigma \in \Sigma_{\text{root}})$ is a collection of finite sets of *rules* such that for every $\sigma \in \Sigma_{\text{root}}$ and $\langle c, \rho\rangle \in \text{ins}_\sigma$ there is exactly one rule of the form $\langle c, \rho\rangle \to \zeta$ in R_σ, where $\zeta \in T_\Delta(\text{outs}_\sigma)$.

For $\sigma \in \Sigma^{(k)}_{\text{root}}$ and $\langle c, \rho\rangle \in \text{ins}_\sigma$, the rule $\langle c, \rho\rangle \to \zeta$ in R_σ is called the $(\langle c, \rho\rangle, \sigma)$-rule and ζ is denoted by $\text{rhs}_A(\langle c, \rho\rangle, \sigma)$ (A is dropped if it is clear from the context).

Note that, by the rules in R_σ, each inside attribute of σ is defined in terms of the outside attributes of σ (i.e., we assume Bochmann Normal Form [Boc76]). Note also that our definition of ATTs in Definition 3.2 is different from the original definition in [Fül81]. There, for every inherited attribute b, the right-hand side of the $(\langle b, \pi 1\rangle, \text{root})$-rule is restricted to trees over Δ. In the appendix of [Gie88] this difference was pointed out and the term *full attributed tree transducer* was used to refer to the transducers as in Definition 3.2.

In what follows let $A = (\text{Syn}, \text{Inh}, \Sigma, \Delta, \text{root}, a_0, R)$ be an ATT.

Definition 3.10 (Derivation relation induced by an ATT). Let s be a tree in $T_{\Sigma_{\text{root}}}$. The *derivation relation induced by A on s*, denoted by $\Rightarrow_{A,s}$, is the binary relation over $T_\Delta(\langle \text{Att}, \text{Occ}(s)\rangle)$ such that $\xi_1 \Rightarrow_{A,s} \xi_2$ for $\xi_1, \xi_2 \in T_\Delta(\langle \text{Att}, \text{Occ}(s)\rangle)$, if there is an attribute instance $\langle c, v\rangle \in \langle \text{Att}, \text{Occ}(s)\rangle$, an occurrence $u \in \text{Occ}(\xi_1)$ with $\xi_1/u = \langle c, v\rangle$, and

- either $c \in \text{Syn}$, $s[v] = \sigma$ with $\sigma \in \Sigma^{(k)}_{\text{root}}$, $k \geqslant 0$, and $\xi_2 = \xi_1[u \leftarrow \zeta]$ with $\zeta = \text{rhs}(\langle c, \pi\rangle, \sigma)[\langle d, \pi i\rangle \leftarrow \langle d, vi\rangle \mid d \in \text{Att},\ 0 \leqslant i \leqslant k]$
- or $c \in \text{Inh}$ and $v = \bar{v}j$ for some $\bar{v} \in \text{Occ}(s)$, $s[\bar{v}] = \sigma$ with $\sigma \in \Sigma^{(k)}_{\text{root}}$, $k \geqslant 1$, $j \in [k]$, and $\xi_2 = \xi_1[u \leftarrow \zeta]$ with $\zeta = \text{rhs}(\langle c, \pi j\rangle, \sigma)[\langle d, \pi i\rangle \leftarrow \langle d, \bar{v}i\rangle \mid d \in \text{Att},\ 0 \leqslant i \leqslant k]$.

In the same sense as attribute grammars, ATTs can be *circular* (see [Fül81] for the notion of circularity of ATTs). In the remainder of this paper we always mean noncircular ATTs when referring to ATTs.

If an ATT is noncircular, then the derivation relation on any tree $\text{root}(s)$, $s \in T_\Sigma$, is confluent and terminating (see, e.g., [FHVV93]). Thus, every attribute instance $\langle c, v\rangle \in \langle \text{Att}, \text{Occ}(\text{root}(s))\rangle$ has a unique normal form $\text{nf}(\Rightarrow_{A,\text{root}(s)}, \langle c, v\rangle) \in T_\Delta$;

intuitively, this is the value of the attribute c at node v. Let us now define the translation realized by an ATT.

DEFINITION 3.11 (Translation realized by A). The *translation realized by* A, denoted by τ_A, is the total function

$$\{(s,t) \in T_\Sigma \times T_\Delta \mid t = \mathrm{nf}(\Rightarrow_{A,\mathrm{root}(s)}, \langle a_0, \varepsilon\rangle)\}.$$

The class of all translations which can be realized by attributed tree transducers is denoted by *ATT*.

Consider an input tree $s \in T_\Sigma$ and an attribute instance $\langle c, v\rangle \in \langle \mathrm{Att}, \mathrm{Occ}(s)\rangle$. The normal form of $\langle c, v\rangle$ with respect to $\Rightarrow_{A,s}$ depends only on attribute instances of the form $\langle b, \varepsilon\rangle$, where b is an inherited attribute; in other words, $\mathrm{nf}(\Rightarrow_{A,s}, \langle c, v\rangle) \in T_\Delta(\langle \mathrm{Inh}, \{\varepsilon\}\rangle)$. The next lemma shows how the attributes of an input tree $\sigma(s_1, ..., s_k)$ can be expressed in those of its subtrees $s_1, ..., s_k$, allowing proofs by induction on the structure of the input tree.

LEMMA 3.12. *Let* $s = \sigma(s_1, ..., s_k)$, $k \geq 0$, $\sigma \in \Sigma^{(k)}_{\mathrm{root}}$, *and* $s_1, ..., s_k \in T_\Sigma$.

(1) *For* $i \in [k]$ *and* $\langle c, v\rangle \in \langle \mathrm{Att}, \mathrm{Occ}(s_i)\rangle$,

$$\mathrm{nf}(\Rightarrow_{A,s}, \langle c, iv\rangle) = \mathrm{nf}(\Rightarrow_{A,s_i}, \langle c, v\rangle)[\langle \beta, \varepsilon\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle \beta, i\rangle) \mid \beta \in \mathrm{Inh}].$$

(2) *Let* $a \in \mathrm{Syn}$ (*with* $a = a_0$ *if* $\sigma = \mathrm{root}$), $b \in \mathrm{Inh}$, *and* $j \in [k]$.

Then $\mathrm{nf}(\Rightarrow_{A,s}, \langle a, \varepsilon\rangle) = \mathrm{rhs}(\langle a, \pi\rangle, \sigma)\, \Theta_1 \Theta_2$ *and* $\mathrm{nf}(\Rightarrow_{A,s}, \langle b, j\rangle) = \mathrm{rhs}(\langle b, \pi j\rangle, \sigma)\, \Theta_1 \Theta_2$, *where* Θ_1 *is the substitution*

$$[\langle \alpha, \pi i\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle \alpha, i\rangle) \mid \alpha \in \mathrm{Syn}, i \in [k]]$$

and Θ_2 *is the substitution*

$$[\langle \beta, \pi\rangle \leftarrow \langle \beta, \varepsilon\rangle \mid \beta \in \mathrm{Inh}].$$

Note that (1) *is applicable to* $\mathrm{nf}(\Rightarrow_{A,s}, \langle \alpha, i\rangle)$, *with* $c = \alpha$ *and* $v = \varepsilon$.

Dependencies. Often one is interested in the set of inherited attributes on which a synthesized attribute depends, at the root of an input tree s. We call such a dependency *is-dependency*. If we know the is-dependencies for $s_1, ..., s_k$, then we can easily determine the is-dependency for the tree $s = \sigma(s_1, ..., s_k)$ using the rules in R_σ.

DEFINITION 3.13 (Is-dependency). An *is-dependency* is a subset of $\mathrm{Inh} \times \mathrm{Syn}$. Let $\sigma \in \Sigma^{(k)}_{\mathrm{root}}$ and let $d_1, ..., d_k$ be is-dependencies. The *is-dependency of* σ *with* $d_1, ..., d_k$, denoted by $\mathrm{is}_\sigma(d_1, ..., d_k)$, equals

$$\{(b, a) \in \mathrm{Inh} \times \mathrm{Syn} \mid \text{there is a path in } g \text{ from } \langle b, \pi\rangle \text{ to } \langle a, \pi\rangle\},$$

where g is the directed graph (V, E) with $V = \text{ins}_\sigma \cup \text{outs}_\sigma$ and $E = \{(\langle c, \pi i\rangle, \langle c', \pi j\rangle) \in \text{outs}_\sigma \times \text{ins}_\sigma \mid \langle c, \pi i\rangle$ occurs in $\text{rhs}(\langle c', \pi j\rangle, \sigma)\} \cup \{(\langle b, \pi i\rangle, \langle a, \pi i\rangle) \mid (b, a) \in d_i,\ i \in [k]\}$. The graph g is called the *dependency graph of σ with $d_1, ..., d_k$* and is denoted by $D_\sigma(d_1, ..., d_k)$.

Let $s = \sigma(s_1, ..., s_k)$, $\sigma \in \Sigma^{(k)}_{\text{root}}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_\Sigma$. The *is-dependency of s*, denoted by $\text{is}(s)$, is recursively defined as $\text{is}_\sigma(\text{is}(s_1), ..., \text{is}(s_k))$.

The *set of all is-dependencies of A*, denoted by $\text{IS}(A)$, is the set $\{\text{is}(s) \mid s \in T_\Sigma\}$.

Note that $\text{IS}(A)$ is finite and can be constructed effectively. In fact, it is the smallest set of is-dependencies that is closed under all is_σ, $\sigma \in \Sigma$. Note that A is noncircular if and only if all graphs $D_\sigma(d_1, ..., d_k)$ with $\sigma \in \Sigma^{(k)}_{\text{root}}$ and $d_1, ..., d_k \in \text{IS}(A)$ are acyclic (cf. [Knu68, FV98]).

We will use the following lemma that relates is-dependencies to normal forms. It can easily be proved using Lemma 3.12.

Lemma 3.14. *Let $s = \sigma(s_1, ..., s_k)$, $\sigma \in \Sigma^{(k)}_{\text{root}}$, and $s_1, ..., s_k \in T_\Sigma$. Then, for $a \in \text{Syn}$, $b \in \text{Inh}$, $c \in \text{Att}$, and $0 \leqslant i \leqslant k$,*

(i) $\text{nf}(\Rightarrow_{A,s}, \langle a, \varepsilon\rangle)$ *contains* $\langle b, \varepsilon\rangle$ *if and only if* $(b, a) \in \text{is}(s)$.

(ii) $\text{nf}(\Rightarrow_{A,s}, \langle c, i\rangle)$ *contains* $\langle b, \varepsilon\rangle$ *if and only if there is a path from* $\langle b, \pi\rangle$ *to* $\langle c, \pi i\rangle$ *in* $D_\sigma(\text{is}(s_1), ..., \text{is}(s_k))$.

Note that $\text{is}(\text{root}(s)) = \varnothing$; in (i) above, this corresponds to the fact that $\text{nf}(\Rightarrow_{A,\text{root}(s)}, \langle a_0, \varepsilon\rangle) \in T_\Delta$.

The dependency graph of a tree s has as nodes all attribute instances $\langle \text{Att}, \text{Occ}(s)\rangle$ and as edges the dependencies according to the rules in R.

Definition 3.15 (Dependency graph). Let $s = \sigma(s_1, ..., s_k)$, $\sigma \in \Sigma^{(k)}_{\text{root}}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_\Sigma$. The *dependency graph of s*, denoted by $D(s)$, is the graph (V, E) with $V = \langle \text{Att}, \text{Occ}(s)\rangle$ and $E = \{(\langle c, ui\rangle, \langle c', uj\rangle) \in V \times V \mid i, j \in \mathbb{N},\ \langle c, \pi i\rangle$ occurs in $\text{rhs}(\langle c', \pi j\rangle, s[u])\}$.

Attributed Relabelings. A translation τ from trees to trees is called a *relabeling*, if for $(s, t) \in \tau$, t is obtained from s by merely changing the labels of the nodes of s. The classes *DBQREL* and *DTQREL* of finite state relabelings of [Eng77, Eng75], which are based on bottom-up and top-down tree transducers, respectively, are well-known classes of (partial) relabelings. The class *T-REL* of top-down relabelings (Definition 3.1) is, in fact, the class of total relabelings in *DTQREL*, and T^R-*REL* is its obvious extension with regular look-ahead. If we denote by *B-REL* the class of total relabelings in *DBQREL*, then it is easy to show that *B-REL* $\subseteq T^R$-*REL* (cf. Lemma 2.10(3) of [Eng77]).

In [BE98] a class of relabelings which is based on attribute grammars was considered. We will show in Section 4 that these *attributed relabelings* have the same power as top-down relabelings with regular look-ahead (T^R-*REL*). An attributed relabeling is an attribute grammar, all attributes of which may only have finitely many values. Depending on these values the labels of the output tree are computed.

We want to define attributed relabelings in terms of ATTs. Since it is not possible for an ATT to compute a label according to values of other attributes, we need to add the ability to evaluate expressions in an algebra W over finite domains (cf. [CF82]). An ordinary ATT A can be used to compute the new label for each node as a W-expression (in some attribute a_0). For an input tree s the relabeled tree t is obtained by replacing the label of each node u by the value of $\mathrm{nf}(\Rightarrow_{A,\,\mathrm{root}(s)}, \langle a_0, 1u\rangle)$, interpreted in W. Note that $1u$ is the node of $\mathrm{root}(s)$ corresponding to node u of s.

Definition 3.16 (Attributed relabeling). An *attributed relabeling* is a triple $\mathscr{A} = (A, \Delta, W)$, where $A = (\mathrm{Syn}, \mathrm{Inh}, \Sigma, \Gamma, \mathrm{root}, a_0, R)$ is an ATT, Δ is a ranked alphabet, and W is a function such that for every $c \in \mathrm{Att}$, $W(c)$ is a finite set, $W(a_0) = \Delta$, and for every $\gamma \in \Gamma^{(k)}$ with $k \geqslant 0$, $W(\gamma)$ is a finite function. For $\sigma \in \Sigma^{(k)}_{\mathrm{root}}$, each rule in R_σ is of the form $\langle c, \rho\rangle \to \gamma(\langle d_1, \rho_1\rangle, ..., \langle d_n, \rho_n\rangle)$ with $\langle c, \rho\rangle \in \mathrm{ins}_\sigma$, $\gamma \in \Gamma^{(n)}$, $n \geqslant 0$, $\langle d_i, \rho_i\rangle \in \mathrm{outs}_\sigma$ for $i \in [n]$, $W(\gamma)\colon W(d_1) \times \cdots \times W(d_n) \to W(c)$, and if $c = a_0$, then $W(\gamma)(s_1, ..., s_n) \in \Delta^{(k)}$ for all $s_i \in W(d_i)$.

The *translation realized by* $\mathscr{A}$, denoted by $\tau_{\mathscr{A}}$, is the total function

$$\{(s, t) \in T_\Sigma \times T_\Delta \mid \mathrm{Occ}(s) = \mathrm{Occ}(t),$$
$$\forall u \in \mathrm{Occ}(s)\colon t[u] = \mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,\,\mathrm{root}(s)}, \langle a_0, 1u\rangle))\},$$

where val_W is the (partial) *valuation function* induced by the Γ-algebra W, defined for all well-typed trees in T_Γ in the usual way.

Note that the translation realized by $\mathscr{A}$ does not depend on the $(\langle a_0, \pi\rangle, \mathrm{root})$-rule of A.

The class of all translations which can be realized by attributed relabelings is denoted by *ATT-REL*. An *attributed tree transducer with look-ahead* (for short, $\mathrm{ATT^R}$) M consists of an attributed relabeling $\mathscr{A}$ and an ATT B. Intuitively, $\mathscr{A}$ gathers "look-ahead" information that can be used by B. The translation realized by M is the composition $\tau_{\mathscr{A}} \circ \tau_B$. The class $\mathit{ATT\text{-}REL} \circ \mathit{ATT}$ of all translations which can be realized by $\mathrm{ATT^R}$s is denoted by ATT^R. It should be clear that the $\mathrm{ATT^R}$s realize the same class of translations as those of [BE98] which are defined using a model of attribute grammars closely related to ATTs. As discussed in the Introduction, it is shown in [BE98] that the class of MSO definable tree translations equals $\mathit{ATT}^R_{\mathrm{sur}}$, the class of translations realized by single use restricted $\mathrm{ATT^R}$s.

4. TOP-DOWN AND ATTRIBUTED RELABELINGS

In the next section we want to prove our first main result: the classes $\mathit{ATT}^R_{\mathrm{sur}}$ and $\mathit{MTT}^R_{\mathrm{sur}}$ of translations realized by single use restricted $\mathrm{ATT^R}$s and single use restricted $\mathrm{MTT^R}$s, respectively, are equal (Theorem 5.14). By definition, $\mathit{ATT}^R_{\mathrm{sur}} = \mathit{ATT\text{-}REL} \circ \mathit{ATT}_{\mathrm{sur}}$ (cf. Definition 5.1). Thus, as a first step towards this result, we characterize the class *ATT-REL* in terms of (very simple) macro tree transducers; we prove in this section that the classes *ATT-REL* and T^R-*REL* are equal. The proof is split into two lemmas.

LEMMA 4.1. *T^R-REL $\subseteq$ ATT-REL.*

Proof. Let $M = (Q, P, \Sigma, \Delta, q_{\text{init}}, R, h)$ be a T^R-REL. We will construct an attributed relabeling $\mathscr{A} = (A, \Delta, W)$ such that $\tau_{\mathscr{A}} = \tau_M$. For each node u of the input tree s of M, the ATT A computes in an inherited attribute the state q in which M processes the subtree s/u (i.e., the unique state q in the state sequence of s at u). The look-ahead state p of M on s/u is computed in a synthesized attribute.

Let $A = (\text{Syn}, \text{Inh}, \Sigma, \Gamma, \text{root}, a_0, R)$ with

- $\text{Syn} = \{a_0, p\}$
- $\text{Inh} = \{q\}$
- $\Gamma = \{\text{init}^{(0)}\} \cup \{l_\sigma^{(k+1)} \mid \sigma \in \Sigma^{(k)}\} \cup \{g_\sigma^{(k)} \mid \sigma \in \Sigma^{(k)}\} \cup \{f_{\sigma,i}^{(k+1)} \mid \sigma \in \Sigma^{(k)},\ i \in [k]\}$
- For $\sigma \in \Sigma^{(k)}$, R_σ contains the rules

$$\langle a_0, \pi \rangle \to l_\sigma(\langle q, \pi \rangle, \langle p, \pi 1 \rangle, ..., \langle p, \pi k \rangle)$$

$$\langle p, \pi \rangle \to g_\sigma(\langle p, \pi 1 \rangle, ..., \langle p, \pi k \rangle)$$

$$\langle q, \pi i \rangle \to f_{\sigma, i}(\langle q, \pi \rangle, \langle p, \pi 1 \rangle, ..., \langle p, \pi k \rangle) \qquad \text{for every} \quad i \in [k].$$

$$R_{\text{root}} = \{\langle q, \pi 1 \rangle \to \text{init}, \langle a_0, \pi \rangle \to \langle a_0, \pi 1 \rangle\}.$$

It is straightforward to show that $\text{is}(s) = \{(q, a_0)\}$ for every $s \in T_\Sigma$ and that, consequently, A is noncircular.

The mapping W is defined as $W(a_0) = \Delta$, $W(p) = P$, and $W(q) = Q$ for the attributes of A and, as follows, for the symbols in Γ. Let $W(\text{init}) = q_{\text{init}}$ and for every $\sigma \in \Sigma^{(k)}$, $r \in Q$, and $p_1, ..., p_k \in P$, $W(l_\sigma)(r, p_1, ..., p_k) = \text{rhs}_M(r, \sigma, \langle p_1, ..., p_k \rangle)[\varepsilon]$, $W(g_\sigma)(p_1, ..., p_k) = h_\sigma(p_1, ..., p_k)$, and for $i \in [k]$, $W(f_{\sigma,i})(r, p_1, ..., p_k) = q_i$, where $q_i \in Q$ is the state such that $\text{rhs}_M(r, \sigma, \langle p_1, ..., p_k \rangle)[i] = \langle q_i, x_i \rangle$.

Although it is quite obvious that $\tau_{\mathscr{A}} = \tau_M$, we will give a detailed correctness proof in order to illustrate the use of Lemmas 3.12 and 3.4. In the correctness proof below, we assume that the valuation function val_W is extended to trees in $T_\Gamma(Q \cup P)$ in the obvious way.

In the following let $s \in T_\Sigma$ and $u \in \text{Occ}(s)$. By the definition of attributed relabelings, $\tau_{\mathscr{A}}(s)[u] = \text{val}_W(\text{nf}(\Rightarrow_{A, \text{root}(s)}, \langle a_0, 1u \rangle))$. By Lemma 3.12(1), this equals $\text{val}_W(\text{nf}(\Rightarrow_{A,s}, \langle a_0, u \rangle)\Pi)$, where $\Pi = [\langle b, \varepsilon \rangle \leftarrow \text{nf}(\Rightarrow_{A, \text{root}(s)}, \langle b, 1 \rangle) \mid b \in \text{Inh}]$. Since $\text{Inh} = \{q\}$ and $\text{nf}(\Rightarrow_{A, \text{root}(s)}, \langle q, 1 \rangle) = \text{init}$, $\Pi = [\langle q, \varepsilon \rangle \leftarrow \text{init}]$. By applying val_W inside Π we get $\text{val}_W(\text{nf}(\Rightarrow_{A,s}, \langle a_0, u \rangle)[\langle q, \varepsilon \rangle \leftarrow q_{\text{init}}])$. By point (b) of the claim below (with $r = q_{\text{init}}$) this means that $\tau_{\mathscr{A}}(s)[u] = M_{q_{\text{init}}}(s)[u]$ which proves the correctness of the construction.

Claim. (a) $\text{val}_W(\text{nf}(\Rightarrow_{A,s}, \langle p, u \rangle)) = h(s/u)$ and (b) for $r \in Q$, $\text{val}_W(\text{nf}(\Rightarrow_{A,s}, \langle a_0, u \rangle)[\langle q, \varepsilon \rangle \leftarrow r]) = M_r(s)[u]$.

Proof. Both statements are proved by induction on the structure of s. Let $s = \sigma(s_1, ..., s_k)$, $k \geqslant 0$, $\sigma \in \Sigma^{(k)}$, and $s_1, ..., s_k \in T_\Sigma$.

(a) If $u=\varepsilon$, then by Lemma 3.12(2) and (1), $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle p, \varepsilon\rangle)) = \mathrm{val}_W(\zeta\Theta\Psi)$ with $\zeta = \mathrm{rhs}_A(\langle p, \pi\rangle, \sigma)$,

$$\Theta = [\langle a, \pi j\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s_j}, \langle a, \varepsilon\rangle)$$
$$[\langle b, \varepsilon\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle b, j\rangle) \mid b \in \mathrm{Inh}] \mid a \in \mathrm{Syn},\ j \in [k]],$$

and $\Psi = [\langle b, \pi\rangle \leftarrow \langle b, \varepsilon\rangle \mid b \in \mathrm{Inh}]$. Since $\zeta = g_\sigma(\langle p, \pi 1\rangle, ..., \langle p, \pi k\rangle)$, we can reduce the substitution Θ to synthesized attributes of the form $\langle p, \pi j\rangle$. Applying val_W, Θ becomes

$$[\langle p, \pi j\rangle \leftarrow \mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_j}, \langle p, \varepsilon\rangle)[\langle b, \varepsilon\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle b, j\rangle) \mid b \in \mathrm{Inh}]) \mid j \in [k]]$$

which equals $[\langle p, \pi j\rangle \leftarrow h(s_j) \mid j \in [k]]$ by the induction hypothesis. Note that no elements of $\langle \mathrm{Inh}, \{\varepsilon\}\rangle$ occur in $\mathrm{nf}(\Rightarrow_{A,s_j}, \langle p, \varepsilon\rangle)$. Thus, $\mathrm{val}_W(\zeta\Theta\Psi) = \mathrm{val}_W(g_\sigma(h(s_1), ..., h(s_k))\Psi) = h_\sigma(h(s_1), ..., h(s_k)) = h(s) = h(s/\varepsilon)$.

Otherwise $u = iv$ with $i \in [k]$ and $v \in \mathrm{Occ}(s_i)$. Then by Lemma 3.12(1) $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle p, iv\rangle)) = \mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_i}, \langle p, v\rangle))$ which equals $h(s_i/v) = h(s/iv)$ by the induction hypothesis.

(b) If $u=\varepsilon$, then $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle a_0, \varepsilon\rangle)[\langle q, \varepsilon\rangle \leftarrow r]) = \mathrm{val}_W(\zeta\Theta\Psi[\langle q, \varepsilon\rangle \leftarrow r])$, where $\zeta = \mathrm{rhs}_A(\langle a_0, \pi\rangle, \sigma)$ and Θ and Ψ are as above. By Claim (a) we can replace $\langle p, \pi j\rangle$ by $h(s_j)$ in $\zeta = l_\sigma(\langle q, \pi\rangle, \langle p, \pi 1\rangle, ..., \langle p, \pi k\rangle)$ to get $\mathrm{val}_W(l_\sigma(r, h(s_1), ..., h(s_k)))$. By the definition of $W(l_\sigma)$, this equals $\mathrm{rhs}_M(r, \sigma, \langle h(s_1), ..., h(s_k)\rangle)[\varepsilon] = M_r(s)[\varepsilon]$.

Otherwise $u = iv$ with $i \in [k]$ and $v \in \mathrm{Occ}(s_i)$. By Lemma 3.12(1) $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle a_0, iv\rangle)[\langle q, \varepsilon\rangle \leftarrow r]) = \mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_i}, \langle a_0, v\rangle)[\langle q, \varepsilon\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle q, i\rangle)][\langle q, \varepsilon\rangle \leftarrow r])$. With Lemma 3.12(2) we get $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_i}, \langle a_0, v\rangle)[\langle q, \varepsilon\rangle \leftarrow \zeta\Theta\Psi[\langle q, \varepsilon\rangle \leftarrow r]])$, where ζ equals $\mathrm{rhs}_A(\langle q, \pi i\rangle, \sigma)$ and Θ and Ψ are as above. By Claim (a), we can replace $\langle p, \pi j\rangle$ by $h(s_j)$ in $\zeta = f_{\sigma,i}(\langle q, \pi\rangle, \langle p, \pi 1\rangle, ..., \langle p, \pi k\rangle)$ to get $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_i}, \langle a_0, v\rangle)[\langle q, \varepsilon\rangle \leftarrow f_{\sigma,i}(r, h(s_1), ..., h(s_k))])$. Let $\xi = \mathrm{rhs}_M(r, \sigma, \langle h(s_1), ..., h(s_k)\rangle) = \delta(\langle r_1, x_1\rangle, ..., \langle r_k, x_k\rangle)$. If we apply val_W inside the substitution, then since $\mathrm{val}_W(f_{\sigma,i}(r, h(s_1), ..., h(s_k))) = r_i$, we get $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_i}, \langle a_0, v\rangle)[\langle q, \varepsilon\rangle \leftarrow r_i])$ which equals, by the induction hypothesis, $M_{r_i}(s_i)[v] = \delta(M_{r_1}(s_1), ..., M_{r_k}(s_k))[iv] = \xi[\![\langle q', x_i\rangle \leftarrow M_{q'}(s_i) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!][iv] = M_r(s)[iv]$ by Lemma 3.4. ∎

In the proof of the next lemma we first present the formal construction and then explain the intuition behind it (and finally, of course, we prove the correctness of the construction).

LEMMA 4.2. *ATT-REL* $\subseteq T^R$*-REL*.

Proof. Let $\mathscr{A} = (A, \Delta, W)$ be an attributed relabeling with $A = (\mathrm{Syn}, \mathrm{Inh}, \Sigma, \Gamma, \mathrm{root}, a_0, R)$. The T^R-REL $M = (Q, P, \Sigma, \Delta, q_{\mathrm{init}}, \bar{R}, h)$ is defined as follows. Let $Q = \{q_{\mathrm{init}}\} \cup Q'$, where Q' consists of all mappings which associate with every $c \in \mathrm{Att}$ a value in $W(c)$. For $\sigma \in \Sigma^{(k)}$ and $q_0, q_1, ..., q_k \in Q$ we denote by $R_\sigma(q_0, q_1, ..., q_k)$ the predicate which is true if and only if all rules in R_σ are consistent with the values given by the mappings $q_0, q_1, ..., q_k$; this means that for every rule $\langle c, \pi i\rangle \to \gamma(\langle d_1, \pi i_1\rangle, ..., \langle d_n, \pi i_n\rangle)$ in R_σ, $q_i(c) = W(\gamma)(q_{i_1}(d_1), ..., q_{i_n}(d_n))$

(recall that $\pi 0$ denotes π). Similarly, $R_{\text{root}}(q_1)$ denotes the predicate which is true if and only if all rules of the form $\langle b, \pi 1\rangle \to \zeta$ in R_{root} with $b \in \text{Inh}$ are consistent with the values given by q_1. Let $P = \mathscr{P}(Q')$ and define the look-ahead automaton of M as follows. For $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $p_1, ..., p_k \in P$, let $h_\sigma(p_1, ..., p_k) = \{q \in Q' \mid \text{there are unique } q_i \in p_i \text{ for } i \in [k] \text{ such that } R_\sigma(q, q_1, ..., q_k)\}$. For $q \in Q$, $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $p_1, ..., p_k \in P$, let

$$\langle q, \sigma(x_1, ..., x_k)\rangle \to \delta(\langle q_1, x_1\rangle, ..., \langle q_k, x_k\rangle) \qquad \langle p_1, ..., p_k\rangle$$

be in $\bar{R}$, where $\delta \in \Delta^{(k)}$ and $q_1, ..., q_k \in Q'$ are defined as

- $q \in Q'$. If there are unique $q_i \in p_i$ for all $i \in [k]$ such that $R_\sigma(q, q_1, ..., q_k)$, then these are the $q_1, ..., q_k$ and $\delta = q(a_0)$; otherwise δ and $q_1, ..., q_k$ are chosen arbitrarily.

- $q = q_{\text{init}}$. If there are unique $q_0 \in h_\sigma(p_1, ..., p_k)$ and $q_i \in p_i$ for all $i \in [k]$ such that $R_{\text{root}}(q_0)$ and $R_\sigma(q_0, q_1, ..., q_k)$, then these are the $q_1, ..., q_k$ and $\delta = q_0(a_0)$; otherwise δ and $q_1, ..., q_k$ are chosen arbitrarily.

Let us explain the construction informally (an example is given in Example 4.3). From now on let val_W be extended to $T_\Gamma(\bigcup_{c \in \text{Att}} W(c))$ in the obvious way. Let $s \in T_\Sigma$. Every attribute instance $\langle c, u\rangle \in \langle \text{Att}, \text{Occ}(s)\rangle$ has a unique normal form with respect to $\Rightarrow_{A,s}$. It is a tree in $T_\Gamma(\langle \text{Inh}, \{\varepsilon\}\rangle)$ (note that $s[\varepsilon] \neq \text{root}$). The look-ahead automaton of M is defined in such a way that $h(s)$ contains all mappings q in Q' such that for every $a \in \text{Syn}$, $q(a) = \text{val}_W(\text{nf}(\Rightarrow_{A,s}, \langle a, \varepsilon\rangle)[\langle b, \varepsilon\rangle \leftarrow q(b) \mid b \in \text{Inh}])$. Hence, if we fix the values of the inherited attributes at the root of $s = \sigma(s_1, ..., s_k)$ by a mapping $q \in Q'$, then the rules in R_σ fix all mappings $q_1, ..., q_k \in Q'$ at the subtrees $s_1, ..., s_k$ (with $q_i \in h(s_i)$). In particular this means that the value of a_0 at the root of s is fixed as $q(a_0)$ which, by the definition of attributed relabeling, is the symbol which replaces σ in s. Hence for every $q \in Q'$, $\sigma \in \Sigma^{(k)}$, and $p_1, ..., p_k \in P$, the states $q_1, ..., q_k \in Q'$ and the symbol $\delta \in \Delta$ are uniquely determined. For the tree $\text{root}(s)$, the rules in R_{root} together with $h_\sigma(h(s_1), ..., h(s_k))$ fix the state q_0 at the root of s, and thus, by the arguments above, the states $q_1, ..., q_k$ which must process the subtrees $s_1, ..., s_k$. Thus, for every node u of s that is not the root, the state q in which M processes the subtree s/u consists of the values of the attributes of that node in $\text{root}(s)$.

We now prove the correctness of M, starting with the correctness of the look-ahead automaton. In the following let, for $q \in Q'$, $\Theta_q = [\langle b, \varepsilon\rangle \leftarrow q(b) \mid b \in \text{Inh}]$ and let, for $s \in T_\Sigma$, $\text{ism}(s) = \{q \in Q' \mid \forall a \in \text{Syn}: q(a) = \text{val}_W(\text{nf}(\Rightarrow_{A,s}, \langle a, \varepsilon\rangle)\, \Theta_q)\}$. We want to prove that for every $s \in T_\Sigma$, $h(s) = \text{ism}(s)$.

Let $s = \sigma(s_1, ..., s_k)$ with $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_\Sigma$. Let (V, E) be the dependency graph $D_\sigma(\text{is}(s_1), ..., \text{is}(s_k))$ and define $V_0 = \{v \in V \mid \neg\exists v' \in V \text{ such that } (v', v) \in E\}$ and for $n \geqslant 1$, $V_n = \{v \in V \mid \forall v' \in V, \text{ if } (v', v) \in E, \text{ then } v' \in V_\nu \text{ for some } \nu < n\}$. Since A is noncircular, (V, E) is acyclic and each $v \in V$ is in a unique V_n.

First we show that for $q_i \in \text{ism}(s_i)$ and an arbitrary $q_0 \in Q'$ such that $R_\sigma(q_0, q_1, ..., q_k)$, the values of all attributes in $\langle \text{Att}, \{0, ..., k\}\rangle$ are correct with respect to the values of the inherited attributes given by q_0.

Claim 1. If $\langle c, \pi j\rangle \in \mathrm{ins}_\sigma \cup \mathrm{outs}_\sigma$, $q_0 \in Q'$, and, for $i \in [k]$, $q_i \in \mathrm{ism}(s_i)$ such that $R_\sigma(q_0, q_1, ..., q_k)$, then $q_j(c) = \mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle c, j\rangle)\, \Theta_{q_0})$.

Proof. By induction on n, where $v = \langle c, \pi j\rangle \in V_n$.

If $c \in \mathrm{Inh}$ and $j = 0$, then $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle c, \varepsilon\rangle)\, \Theta_{q_0}) = \mathrm{val}_W(\langle c, \varepsilon\rangle\, \Theta_{q_0}) = q_0(c)$.

If $\langle c, j\rangle \in \langle \mathrm{Syn}, [k]\rangle$, then by Lemma 3.12(1), $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle c, j\rangle)\, \Theta_{q_0})$ equals $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_j}, \langle c, \varepsilon\rangle)\, \Psi_j \Theta_{q_0})$, where $\Psi_j = [\langle b, \varepsilon\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle b, j\rangle) \mid b \in \mathrm{Inh}]$. Since substitution is associative we can move Θ_{q_0} inside Ψ_j. Now the substitution Ψ_j can be reduced to those attributes $\langle b, \varepsilon\rangle$ which appear in $\mathrm{nf}(\Rightarrow_{A,s_j}, \langle c, \varepsilon\rangle)$. By Lemma 3.14(i) and the definition of V_n, the corresponding attributes $\langle b, j\rangle$ are in V_v for some $v < n$. Thus, after applying val_W inside the substitution, we can use the induction hypothesis to get $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_j}, \langle c, \varepsilon\rangle)\, \Theta_{q_j})$. Since $q_j \in \mathrm{ism}(s_j)$, this equals $q_j(c)$.

If $\langle c, \pi j\rangle \in \mathrm{ins}_\sigma$, then by Lemma 3.12(2), $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle c, j\rangle)\, \Theta_{q_0}) = \mathrm{val}_W(\zeta \Theta_1 \Theta_2 \Theta_{q_0})$, where $\zeta = \mathrm{rhs}(\langle c, \pi j\rangle, \sigma)$, $\Theta_1 = [\langle a, \pi i\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle a, i\rangle) \mid a \in \mathrm{Syn}, i \in [k]]$, and $\Theta_2 = [\langle b, \pi\rangle \leftarrow \langle b, \varepsilon\rangle \mid b \in \mathrm{Inh}]$. We can move Θ_{q_0} inside Θ_1 and Θ_2 (by associativity and since $\zeta \Theta_{q_0} = \zeta$). By applying val_W inside Θ_1 this means that $\langle a, \pi i\rangle$ is replaced by $\mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle a, i\rangle)\, \Theta_{q_0})$. Since, by the definition of V_n, $\langle a, \pi i\rangle \in V_v$ for some $v < n$, this equals $q_i(a)$ by the induction hypothesis. Altogether we get $\mathrm{val}_W(\zeta[\langle a, \pi i\rangle \leftarrow q_i(a) \mid a \in \mathrm{Syn}, i \in [k]][\langle b, \pi\rangle \leftarrow q_0(b) \mid b \in \mathrm{Inh}])$. By $R_\sigma(q_0, q_1, ..., q_k)$ this equals $q_j(c)$. This ends the proof of Claim 1.

The correctness of the look-ahead, i.e., $h(s) = \mathrm{ism}(s)$ follows directly from Claim 2.

Claim 2. $q_0 \in \mathrm{ism}(s)$ if and only if there are unique $q_i \in \mathrm{ism}(s_i)$ for $i \in [k]$ such that $R_\sigma(q_0, q_1, ..., q_k)$.

Proof. The if-part of Claim 2 follows from Claim 1 with $j = 0$ and $c \in \mathrm{Syn}$.

We now show the only-if-part. The uniqueness of the q_j is immediate from Claim 1. It remains to prove their existence. For $n \geqslant 0$ and $v = \langle c, \pi j\rangle \in V_n$ with $j \in [k]$ define

$$q_j(c) = \begin{cases} \mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s_j}, \langle c, \varepsilon\rangle)\, \Theta_{q_j}), \\ \qquad \text{if} \quad c \in \mathrm{Syn}, \\ \mathrm{val}_W(\zeta[\langle a, \pi i\rangle \leftarrow q_i(a) \mid a \in \mathrm{Syn}, i \in [k]][\langle b, \pi\rangle \leftarrow q_0(b) \mid b \in \mathrm{Inh}]), \\ \qquad \text{if} \quad c \in \mathrm{Inh}, \end{cases}$$

where $\zeta = \mathrm{rhs}(\langle c, \pi j\rangle, \sigma)$. All $\langle a, \pi i\rangle$ which appear in ζ are in V_v for some $v < n$ and, by Lemma 3.14(i), the same is true for all $\langle b, \pi j\rangle$ such that $\langle b, \varepsilon\rangle$ occurs in $\mathrm{nf}(\Rightarrow_{A,s_j}, \langle c, \varepsilon\rangle)$. Hence, the above defines all q_j by induction on n. Clearly, $q_j \in \mathrm{ism}(s_j)$ for $j \in [k]$. Now define $q'_0 \in Q'$ as follows. For $a \in \mathrm{Syn}$ let $q'_0(a) = \mathrm{rhs}(\langle a, \pi\rangle, \sigma)[\langle \alpha, \pi i\rangle \leftarrow q_i(\alpha) \mid \alpha \in \mathrm{Syn},\ i \in [k]][\langle b, \pi\rangle \leftarrow q_0(b) \mid b \in \mathrm{Inh}]$ and for $b \in \mathrm{Inh}$ let $q'_0(b) = q_0(b)$. Then $R_\sigma(q'_0, q_1, ..., q_k)$. By Claim 1, $q'_0(a) = \mathrm{val}_W(\mathrm{nf}(\Rightarrow_{A,s}, \langle a, \varepsilon\rangle)\, \Theta_{q'_0})$ for all $a \in \mathrm{Syn}$. Since $\Theta_{q'_0} = \Theta_{q_0}$ and, by assumption, $q_0 \in \mathrm{ism}(s)$, this means that $q'_0 = q_0$ and, hence, $R_\sigma(q_0, q_1, ..., q_k)$. This ends the proof of Claim 2. ∎

Let us now prove the correctness of τ_M. Let $s \in T_\Sigma$. Analogous to Claims 1 and 2, but using the (acyclic) graph $(V, E) = D_{\text{root}}(\text{is}(s))$, the following two claims can easily be shown.

Claim 1 (Root). If $q_0 \in \text{ism}(s)$ and $R_{\text{root}}(q_0)$, then $q_0(c) = \text{val}_W(\text{nf}(\Rightarrow_{A, \text{root}(s)}, \langle c, 1\rangle))$ for every $c \in \text{Att}$.

Claim 2 (Root). There is a unique $q_0 \in \text{ism}(s)$ such that $R_{\text{root}}(q_0)$.

It should now be clear from the definition of the rules of M that $M_{q_{\text{init}}}(s) = M_{q_0}(s)$, where q_0 is the unique element of $\text{ism}(s)$ with $R_{\text{root}}(q_0)$.
The correctness of τ_M, i.e., that $M_{q_{\text{init}}}(s)[u] = \text{val}_W(\text{nf}(\Rightarrow_{A, \text{root}(s)}, \langle a_0, 1u\rangle))$ then follows from Claim 3 with Lemma 3.12(1) and the fact that, by Claim 1 (Root), $\Theta_{q_0} = [\langle b, \varepsilon\rangle \leftarrow \text{val}_W(\text{nf}(\Rightarrow_{A, \text{root}(s)}, \langle b, 1\rangle)) \mid b \in \text{Inh}]$.

Claim 3. For every $q \in h(s)$ and every $u \in \text{Occ}(s)$, $M_q(s)[u] = \text{val}_W(\text{nf}(\Rightarrow_{A, s}, \langle a_0, u\rangle)\, \Theta_q)$.

Proof. By induction on the structure of s. Let $s = \sigma(s_1, ..., s_k)$ with $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_\Sigma$. Then $M_q(s)[u] = \zeta[\![\cdots]\!][u]$, where $\zeta = \text{rhs}_M(q, \sigma, \langle h(s_1), ..., h(s_k)\rangle)$ and $[\![\cdots]\!] = [\![\langle q', x_i\rangle \leftarrow M_{q'}(s_i) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!]$. Since $q \in h(s)$, there are unique $q_i \in h(s_i)$ for $i \in [k]$ such that $R_\sigma(q, q_1, ..., q_k)$. Then $\zeta = \delta(\langle q_1, x_1\rangle, ..., \langle q_k, x_k\rangle)$ with $\delta = q(a_0)$. Note that, by Claim 2, $q \in \text{ism}(s)$ and $q_i \in \text{ism}(s_i)$.
If $u = \varepsilon$ then $M_q(s)[\varepsilon] = \zeta[\![\cdots]\!][\varepsilon] = q(a_0)$. Since $q \in \text{ism}(s)$ this equals $\text{val}_W(\text{nf}(\Rightarrow_{A, s}, \langle a_0, u\rangle)\, \Theta_q)$.
If $u = iv$ with $i \in [k]$ and $v \in \text{Occ}(s_i)$, then $M_q(s)[iv] = \zeta[\![\cdots]\!][iv] = \delta(M_{q_1}(s_1), ..., M_{q_k}(s_k))[iv] = M_{q_i}(s_i)[v]$ which, by induction, equals $\text{val}_W(\text{nf}(\Rightarrow_{A, s_i}, \langle a_0, v\rangle)\, \Theta_{q_i}$. By Claim 1, $q_i(b) = \text{val}_W(\text{nf}(\Rightarrow_{A, s}, \langle b, i\rangle)\, \Theta_q)$ for all $b \in \text{Inh}$. Thus, omitting val_W and moving Θ_q outside of Θ_{q_i}, the above becomes $\text{val}_W(\text{nf}(\Rightarrow_{A, s_i}, \langle a_0, v\rangle)$ $[\langle b, \varepsilon\rangle \leftarrow \text{nf}(\Rightarrow_{A, s}, \langle b, i\rangle) \mid b \in \text{Inh}]\, \Theta_q)$ which, by Lemma 3.12(1), equals $\text{val}_W(\text{nf}(\Rightarrow_{A, s}, \langle a_0, iv\rangle)\, \Theta_q)$. This ends the proof of Claim 3. ∎

The following example illustrates the construction in the proof of Lemma 4.2.

EXAMPLE 4.3. Let $\mathscr{A} = (A, \Delta, W)$ with $A = (\text{Syn}, \text{Inh}, \Sigma, \Gamma, \text{root}, a_0, R)$ with $\text{Syn} = \{a_0, \text{at}\}$, $\text{Inh} = \{\text{below}\}$, $\Sigma = \{\gamma^{(1)}, *^{(1)}, e^{(0)}\}$, $\Delta = \{\gamma^{(1)}, \#^{(1)}, *^{(1)}, e^{(0)}\}$, and $\Gamma = \{l_\#^{(2)}, l_*^{(0)}, l_e^{(0)}, 0^{(0)}, 1^{(0)}\}$. In the sequel let $\mathbb{B} = \{0, 1\}$. Let $W(\text{at}) = \mathbb{B}$, $W(\text{below}) = \mathbb{B}$, $W(a_0) = \Delta$, $W(l_\#)\colon \mathbb{B} \times \mathbb{B} \to \Delta$, $W(l_*), W(l_e) \in \Delta$, $W(0), W(1) \in \mathbb{B}$ and let $W(l_\#)(x, y)$ equal γ if $x = y = 0$ and $\#$ otherwise, $W(l_*)() = *$, $W(l_e)() = e$, $W(0)() = 0$, and $W(1)() = 1$.
R consists of the following sets of rules:

$$R_\gamma = \{\langle \text{below}, \pi 1\rangle \to 0, \langle \text{at}, \pi\rangle \to 0, \langle a_0, \pi\rangle \to l_\#(\langle \text{below}, \pi\rangle, \langle \text{at}, \pi 1\rangle)\},$$
$$R_* = \{\langle \text{below}, \pi 1\rangle \to 1, \langle \text{at}, \pi\rangle \to 1, \langle a_0, \pi\rangle \to l_*\},$$
$$R_e = \{\langle \text{at}, \pi\rangle \to 0, \langle a_0, \pi\rangle \to l_e\}, \text{ and}$$
$$R_{\text{root}} = \{\langle \text{below}, \pi 1\rangle \to 0, \langle a_0, \pi\rangle \to \langle a_0, \pi 1\rangle\},$$

The attributed relabeling $\mathscr{A}$ takes a monadic tree s over Σ as input and generates a tree over Δ which is obtained from s by changing all γ's occurring directly above or directly below a star in s into a cross ($\#$). Thus, the tree $s = \gamma\gamma{*}\gamma\gamma e$ is translated by $\mathscr{A}$ into the tree $\gamma\#{*}\#\gamma e$ (in monadic trees we might leave out the parentheses). If $s[u] = *$, then the attributes $\langle \text{at}, u\rangle$ and $\langle \text{below}, u1\rangle$ have the value 1 (and otherwise 0). The reader may verify that for $s = \gamma\gamma * \gamma\gamma e$, $\text{nf}(\Rightarrow_{A,\,\text{root}(s)}, \langle a_0, u\rangle)$ equals $l_{\#}(0, 0)$ for $u = 1$, $l_{\#}(0, 1)$ for $u = 11$, l_{*} for $u = 1^3$, $l_{\#}(1, 0)$ for $u = 1^4$, $l_{\#}(0, 0)$ for $u = 1^5$, and l_e for $u = 1^6$. Hence, $\tau_{\mathscr{A}}(s) = \gamma\# * \#\gamma e$.

Let us now construct a top-down relabeling M with regular look-ahead by the construction given in the proof of Lemma 4.2. Note that the translation realized by $\mathscr{A}$ can neither be realized by a top-down relabeling nor by a bottom-up relabeling, i.e., $\tau_M \notin T\text{-}REL \cup B\text{-}REL$ (cf. the discussion in Section 3). Let $M = (Q' \cup \{q_{\text{init}}\}, P, \Sigma, \Delta, q_{\text{init}}, \bar{R}, h)$. The set Q' equals $\{q_{bb'g} = \{(\text{below}, b), (\text{at}, b'), (a_0, g)\} \mid b, b' \in \mathbb{B}, g \in \Delta\}$ and $P = \mathscr{P}(Q')$. The rules of the look-ahead automaton of M are $h_e() = \{q_{00e}, q_{10e}\}$ and for $p \in P$, $h_*(p) = \{q_{01*}, q_{11*}\}$ if p contains exactly one state of the form q_{1bg} with $b \in \mathbb{B}$ and $g \in \Delta$, and $h_*(p) = \varnothing$ otherwise; $h_\gamma(p) = \{q_{00\gamma}, q_{10\#}\}$ if p contains exactly one state of the form q_{00g} with $g \in \Delta$ but no state of the form q_{01g}, $h_\gamma(p) = \{q_{00\#}, q_{10\#}\}$ if p contains exactly one q_{01g} but no q_{00g}, $h_\gamma(p) = \{q_{00\gamma}, q_{00\#}\}$ if p contains exactly one state of the form q_{00g} and one state of the form q_{01g}, and $h_\gamma(p) = \varnothing$ otherwise.

Clearly, only the look-ahead states $\{q_{00e}, q_{10e}\}$, $\{q_{01*}, q_{11*}\}$, $\{q_{00\gamma}, q_{10\#}\}$, and $\{q_{00\#}, q_{10\#}\}$ which we denote by e, $*$, γ, and $\#$, respectively, are needed.

Consider the input tree $s = \gamma\gamma{*}\gamma\gamma e$. The look-ahead automaton arrives for $s/1$ in state γ, for $s/11$ in $\#$, for $s/111$ in $*$, for $s/1^4$ and $s/1^5$ in γ, and for $s/1^6$ in e. The non-dummy γ-rules and the derivation for s by M are as follows.

$$\begin{array}{ll}
\langle q_{00\gamma}, \gamma(x_1)\rangle \to \gamma(\langle q_{00e}, x_1\rangle) & \langle e\rangle \\
\langle q_{00\gamma}, \gamma(x_1)\rangle \to \gamma(\langle q_{00\gamma}, x_1\rangle) & \langle \gamma\rangle \\
\langle q_{00\gamma}, \gamma(x_1)\rangle \to \gamma(\langle q_{00\#}, x_1\rangle) & \langle \#\rangle \\
\langle q_{00\#}, \gamma(x_1)\rangle \to \#(\langle q_{01*}, x_1\rangle) & \langle *\rangle \\
\langle q_{10\#}, \gamma(x_1)\rangle \to \#(\langle q_{00e}, x_1\rangle) & \langle e\rangle \\
\langle q_{10\#}, \gamma(x_1)\rangle \to \#(\langle q_{00\gamma}, x_1\rangle) & \langle \gamma\rangle \\
\langle q_{10\#}, \gamma(x_1)\rangle \to \#(\langle q_{00\#}, x_1\rangle) & \langle \#\rangle \\
\langle q_{10\#}, \gamma(x_1)\rangle \to \#(\langle q_{01*}, x_1\rangle) & \langle *\rangle
\end{array}$$

$$\begin{array}{rl}
\langle q_{\text{init}}, \gamma\gamma *\gamma\gamma e\rangle \Rightarrow_M & \gamma\langle q_{00\#}, \gamma * \gamma\gamma e\rangle \\
\Rightarrow_M & \gamma\#\langle q_{01*}, *\gamma\gamma e\rangle \\
\Rightarrow_M & \gamma\# * \langle q_{10\#}, \gamma\gamma e\rangle) \\
\Rightarrow_M & \gamma\# * \#\langle q_{00\gamma}, \gamma e\rangle \\
\Rightarrow_M & \gamma\# * \#\gamma\langle q_{00e}, e\rangle \\
\Rightarrow_M & \gamma\# * \#\gamma e
\end{array}$$

The unique state $q_0 \in h(s)$ for which $R_{\text{root}}(q_0)$ holds is $q_{00\gamma}$. Hence, $M_{q_{\text{init}}}(s) = M_{q_{00\gamma}}(s)$ for $s \in T_\Sigma$.

From Lemmas 4.1 and 4.2 the following theorem is obtained.

THEOREM 4.4. $ATT\text{-}REL = T^R\text{-}REL$.

In Theorem 10 of [BE98] it is proved that the class $ATT\text{-}REL$ of attributed relabelings is equal to the class $MSO\text{-}REL$ of MSO definable relabelings. By Proposition 2 of [BE98], MSO relabelings are closed under composition, and hence, $ATT\text{-}REL$ is closed under composition. Together with Theorem 4.4 this means that the class of top-down relabelings with regular look-ahead is closed under composition.

LEMMA 4.5. $T^R\text{-}REL \circ T^R\text{-}REL = T^R\text{-}REL$.

Note that in the framework of top-down tree transducers a proof of Lemma 4.5 would involve a straightforward product construction.

From Lemma 4.5 it follows in particular that $T^R\text{-}REL$ is closed under composition with $T\text{-}REL$. The class $B\text{-}REL$ of (total deterministic) bottom-up finite state relabelings is included in $T^R\text{-}REL$; cf. the discussion at the beginning of the subsection on attributed relabelings in Section 3. Hence, $T^R\text{-}REL$ is closed under composition with $B\text{-}REL$. Since, moreover, $T^R\text{-}REL = B\text{-}REL \circ T\text{-}REL$ (cf. Theorem 2.6 of [Eng77]), it follows that $ATT\text{-}REL$ is the composition closure of $B\text{-}REL$ and $T\text{-}REL$, i.e., of the (total deterministic) bottom-up and top-down (finite state) relabelings. This and its equivalence with the MSO relabelings, shows that it is a natural and robust class of relabelings.

5. SINGLE USE RESTRICTED TREE TRANSDUCERS

The main aim of this paper is to give a characterization in terms of MTTs of the class ATT^R_{sur} of translations realized by single use restricted (for short, sur) ATT$^{\text{R}}$s, which coincides with the class of MSO definable tree translations [BE98]. Such a characterization is given in this section by generalizing the sur property from ATTs to MTTs. As it turns out, using the straightforward extension of the sur property from ATTs to MTTs (called strongly single use restricted, or ssur) it is not possible to prove the equivalence between MTT^R_{ssur} and ATT^R_{sur}. In fact, MTT^R_{ssur} does not even contain all top-down relabelings (Theorem 5.6). Rather, using a slightly weaker restriction, which for MTTs might be the more natural notion of single use restriction, we prove that $MTT^R_{\text{sur}} = ATT^R_{\text{sur}}$. Indeed, the sur MTT$^{\text{R}}$s are equivalent to the composition $T^R\text{-}REL \circ MTT_{\text{ssur}}$, which allows us to use Theorem 4.4 for the proof of $MTT^R_{\text{sur}} = ATT^R_{\text{sur}}$, because $ATT^R_{\text{sur}} = ATT\text{-}REL \circ ATT_{\text{sur}}$ by definition.

5.1. *Single Use Restricted ATTs*

Consider for an ATT A the dependency graph $D(s)$ of an input tree s. This graph has the following properties. For an attribute c and a node u of s there is an edge from $\langle c, u\rangle$ to an attribute instance $\langle d, v\rangle$ in $D(s)$, if there is a derivation step

$\langle d, v\rangle \Rightarrow_{A,s} \xi$ such that $\langle c, u\rangle$ occurs in ξ. In other words, $\langle d, v\rangle$ depends on $\langle c, u\rangle$. There may be several attribute instances $\langle d, v\rangle$ that depend on $\langle c, u\rangle$, or none at all. In terms of the dependency graph this means that $D(s)$ is a *jungle*; that is, a forest with *sharing of subtrees* between the trees of the forest.

Let us now consider the special case that for every input tree s, the graph $D(s)$ is a forest; hence, no sharing occurs and, thus, the out-degree of every node is either zero or one. This can be ensured by allowing each outside attribute of an input symbol σ to be used *at most once* in the rules in R_σ. An ATT A with the latter property is called *single use restricted.* This property was introduced by Ganzinger [Gan83] as the "syntactic single used restriction." An interesting property of such ATTs is that the class of translations which can be realized by them is closed under composition [Gan83, Gie88, Küh97], whereas this is not the case for the class ATT (cf. Corollary 4.1 of [Fül81]).

DEFINITION 5.1 (Single use restricted). Let $A = (\mathrm{Syn}, \mathrm{Inh}, \Sigma, \Delta, \mathrm{root}, a_0, R)$ be an ATT. Then A is *single use restricted* (for short, sur), if for all $\sigma \in \Sigma^{(k)}_{\mathrm{root}}$, $\langle c, \rho\rangle$, $\langle c', \rho'\rangle \in \mathrm{ins}_\sigma$, $\zeta = \mathrm{rhs}(\langle c, \rho\rangle, \sigma)$, $\zeta' = \mathrm{rhs}(\langle c', \rho'\rangle, \sigma)$, $u \in \mathrm{Occ}(\zeta)$, and $u' \in \mathrm{Occ}(\zeta')$:

$$\text{if} \quad \zeta[u] = \zeta'[u'] \in \mathrm{outs}_\sigma, \qquad \text{then} \quad \langle c, \rho\rangle = \langle c', \rho'\rangle \quad \text{and} \quad u = u'.$$

The class of all translations which can be realized by sur ATTs is denoted by ATT_{sur}. The class $ATT\text{-}REL \circ ATT_{\mathrm{sur}}$ is denoted by ATT^R_{sur}. Indeed, ATT^R_{sur} is the class which will be proved to be equal to MTT^R_{sur}.

An obvious dynamic consequence of the static sur property is that if $\langle a_0, \varepsilon\rangle \Rightarrow^*_{A,\mathrm{root}(s)} \xi$, then each attribute instance $\langle c, u\rangle \in \langle \mathrm{Att}, \mathrm{Occ}(\mathrm{root}(s))\rangle$ occurs at most once in ξ. Intuitively this means that $\mathrm{nf}(\Rightarrow_{A,\mathrm{root}(s)}, \langle c, u\rangle)$ occurs at most once as a subtree in the output tree $\mathrm{nf}(\Rightarrow_{A,\mathrm{root}(s)}, \langle a_0, \varepsilon\rangle)$.

Let us investigate the is-dependencies of sur ATTs (cf. Definition 3.13). Let A be an ATT and let $f \subseteq \mathrm{Inh} \times \mathrm{Syn}$ be an is-dependency. It should be intuitively clear that if A is sur, then f is a partial function of type $\mathrm{Inh} \to \mathrm{Syn}$. This is proved in the next lemma. (Recall that $\mathrm{IS}(A) = \{\mathrm{is}(s) \mid s \in T_\Sigma\}$.)

LEMMA 5.2. Let A be a sur ATT. Then every element of $\mathrm{IS}(A)$ is a partial function.

Proof. Let $A = (\mathrm{Syn}, \mathrm{Inh}, \Sigma, \Delta, \mathrm{root}, a_0, R)$. We prove that $\mathrm{is}(s)$ is a partial function by induction on the structure of s. Let $s = \sigma(s_1, ..., s_k)$, $k \geqslant 0$, $\sigma \in \Sigma^{(k)}$, $s_1, ..., s_k \in T_\Sigma$, and $f = \mathrm{is}(s) = \mathrm{is}_\sigma(\mathrm{is}(s_1), ..., \mathrm{is}(s_k))$.

We assume that f is not a partial function and show that a contradiction follows. If f is not a partial function, then there are distinct $a, a' \in \mathrm{Syn}$, $b \in \mathrm{Inh}$ such that $(b, a), (b, a') \in f$. Thus, there are paths $w = \langle c_1, \rho_1\rangle \cdots \langle c_n, \rho_n\rangle$ and $w' = \langle c'_1, \rho'_1\rangle \cdots \langle c'_m, \rho'_m\rangle$ in $D_\sigma(\mathrm{is}(s_1), ..., \mathrm{is}(s_k))$ such that $\langle c_i, \rho_i\rangle, \langle c'_j, \rho'_j\rangle \in (\mathrm{ins}_\sigma \cup \mathrm{outs}_\sigma)$ for $i \in [n]$ and $j \in [m]$, $\langle c_1, \rho_1\rangle = \langle c'_1, \rho'_1\rangle = \langle b, \pi\rangle$, $\langle c_n, \rho_n\rangle = \langle a, \pi\rangle$, and $\langle c'_m, \rho'_m\rangle = \langle a', \pi\rangle$. Let $i \in [\min(m, n)]$ such that (i) for every $j < i$, $\langle c_j, \rho_j\rangle = \langle c'_j, \rho'_j\rangle$ and (ii) $\langle c_i, \rho_i\rangle \neq \langle c'_i, \rho'_i\rangle$. Thus, i is the smallest index such that the ith elements of w and w' are different. Such an index exists because the paths w and w' end in different attribute instances (of out-degree zero). If $\langle c_i, \rho_i\rangle \in \mathrm{ins}_\sigma$, then $\langle c_{i-1}, \rho_{i-1}\rangle$ occurs in both $\mathrm{rhs}(\langle c_i, \rho_i\rangle, \sigma)$ and $\mathrm{rhs}(\langle c_i', \rho_i'\rangle, \sigma)$ which contradicts the sur property of A; if $\langle c_i, \rho_i\rangle \in \mathrm{outs}_\sigma$, then a contradiction to the induction hypothesis follows (for $\mathrm{is}(s_v)$, where $\pi v = \rho_i$). ▮

Let A be a sur ATT and let g be a dependency graph of some $\sigma \in \Sigma^{(k)}$ with $d_1, ..., d_k \in \mathrm{IS}(A)$. If each node of g has out-degree zero or one, then we say that g is a forest (note that g is also acyclic). Every dependency graph g of A is a forest.

LEMMA 5.3. Let $A = (\mathrm{Syn}, \mathrm{Inh}, \Sigma, \Delta, \mathrm{root}, a_0, R)$ be a sur ATT. For every $\sigma \in \Sigma^{(k)}_{\mathrm{root}}$ and $d_1, ..., d_k \in \mathrm{IS}(A)$, $D_\sigma(d_1, ..., d_k)$ is a forest.

Proof. We assume that $D_\sigma(d_1, ..., d_k) = (V, E)$ is not a forest and show that a contradiction follows. If (V, E) is not a forest, then there is a node $\langle c, \pi i\rangle \in V$ with out-degree greater than or equal to two. Hence, there are $\langle c_1, \pi j_1\rangle, \langle c_2, \pi j_2\rangle \in V$ with $\langle c_1, \pi j_1\rangle \neq \langle c_2, \pi j_2\rangle$ such that $(\langle c, \pi i\rangle, \langle c_1, \pi j_1\rangle), (\langle c, \pi i\rangle, \langle c_2, \pi j_2\rangle) \in E$. If $\langle c, \pi i\rangle \in \mathrm{outs}_\sigma$ then $\langle c, \pi i\rangle$ occurs in both $\mathrm{rhs}(\langle c_1, \pi j_1\rangle, \sigma)$ and $\mathrm{rhs}(\langle c_2, \pi j_2\rangle, \sigma)$ which contradicts the sur property of A. Otherwise, $c \in \mathrm{Inh}$ and $i \in [k]$ and, hence, by the definition of dependency graph, both (c, c_1) and (c, c_2) are in d_i. Thus d_i is not a partial function which contradicts Lemma 5.2. ▮

5.2. Single Use Restricted MTTs

The aim of this subsection is to define a natural and static notion of single use restriction for MTTs such that the class MTT^R_{sur} coincides with the class ATT^R_{sur} (as proved in the next subsection). An MTT generates trees according to the states in which a subtree is processed and according to the parameters in which the context information is processed (just like recursive procedures with parameters). Roughly speaking the states of an MTT correspond to the synthesized attributes of an ATT and the parameters of an MTT correspond to the inherited attributes of an ATT. However, one reason why this comparison falls short is that each state of an MTT has its "own" set of parameters, whereas inherited attributes are associated with every symbol of the input tree. In this sense the single use restriction of using an (outside) inherited attribute at most once means that any parameter y_j may occur at most once in each right-hand side of a rule of an MTT (i.e., each right-hand side is linear or noncopying, with respect to the parameters). We call this property *single use restricted in the parameters.*

DEFINITION 5.4 (Single use restricted in the parameters). An $\mathrm{MTT}^{\mathrm{R}}$ $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ is *single use restricted in the parameters* (for short, surp), if for every $q \in Q^{(m)}$, $j \in [m]$, $\sigma \in \Sigma^{(k)}$, and $p_1, ..., p_k \in P$, y_j occurs at most once in $\mathrm{rhs}(q, \sigma, \langle p_1, ..., p_k\rangle)$.

The class of all translations which can be realized by surp $\mathrm{MTT}^{\mathrm{R}}$s is denoted by MTT^R_{surp}. Recall that, implicitly, this defines the class MTT_{surp} by the convention below Definition 3.3.

A rather obvious dynamic consequence of the surp property is that every $M_q(s) \in T_\Delta(Y)$ contains each parameter at most once.

It remains to find a restriction for MTTs which corresponds to the restriction on (outside) synthesized attributes of sur ATTs, i.e., a restriction on the states which appear in the right-hand sides of the rules of an MTT. For sur ATTs we wanted the dependency graphs to be forests, which could be ensured by allowing every outside attribute to be used at most once. In the case of an MTT M we can find a similar notion of dependency for the states of M (disregarding its parameters). In fact, it is well known and can easily be understood from Lemma 3.4 that an MTT can be viewed as an attribute grammar with the states as synthesized attributes (and no inherited attributes). The value of an attribute q at node u of s is the q-translation $M_q(s/u)$ and the only operation in the semantic rules is second-order substitution. We now consider the dependency graphs of this attribute grammar. More precisely, a state q depends on the states which appear in the right-hand sides of the q-rules. Thus, for every input symbol σ, q may depend on different states (and in the case of regular look-ahead for every tuple of look-ahead states). If, for an input tree s, we associate with every node in s the states of M, then we can define a notion of dependency graph similar to the one for ATTs. There is an edge from q at node ui to q' at node u, if $\langle q, x_i\rangle$ occurs in the (q', σ)-rule of M, where σ is the label of u. A natural way to ensure that each such dependency graph is a forest, is to require that, for each q and each x_i, there is at most one occurrence of $\langle q, x_i\rangle$ in the right-hand sides of all σ-rules (or, all $(\sigma, \langle p_1, ..., p_k\rangle)$-rules in the case of regular look-ahead). We call an MTT with the latter property *strongly single use restricted in the input* (for short, ssuri). This property was introduced in [Küh97], where it is called "single-used" (see also [Küh98]).

DEFINITION 5.5 (Strongly single use restricted in the input). Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\mathrm{MTT}^{\mathrm{R}}$ and let $\bar{Q}$ be a nonempty subset of Q. Then M is *strongly single use restricted in the input* (for short, ssuri) *with respect to* $\bar{Q}$, if for all $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, $p_1, ..., p_k \in P$, $q, q' \in \bar{Q}$, $\zeta = \mathrm{rhs}(q, \sigma, \langle p_1, ..., p_k\rangle)$, $\zeta' = \mathrm{rhs}(q', \sigma, \langle p_1, ..., p_k\rangle)$, $u \in \mathrm{Occ}(\zeta)$, and $u' \in \mathrm{Occ}(\zeta')$:

$$\text{if} \quad \zeta[u] = \zeta'[u'] \in \langle Q, X_k\rangle, \qquad \text{then} \quad q = q' \quad \text{and} \quad u = u'.$$

If M is ssuri with respect to Q, then M is called ssuri.

The class of all translations that can be realized by ssuri $\mathrm{MTT}^{\mathrm{R}}$s is denoted by MTT^R_{ssuri}. The class $MTT^R_{\mathrm{ssuri,\,surp}}$ will also be denoted by MTT^R_{ssur}. We note that the class of translations realized by the single-used MTTs of Kühnemann is denoted by MT_{su} in [Küh97]. There is a subtle difference between our class MTT_{ssur} and the class MT_{su} because Kühnemann uses a slightly different model of MTTs in which, just like for ATTs, input trees are of the form root(s) (cf. the discussion following Lemma 5.11).

As it turns out, with the above definition of ssuri, $MTT^R_{\mathrm{ssuri,\,surp}}$ does not equal ATT^R_{sur}. In fact, the restriction of being ssuri is so strong that not even every top-down relabeling can be realized by an MTT^R_{ssur}. This fact is proved in the next theorem (using Lemmas 6.6 and 6.7 which will be proved in the next section).

THEOREM 5.6. $T\text{-}REL \nsubseteq MTT^R_{\mathrm{ssur}}$.

Proof. Consider the T-REL $A=(Q_A, \Sigma, \Delta, q_{\rm in}, R_A)$ with $Q_A=\{q_{\rm in}, q\}$, $\Sigma=\Delta=\{a^{(1)}, b^{(1)}, e^{(0)}\}$, and $R_A=\{\langle q_0, a(x_1)\rangle \to a(\langle q_0, x_1\rangle), \langle q_0, b(x_1)\rangle \to b(\langle q, x_1\rangle), \langle q, a(x_1)\rangle \to b(\langle q, x_1\rangle), \langle q, b(x_1)\rangle \to b(\langle q, x_1\rangle), \langle q_0, e\rangle \to e, \langle q, e\rangle \to e\}$. Clearly, $\tau_A(a^n e)=a^n e$ and $\tau_A(a^n bwe)=a^n bb^{|w|}e$ with $n\geqslant 0$ and $w\in\{a,b\}^*$.

Assume now that there is an $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{ssur}}$ $M=(Q, P, \Sigma, \Delta, q_0, R, h)$ such that $\tau_M=\tau_A$. By Lemma 6.6 we may assume that M is nondeleting. This means (by Lemma 6.7) that for every $q\in Q^{(m)}$, $s\in T_\Sigma$, and $j\in[m]$, y_j appears in $M_q(s)$. Now consider the look-ahead automaton of M for input trees of the form $(ab)^l e$ with $l\geqslant 0$. Since P is finite there must be $m'>m\geqslant 0$ such that $p=h((ab)^{m'}e)=h((ab)^m e)$. Let $n=m'-m$ and $s=(ab)^m e$. Then $h((ab)^{kn} s)=p$ for every $k\geqslant 0$.

For all $i\geqslant 1$, $\xi_i=\hat{M}_{q_0}((ab)^{in}p)$ must contain at least one element of $\langle\!\langle Q, \{p\}\rangle\!\rangle$, where $\hat{M}$ is the extension of M (see Definition 3.5); this is because otherwise, by Lemma 3.6 for $u=1^{2in}$, $M_{q_0}((ab)^{(i+k)n} s)=\xi_i[\![\langle\!\langle q, p\rangle\!\rangle \leftarrow M_q((ab)^{kn} s) \mid q\in Q]\!]=\xi_i$ for every $k\geqslant 0$, which contradicts $\tau_M=\tau_A$. However, ξ_i cannot contain $\langle\!\langle q_0, p\rangle\!\rangle$. In fact, suppose that it does contain $\langle\!\langle q_0, p\rangle\!\rangle$. Then, by Lemma 3.6, $\tau_M((ab)^{in}s)=\hat{M}_{q_0}((ab)^{in}s)=\xi_i[\![\langle\!\langle q, p\rangle\!\rangle \leftarrow M_q(s) \mid q\in Q]\!]$. Since $M_{q_0}(s)=ab^{2m-1}e$, this equals $\xi'[\langle\!\langle q_0, p\rangle\!\rangle \leftarrow ab^{2m-1}e]$, where $\xi'=\xi_i[\![\langle\!\langle q, p\rangle\!\rangle \leftarrow M_q(s) \mid q\in Q-\{q_0\}]\!]$. By the nondeleting property of M, ξ' contains $\langle\!\langle q_0, p\rangle\!\rangle$, and so $\tau_M((ab)^{in}s)$ has a subtree $ab^{2m-1}e$. This contradicts the fact that $\tau_M((ab)^{in}s)=ab^{2(in+m)-1}e$, because $in\geqslant 1$.

Since Q is finite, there are $q\in Q$ and $i,k\geqslant 1$ such that $\langle\!\langle q, p\rangle\!\rangle$ occurs in both $\hat{M}_{q_0}((ab)^{in}p)$ and $\hat{M}_{q_0}((ab)^{(i+k)n}p)$. By Lemma 3.6, $\hat{M}_{q_0}((ab)^{(i+k)n}p)$ equals $\hat{M}_{q_0}((ab)^{kn}p)[\![\langle\!\langle q', p\rangle\!\rangle \leftarrow \hat{M}_{q'}((ab)^{in}p) \mid q'\in Q]\!]$. Hence, there is a $q'\in Q$ such that $\langle\!\langle q', p\rangle\!\rangle$ occurs in $\hat{M}_{q_0}((ab)^{kn}p)$ and $\langle\!\langle q, p\rangle\!\rangle$ occurs in $\hat{M}_{q'}((ab)^{in}p)$. We know that ξ_k does not contain $\langle\!\langle q_0, p\rangle\!\rangle$, and so $q'\neq q_0$. This contradicts the following claim (with $w=(ab)^{in}$, $q_1=q_0$, and $q_2=q'$).

Claim. Let $q_1, q_2\in Q$ and $w\in\{a,b\}^*$. If $\langle\!\langle q, p\rangle\!\rangle$ occurs in both $\hat{M}_{q_1}(wp)$ and $\hat{M}_{q_2}(wp)$, then $q_1=q_2$.

Proof. By induction on the length of w. If $w=\varepsilon$ then $\hat{M}_{q_1}(wp)=\langle\!\langle q_1, p\rangle\!\rangle$ and $\hat{M}_{q_2}(wp)=\langle\!\langle q_2, p\rangle\!\rangle$. Thus $q=q_1=q_2$.

If $w=cv$ with $c\in\{a,b\}$ and $v\in\{a,b\}^*$, then, by Lemma 3.4,

$$\hat{M}_{q_1}(cvp)=\mathrm{rhs}_M(q_1, c, \langle p'\rangle)[\![\langle r, x_1\rangle \leftarrow \hat{M}_r(vp) \mid r\in Q]\!] \text{ and}$$

$$\hat{M}_{q_2}(cvp)=\mathrm{rhs}_M(q_2, c, \langle p'\rangle)[\![\langle r, x_1\rangle \leftarrow \hat{M}_r(vp) \mid r\in Q]\!], \text{ where } p'=\hat{h}(vp).$$

Hence, for $\nu\in[2]$, there is a state $r_\nu\in Q$ such that $\langle r_\nu, x_1\rangle$ occurs in $\mathrm{rhs}_M(q_\nu, c, \langle p'\rangle)$ and $\langle\!\langle q, p\rangle\!\rangle$ occurs in $\hat{M}_{r_\nu}(vp)$. By induction, $r_1=r_2$. By the definition of ssuri this means that $q_1=q_2$. ∎

We will prove that $T^R\text{-}REL \circ MTT_{\rm ssur}=ATT^R_{\rm sur}$. The T^{R}-REL can be incorporated into the MTT, if we allow a slightly weaker restriction than ssuri, called suri. For this restriction, which we will discuss now, we will prove that $MTT^R_{\rm suri,\,surp}=ATT^R_{\rm sur}$.

In the notion of dependency as described above, since we associate *all* states of M with each node of s, and in particular with the root of s, a dependency graph of a ssuri MTT is, in general, a forest rather than a tree. However, only the tree that contains the initial state q_0 (at the root of s) is involved in the computation of the output tree. Again we want to find a natural restriction on the rules of M such that the "initial dependency graph," i.e., the dependency graph restricted to the states that are connected to q_0 (at the root of s), is a tree. Consider three states q_1, q_2, q_3 such that $\langle q, x_i \rangle$ occurs in the right-hand side of each (q_m, σ)-rule for $m \in [3]$. For the initial dependency graph to be a tree, none of the q_1, q_2, q_3 should occur together in it, at the same node with label σ. We can try to partition the set of states Q of M into sets $Q_1, ..., Q_n$ of states which *may* occur together in this way. Then we have to make sure that at any node u the states occurring in the initial dependency graph at u are all in one particular Q_j. This can be done by requiring that for all right-hand sides of (q, σ)-rules with $q \in Q_j$, the set of states q' such that $\langle q', x_i \rangle$ occurs in them, is again contained in one particular Q_l. We call an MTT for which such a partition $\Pi = \{Q_1, ..., Q_n\}$ exists *single use restricted in the input.*

DEFINITION 5.7 (Single use restricted in the input). Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\mathrm{MTT}^{\mathrm{R}}$. Then M is *single use restricted in the input* (for short, suri), if there is a partition Π of Q and a collection of mappings $\mathscr{T} = (\mathscr{T}_{\sigma, \langle p_1, ..., p_k \rangle} : \Pi \times [k] \to \Pi \mid \sigma \in \Sigma^{(k)}, p_1, ..., p_k \in P)$ such that

(i) for every $\bar{Q} \in \Pi$, M is ssuri with respect to $\bar{Q}$ and

(ii) for all $\sigma \in \Sigma^{(k)}$, $p_1, ..., p_k \in P$, $\bar{Q} \in \Pi$, $i \in [k]$, $q \in \bar{Q}$, $\zeta = \mathrm{rhs}(q, \sigma, \langle p_1, ..., p_k \rangle)$, $u \in \mathrm{Occ}(\zeta)$, and $r \in Q$:

$$\text{if} \quad \zeta[u] = \langle r, x_i \rangle, \qquad \text{then} \quad r \in \mathscr{T}_{\sigma, \langle p_1, ..., p_k \rangle}(\bar{Q}, i).$$

The partition Π is called a *sur partition for M* and $\mathscr{T}$ is called a *collection of sur mappings for M.*

The class of all translations which can be realized by suri $\mathrm{MTT}^{\mathrm{R}}$s is denoted by MTT^{R}_{suri}. Altogether we say that an $\mathrm{MTT}^{\mathrm{R}}$ is *single use restricted* (for short, sur), if it is both suri and surp. We also denote the class $MTT^{R}_{\mathrm{suri,\,surp}}$ by MTT^{R}_{sur}.

A dynamic consequence of the sur property (both suri and surp) is that every state sequence $\mathrm{sts}_M(s, u)$ of M (cf. Definition 3.7) contains each state at most once (cf. the claim in the proof of Theorem 6.12). Intuitively this means, by Lemma 3.6 and the remark below Definition 5.4, that each $M_q(s/u)$ occurs at most once as a part of the output tree $M_{q_0}(s)$.

Note also that M is suri with sur partition $\Pi = \{\{q\} \mid q \in Q\}$ if and only if M is linear in the input (i.e., no right-hand side of a rule contains two occurrences of the same input variable x_i).

We now want to investigate how ssuri MTTs are related to suri MTTs. Clearly, every ssuri MTT M is also suri; just take as a sur partition for M the singleton consisting of the set of states of M. As it turns out, suri MTTs are just top-down relabelings with regular look-ahead, followed by ssuri MTTs. We will only need this result for surp MTTs.

LEMMA 5.8. $T^{R}\text{-}REL \circ MTT_{\mathrm{ssuri,\,surp}} \subseteq MTT^{R}_{\mathrm{suri,\,surp}}$.

Proof. Let $A = (Q_A, P, \Sigma, \Gamma, q_A, R_A, h)$ be a T^{R}-REL and let $M = (Q, \Gamma, \Delta, q_0, R)$ be an $\mathrm{MTT}_{\mathrm{ssuri,\,surp}}$. We now define, by a straightforward product construction, an $\mathrm{MTT}^{\mathrm{R}}_{\text{suri, surp}}$ M' which realizes the translation $\tau_A \circ \tau_M$. Let $M' = (Q', P, \Sigma, \Delta, (q_0, q_A), R', h)$ with

- $Q' = Q \times Q_A$ and for every $(q, r) \in Q'$, $\mathrm{rank}_{Q'}((q, r)) = \mathrm{rank}_Q(q)$,
- and R' constructed as follows: Let $q \in Q^{(m)}$, $r \in Q_A$, $\sigma \in \Sigma^{(k)}$, $p_1, ..., p_k \in P$, and let $\mathrm{rhs}_A(r, \sigma, \langle p_1, ..., p_k\rangle) = \delta(\langle r_1, x_1\rangle, ..., \langle r_k, x_k\rangle)$ with $\delta \in \Gamma^{(k)}$ and $r_1, ..., r_k \in Q_A$. Then, let the rule

$$\langle (q, r), \sigma(x_1, ..., x_k)\rangle (y_1, ..., y_m) \to \zeta \qquad \langle p_1, ..., p_k\rangle$$

be in R', where ζ is obtained from $\mathrm{rhs}_M(q, \delta)$ by replacing every occurrence of $\langle \bar{q}, x_i\rangle$ by $\langle (\bar{q}, r_i), x_i\rangle$, where $\bar{q} \in Q$ and $i \in [k]$.

Obviously M' is surp, because M is surp and the right-hand sides of rules in M' are obtained from those of M by a renaming of states only. Furthermore, M' is suri. In fact, let $Q_r = \{(q, r) \mid q \in Q\}$ for $r \in Q_A$, $\Pi = \{Q_r \mid r \in Q_A\}$, and $\mathscr{T}_{\sigma, \langle p_1, ..., p_k\rangle}(Q_r, i) = Q_{\bar{r}}$ with $\langle \bar{r}, x_i\rangle = \mathrm{rhs}_A(r, \sigma, \langle p_1, ..., p_k\rangle)[i]$. Then Π is a sur partition for M' and $(\mathscr{T}_{\sigma, \langle p_1, ..., p_k\rangle} \mid \sigma \in \Sigma^{(k)}, p_1, ..., p_k \in P)$ is a collection of sur mappings for M'. This is shown as follows.

Let $r \in Q_A$, $\sigma \in \Sigma^{(k)}$, $p \in P^k$, and $i \in [k]$. In the (r, σ, p)-rule of A there is an occurrence of $\langle r_i, x_i\rangle$. Also, there is one particular output symbol $\delta \in \Gamma$ at the root of $\mathrm{rhs}_A(r, \sigma, p)$. If we consider the right-hand sides of the δ-rules of M, then, since M is ssuri, there are no occurrences of $\langle \bar{q}, x_i\rangle$ and $\langle \bar{q}', x_i\rangle$ such that $\bar{q} \neq \bar{q}'$. If we now consider the right-hand sides of the $((q, r), \sigma, p)$-rules of M' for different $q \in Q$, then for x_i there is at most one occurrence of $\langle (\bar{q}, r_i), x_i\rangle$ with $\bar{q} \in Q$. Thus, M' is ssuri with respect to Q_r for $r \in Q_A$, and r_i determines the value of the sur mapping $\mathscr{T}_{\sigma, p}$ for (Q_r, i), viz., Q_{r_i}.

The correctness of the construction can be shown by proving that for every $(q, r) \in Q'$ and $s \in T_\Sigma$, $M'_{(q, r)}(s) = M_q(A_r(s))$; this can easily be proved by induction on the structure of s using Lemma 3.4. ∎

LEMMA 5.9. $MTT^R_{\text{suri, surp}} \subseteq T^R\text{-}REL \circ MTT_{\text{ssuri, surp}}$.

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\mathrm{MTT}^{\mathrm{R}}_{\text{suri, surp}}$, let Π be a sur partition for M, and let $\mathscr{T}$ be a collection of sur mappings for M. Let us now define a T^{R}-REL A and an $\mathrm{MTT}_{\text{ssuri, surp}}$ M' such that $\tau_A \circ \tau_{M'} = \tau_M$.

Let $A = (\Pi, P, \Sigma, \Gamma, Q_0, R_A, h)$ with

- $Q_0 \in \Pi$ such that $q_0 \in Q_0$,
- $\Gamma = \{(\sigma, \bar{Q}, \langle p_1, ..., p_k\rangle)^{(k)} \mid \sigma \in \Sigma^{(k)}, \bar{Q} \in \Pi, p_1, ..., p_k \in P\}$,
- for every $\sigma \in \Sigma^{(k)}$, $\bar{Q} \in \Pi$, and $p_1, ..., p_k \in P$ the rule

$$\langle \bar{Q}, \sigma(x_1, ..., x_k)\rangle \to (\sigma, \bar{Q}, \langle p_1, ..., p_k\rangle)(\langle Q_1, x_1\rangle, ..., \langle Q_k, x_k\rangle) \qquad \langle p_1, ..., p_k\rangle$$

is in R_A, where $Q_i = \mathscr{T}_{\sigma, \langle p_1, ..., p_k\rangle}(\bar{Q}, i)$ for $i \in [k]$.

Let $M' = (Q, \Gamma, \Delta, q_0, R')$ such that for every $q \in Q^{(m)}$, $\sigma \in \Sigma^{(k)}$, $\bar{Q} \in \Pi$, and $p_1, ..., p_k \in P$ the rule

$$\langle q, (\sigma, \bar{Q}, \langle p_1, ..., p_k \rangle) \rangle (y_1, ..., y_m) \to \zeta$$

is in R', where $\zeta = \text{rhs}_M(q, \sigma, \langle p_1, ..., p_k \rangle)$ if $q \in \bar{Q}$ and, otherwise, $\zeta = \text{dummy}$, where dummy is an arbitrary symbol in $\Delta^{(0)}$.

Let s be a tree in T_Σ and let u be a node in s labeled by σ. The top-down relabeling A replaces σ by $(\sigma, \bar{Q}, \langle p_1, ..., p_k \rangle)$, where $\langle p_1, ..., p_k \rangle$ are the look-ahead states at the children of u, and $\bar{Q} \in \Pi$ is determined by $\mathscr{T}$ in an obvious top-down fashion. We observe here that it can be shown that $\bar{Q}$ contains all states of M that appear in its state sequence $\text{sts}_M(s, u)$ (cf. Definition 3.7 and the claim in the proof of Theorem 6.12). This is the intuition behind the requirement $q \in \bar{Q}$ in the definition of the rules of M'.

The rules of M' have the same right-hand sides as the rules of M (plus dummies); thus, M' is surp. Instead of using the look-ahead automaton, M' obtains the look-ahead information from the input symbol. It is ssuri because the set of right-hand sides of $(q, (\sigma, \bar{Q}, \langle p_1, ..., p_k \rangle))$-rules of M' consists of dummies and of right-hand sides of $(q, \sigma, \langle p_1, ..., p_k \rangle)$-rules of M with $q \in \bar{Q}$ which are ssuri by the definition of suri.

The correctness of the construction can be shown by proving that $M'_q(A_{\bar{Q}}(s)) = M_q(s)$ for all $\bar{Q} \in \Pi$, $s \in T_\Sigma$, and $q \in \bar{Q}$. As in the previous lemma, the proof is straightforward by induction on the structure of s, using Lemma 3.4. ∎

Note that in the proof we did not use the fact that Π is a partition; it might be any subset of $\mathscr{P}(Q)$ such that $q_0 \in \bigcup \Pi$.

By Lemmas 5.8 and 5.9 we obtain the following theorem.

THEOREM 5.10. $MTT^R_{\text{sur}} = T^R\text{-}REL \circ MTT_{\text{ssur}}$.

In the constructions in the proofs of Lemmas 5.8 and 5.9, the parameters of the involved macro tree transducers are not taken into account. Therefore, it is easy to see that corresponding results hold for top-down tree transducers, i.e. $T^R_{\text{sur}} = T^R\text{-}REL \circ T_{\text{ssur}}$.

5.3. *Comparison of Single Use Restricted ATTs and MTTs*

It is well known that every ATT A can be turned into an MTT M such that M and A realize the same translation. The states of M correspond to the synthesized attributes of A and the parameters of M correspond to the inherited attributes of A. However, every state of M has a fixed number of parameters, whereas a synthesized attribute may depend on any number of inherited attributes (depending on the input subtrees). For a particular subclass of ATT, called *absolutely noncircular* [KW76], the set of inherited attributes on which a synthesized attribute depends is fixed for every input symbol. Then, M can be constructed straightforwardly [CF82]. For a general ATT A an MTT M can be constructed by assuming "worst case" dependencies, i.e., that each synthesized attribute depends on all inherited attributes [Fra82, FV97, FV98] (technically this means that each state of M is of rank $|\mathrm{Inh}|$). Clearly, a rule of M will delete parameters when it corresponds to a synthesized attribute which actually does not depend on all inherited attributes.

If we add regular look-ahead to M, then the situation is different. The information we need, i.e., for an input tree s the set of inherited attributes that each synthesized attribute depends on at the root of s, is precisely the is-dependency of s (Definition 3.13). The is-dependencies can be determined by regular look-ahead. Thus, for every ATT A an $\mathrm{MTT}^{\mathrm{R}}$ M can be constructed such that $\tau_M = \tau_A$, in exactly the same way as for the absolutely noncircular case. This result has already been mentioned in [Eng80, Eng81]. As the one of [CF82], our construction has the additional property that if A is sur, then M is sur (i.e., suri and surp).

LEMMA 5.11. $ATT \subseteq MTT^R$ and $ATT_{\mathrm{sur}} \subseteq MTT^R_{\mathrm{sur}}$.

Proof. Let $A = (\mathrm{Syn}, \mathrm{Inh}, \Sigma, \Delta, \mathrm{root}, a_0, R)$ be an ATT with $\mathrm{Inh} = \{b_1, ..., b_N\}$ and let $b_1, b_2, ..., b_N$ be an arbitrary but fixed order on Inh.

It suffices to construct a (strongly sur) $\mathrm{MTT}^{\mathrm{R}}$ M with $\tau_M(\mathrm{root}(s)) = \tau_A(s)$ for all $s \in T_\Sigma$. This is due to the fact that if $\tau' = \{(s, t) \mid (\mathrm{root}(s), t) \in \tau\}$ for some $\tau \in MTT^R$, then $\tau' \in MTT^R$ and if $\tau \in MTT^R_{\mathrm{ssur}}$, then $\tau' \in MTT^R_{\mathrm{sur}}$. This can be shown as follows. Let $M = (Q, P, \Sigma_{\mathrm{root}}, \Delta, q_0, R_M, h)$ be an $\mathrm{MTT}^{\mathrm{R}}$ and define $M' = (Q \cup \{q^{(0)}_{\mathrm{root}}\}, P, \Sigma, \Delta, q_{\mathrm{root}}, R', h)$, where R' consists of all non-root rules in R_M and for $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $p_1, ..., p_k \in P$ of the rule $\langle q_{\mathrm{root}}, \sigma(x_1, ..., x_k)\rangle \to \zeta \; \langle p_1, ..., p_k\rangle$, where $\zeta = \mathrm{rhs}_M(q_0, \mathrm{root}, \langle p\rangle)[\![\langle q, x_1\rangle \leftarrow \mathrm{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle) \mid q \in Q]\!]$, with $p = h_\sigma(p_1, ..., p_k)$.

Intuitively, a $(q_{\mathrm{root}}, \sigma)$-rule of M' incorporates both the (q_0, root)-rule and the (q, σ)-rules of M. Obviously $M'_q(s) = M_q(s)$ for every $q \in Q$ and $s \in T_\Sigma$. Using this, Lemma 3.4, and the associativity of second-order substitution, it is straightforward to show that for every $s \in T_\Sigma$, $M'_{q_{\mathrm{root}}}(s) = M_{q_0}(\mathrm{root}(s))$. If M is ssur, then M' is sur with sur partition $\Pi = \{\{q_{\mathrm{root}}\}, Q\}$. Clearly M' is ssuri with respect to Q; M' is ssuri with respect to $\{q_{\mathrm{root}}\}$ because, since M is ssur, no element of $\langle Q, X_k\rangle$ occurs more than once in the right-hand sides $\mathrm{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$ for $q \in Q$. The same is true for $\mathrm{rhs}_M(q_0, \mathrm{root}, \langle p\rangle)$, and hence, also for $\mathrm{rhs}_{M'}(q_{\mathrm{root}}, \sigma, \langle p_1, ..., p_k\rangle)$.

Let us now construct an $\mathrm{MTT}^{\mathrm{R}}$ (or $\mathrm{MTT}^{R}_{\mathrm{ssur}}$) M with $\tau_M(\mathrm{root}(s)) = \tau_A(s)$ for all $s \in T_\Sigma$. The states of M correspond to the synthesized attributes of A and the parameters of M correspond to the inherited attributes of A. Each state gets as parameters only those inherited attributes it depends on (in the order fixed above). Since this depends on the subtree, we need states of the form (a, I), where $a \in \mathrm{Syn}$ and I is the set of inherited attributes that a depends on. The look-ahead automaton is used to determine for an input tree s and every $a \in \mathrm{Syn}$ the correct set I, i.e., to determine the is-dependency $\mathrm{is}(s)$ of s: $I = (\mathrm{is}(s))^{-1}(a)$.

Let $M = (Q, P, \Sigma_{\mathrm{root}}, \Delta, q_0, R', h)$, where

- $Q = \mathrm{Syn} \times \mathscr{P}(\mathrm{Inh})$ with $\mathrm{rank}_Q((a, I)) = |I|$ for $(a, I) \in Q$.
- $q_0 = (a_0, \varnothing)$.
- $P = \mathrm{IS}(A)$.
- For every $(a, I) \in Q^{(m)}$, $\sigma \in \Sigma^{(k)}_{\mathrm{root}}$, and $p_1, ..., p_k \in P$ with $m, k \geqslant 0$, let the rule

$$\langle (a, I), \sigma(x_1, ..., x_k)\rangle(y_1, ..., y_m) \to \zeta \qquad \langle p_1, ..., p_k\rangle$$

be in R', where $\zeta = \text{dummy} \in \Delta^{(0)}$ if $\sigma = \text{root}$ and $(a, I) \neq (a_0, \varnothing)$ or if $(\text{is}_\sigma(p_1, ..., p_k))^{-1}(a) \neq I$; otherwise $\zeta = \text{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle a, \pi \rangle)$. For every $\langle c, \rho \rangle \in \text{ins}_\sigma$, $\text{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle c, \rho \rangle)$ is recursively defined to be obtained from $\text{rhs}_A(\langle c, \rho \rangle, \sigma)$ by the following replacements. Let $I = \{b_{v_1}, ..., b_{v_m}\}$ with $v_1 < v_2 < \cdots < v_m$.

(R1) Replace every occurrence of $\langle b, \pi \rangle$ by y_j, if $b = b_{v_j}$ and $j \in [m]$; otherwise replace it by an arbitrary $\text{dummy} \in \Delta^{(0)}$.

(R2) Replace every occurrence of $\langle a', \pi i \rangle$ with $a' \in \text{Syn}$ and $i \in [k]$ by

$$\langle (a', I'), x_i \rangle (\text{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle b_{\eta_1}, \pi i \rangle), ..., \text{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle b_{\eta_r}, \pi i \rangle)),$$

where $I' = \{b_{\eta_1}, ..., b_{\eta_r}\} = p_i^{-1}(a')$ with $\eta_1 < \cdots < \eta_r$.

- The look-ahead automaton of M is defined as follows. For $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $p_1, ..., p_k \in P$, let $h_\sigma(p_1, ..., p_k) = \text{is}_\sigma(p_1, ..., p_k)$ and, for $p \in P$, let $h_{\text{root}}(p) = \text{dummy} \in P$.

Consider the dependency graph $g = D_\sigma(p_1, ..., p_k)$. Since A is noncircular, g is acyclic. For $\langle c, \rho \rangle \in \text{ins}_\sigma$, the recursive definition of trans follows the paths in g which lead to $\langle c, \rho \rangle$, going backwards. More precisely, if there is a path from $\langle b, \pi i \rangle$ to $\langle c, \rho \rangle$ with $b \in \text{Inh}$ and $i \in [k]$, then the call of trans for $\langle c, \rho \rangle$ recursively calls trans on $\langle b, \pi i \rangle$. The recursion of trans terminates, because g is acyclic.

We now prove the correctness of the construction, i.e., that $\tau_M(\text{root}(s_0)) = \tau_A(s_0)$ for every $s_0 \in T_\Sigma$. This follows from Claim 1(b), with $s = \text{root}(s_0)$ and $(a, I) = (a_0, \varnothing)$.

Claim 1. Let $\sigma \in \Sigma^{(k)}_{\text{root}}$, $k \geqslant 0$, $s_1, ..., s_k \in T_\Sigma$, and $s = \sigma(s_1, ..., s_k)$.

(a) If $\sigma \in \Sigma$ then $h(s) = \text{is}(s)$.

(b) For $a \in \text{Syn}$ (with $a = a_0$ for $\sigma = \text{root}$), $M_{(a, I)}(s)\, \Theta = \text{nf}(\Rightarrow_{A, s}, \langle a, \varepsilon \rangle)$, where $I = \{b_{v_1}, ..., b_{v_m}\} = (\text{is}(s))^{-1}(a)$ with $v_1 < \cdots < v_m$ and Θ denotes the substitution $[y_j \leftarrow \langle b_{v_j}, \varepsilon \rangle \mid j \in [m]]$.

Proof. By induction on the structure of s. Let the induction hypothesis be denoted by IH1. For (a), $h(s) = h_\sigma(h(s_1), ..., h(s_k))$, which, by the definition of h_σ, is equal to $\text{is}_\sigma(h(s_1), ..., h(s_k))$ and by IH1(a) equal to $\text{is}_\sigma(\text{is}(s_1), ..., \text{is}(s_k)) = \text{is}(s)$. In what follows let, for $i \in [k]$, $p_i = h(s_i)$ which, by Claim 1(a), equals $\text{is}(s_i)$. To prove (b), consider Claim 2 which concerns all inside attributes of σ. By the definition of the rules of M, $\text{rhs}_M((a, I), \sigma, \langle p_1, ..., p_k \rangle) = \text{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle a, \pi \rangle)$ and hence, by Lemma 3.4, Claim 1(b) follows from Claim 2 by taking $c = a$ and $l = 0$. Note that $\text{nf}(\Rightarrow_{A, s}, \langle a, 0 \rangle)$ contains no $\langle b, \varepsilon \rangle$ with $b \in \text{Inh} - I$, because $I = (\text{is}(s))^{-1}(a)$ and thus $b \in I$ if and only if $(b, a) \in \text{is}(s)$ if and only if $\text{nf}(\Rightarrow_{A, s}, \langle a, 0 \rangle)$ contains $\langle b, \varepsilon \rangle$, by Lemma 3.14(i).

Claim 2. For every $\langle c, \pi l \rangle \in \text{ins}_\sigma$ such that $\text{nf}(\Rightarrow_{A, s}, \langle c, l \rangle)$ contains no $\langle b, \varepsilon \rangle$ with $b \in \text{Inh} - I$,

$$\text{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle c, \pi l \rangle) [\![\cdots]\!]\, \Theta = \text{nf}(\Rightarrow_{A, s}, \langle c, l \rangle),$$

where $[\![\cdots]\!]$ denotes the substitution $[\![\langle(a', I'), x_i\rangle \leftarrow M_{(a', I')}(s_i) \mid \langle(a', I'), x_i\rangle \in \langle Q, X_k\rangle]\!]$.

Proof. By induction on the recursive definition of trans. Let $\xi = \text{rhs}_A(\langle c, \pi l\rangle, \sigma)$. Then, by the definition of trans, $\text{trans}_{\sigma, I, \langle p_1, ..., p_k\rangle}(\langle c, \pi l\rangle)[\![\cdots]\!]\,\Theta = \xi\Theta_1\Theta_2[\![\cdots]\!]\,\Theta$, where Θ_1 and Θ_2 are the substitutions corresponding to the replacements of the inherited and synthesized attributes in the definition of trans, respectively (see (R1), (R2)). Note that Θ_1 does not introduce dummies because, by assumption, ξ contains no $\langle b, \pi\rangle$ with $b \in \text{Inh} - I$.

Now Θ_2 and $[\![\cdots]\!]$ can be combined, because Θ_2 introduces states of the form $\langle(a', I'), x_i\rangle$ which are replaced by $[\![\cdots]\!]$. We get

$$\xi\Theta_1\Theta[\langle a', \pi i\rangle \leftarrow M_{(a', I')}(s_i)\,\Theta' \mid a' \in \text{Syn}, i \in [k], I' = p_i^{-1}(a') = \{b_{\eta_1}, ..., b_{\eta_r}\}],$$

where Θ' is the substitution $[\,y_j \leftarrow \text{trans}_{\sigma, I, \langle p_1, ..., p_k\rangle}(\langle b_{\eta_j}, \pi i\rangle)[\![\cdots]\!]\,\Theta \mid j \in [r]]$. We can now apply IH1(b) to get

$$\begin{aligned}\xi\Theta_1\Theta[\langle a', \pi i\rangle \leftarrow \text{nf}(\Rightarrow_{A, s_i}, \langle a', \varepsilon\rangle)\,\Theta'' \mid a' \in \text{Syn}, i \in [k],&\\ I' = p_i^{-1}(a') = \{b_{\eta_1}, ..., b_{\eta_r}\}],&\end{aligned}$$

where Θ'' is equal to

$$[\langle b, \varepsilon\rangle \leftarrow \text{trans}_{\sigma, I, \langle p_1, ..., p_k\rangle}(\langle b, \pi i\rangle)[\![\cdots]\!]\,\Theta \mid b \in I'].$$

Since $\langle a', \pi i\rangle$ occurs in ξ, there is an edge from $\langle a', \pi i\rangle$ to $\langle c, \pi l\rangle$ in $g = D_\sigma(p_1, ..., p_k)$. Since $b \in I' = p_i^{-1}(a')$, there is an edge from $\langle b, \pi i\rangle$ to $\langle a', \pi i\rangle$ in g. Thus, if there is a path from $\langle b', \pi\rangle$ to $\langle b, \pi i\rangle$ in g, then there is also a path from $\langle b', \pi\rangle$ to $\langle c, \pi l\rangle$ in g. Hence, by Lemma 3.14(ii), if $\langle b', \varepsilon\rangle$ occurs in $\text{nf}(\Rightarrow_{A, s}, \langle b, i\rangle)$ then it also occurs in $\text{nf}(\Rightarrow_{A, s}, \langle c, l\rangle)$. Hence, $\text{nf}(\Rightarrow_{A, s}, \langle b, i\rangle)$ with $b \in I'$ does not contain occurrences of $\langle b', \varepsilon\rangle$ with $b' \in \text{Inh} - I$ and we can apply IH2 to $\langle b, \pi i\rangle$. Thus, $\Theta'' = [\langle b, \varepsilon\rangle \leftarrow \text{nf}(\Rightarrow_{A, s}, \langle b, i\rangle) \mid b \in I']$. Again by Lemma 3.14(ii), $\text{nf}(\Rightarrow_{A, s_i}, \langle a', \varepsilon\rangle)$ does not contain occurrences of $\langle b, \varepsilon\rangle$ with $b \in \text{Inh} - I'$. Therefore, we can extend Θ'' to replace all $\langle b, \varepsilon\rangle$ with $b \in \text{Inh}$. The same holds for the substitution $\Theta_1\Theta = [\langle b, \pi\rangle \leftarrow \langle b, \varepsilon\rangle | b \in I]$. We get

$$\begin{aligned}&\xi[\langle b, \pi\rangle \leftarrow \langle b, \varepsilon\rangle \mid b \in \text{Inh}]\\ &\quad[\langle a', \pi i\rangle \leftarrow \text{nf}(\Rightarrow_{A, s_i}, \langle a', \varepsilon\rangle)[\langle b, \varepsilon\rangle \leftarrow \text{nf}(\Rightarrow_{A, s}, \langle b, i\rangle) \mid b \in \text{Inh}] \mid a' \in \text{Syn}, i \in [k]].\end{aligned}$$

By Lemma 3.12 this is equal to $\text{nf}(\Rightarrow_{A, s}, \langle c, l\rangle)$ which finishes the proof of Claim 2.

Assume now that A is sur. We need to show that M is ssur, i.e., both ssuri and surp. Intuitively this is because the recursion of trans follows the paths in $D_\sigma(p_1, ..., p_k)$ and $D_\sigma(p_1, ..., p_k)$ is a forest (see Lemma 5.3). Formally, we first prove the following claim by induction on the recursive definition of trans and then show that ssuri of M follows.

Let $\sigma \in \Sigma_{\mathrm{root}}^{(k)}$, $k \geqslant 0$, $p_1, ..., p_k \in P$, $I = \{b_{v_1}, ..., b_{v_m}\}$ with $v_1 < \cdots < v_m$, $\langle c, \rho \rangle \in \mathrm{ins}_\sigma$, and $\langle (a, I'), x_i \rangle \in \langle Q, X_k \rangle$.

Claim 3. (a) If $\langle (a, I'), x_i \rangle$ occurs in $\mathrm{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle c, \rho \rangle)$, then there is a path from $\langle a, \pi i \rangle$ to $\langle c, \rho \rangle$ in $D_\sigma(p_1, ..., p_k)$.

(b) $\langle (a, I'), x_i \rangle$ occurs in $\mathrm{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle c, \rho \rangle)$ at most once.

Proof. In the following let $\zeta = \mathrm{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle c, \rho \rangle)$, $\xi = \mathrm{rhs}_A(\langle c, \rho \rangle, \sigma)$, $\mathrm{Occ}_{a,i} = \{w \in \mathrm{Occ}(\xi) \mid \xi[w] = \langle a, \pi i \rangle\}$, and $g = D_\sigma(p_1, ..., p_k) = (V, E)$. Let $u, v \in \mathrm{Occ}(\zeta)$ with $\zeta[u] = \zeta[v] = \langle (a, I'), x_i \rangle$. Let us denote the induction hypothesis by IH3.

Case (i). $u \in \mathrm{Occ}(\zeta) - \mathrm{Occ}_{a,i}$, i.e., $\langle (a, I'), x_i \rangle$ occurs in a recursive call $\mathrm{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle} \times (\langle b, \pi j \rangle)$, where $b \in p_j^{-1}(a')$ and $\langle a', \pi j \rangle$ occurs in ξ. By IH3(a) there is a path w_0 from $\langle a, \pi i \rangle$ to $\langle b, \pi j \rangle$ in g. Since $(b, a') \in p_j$ and $\langle a', \pi j \rangle$ occurs in ξ, there are edges $(\langle b, \pi j \rangle, \langle a', \pi j \rangle)$ and $(\langle a', \pi j \rangle, \langle c, \rho \rangle)$ in E, respectively, and hence, there is a path $w = w_0 \langle a', \pi j \rangle \langle c, \rho \rangle$ from $\langle a, \pi i \rangle$ to $\langle c, \rho \rangle$ in g, which proves the (a) part of Claim 3. Let us now prove part (b) by showing that $u = v$:

- $v \in \mathrm{Occ}(\zeta) - \mathrm{Occ}_{a,i}$. Let v occur in $\mathrm{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle b', \pi j' \rangle)$, where $b' \in p_{j'}^{-1}(a'')$ and $\langle a'', \pi j' \rangle$ occurs in ξ. As for u, there is a path w'_0 from $\langle a, \pi i \rangle$ to $\langle b', \pi j' \rangle$ in g, and hence, a path $w' = w'_0 \langle a'', \pi j' \rangle \langle c, \rho \rangle$ from $\langle a, \pi i \rangle$ to $\langle c, \rho \rangle$ in g. Since w and w' are both from $\langle a, \pi i \rangle$ to $\langle c, \rho \rangle$ and since g is a forest by Lemma 5.3, they must be the same, and hence in particular $\langle b, \pi j \rangle = \langle b', \pi j' \rangle$. Thus, since both (b, a') and (b, a'') are in p_j and since p_j is a partial function by Lemma 5.2, $a' = a''$. Since A is sur, this means that u and v occur in the same recursive call of trans. Hence, $u = v$ by IH3(b).

- $v \in \mathrm{Occ}_{a,i}$. Thus $\xi[v] = \langle a, \pi i \rangle$. Let $\langle b', \rho' \rangle$ be the second node in w. Since w has more than one edge, $\langle b', \rho' \rangle \neq \langle c, \rho \rangle$. Then $\langle a, \pi i \rangle$ occurs in both $\mathrm{rhs}_A(\langle c, \rho \rangle, \sigma)$ and $\mathrm{rhs}_A(\langle b', \rho' \rangle, \sigma)$ which contradicts sur of A.

Case (ii). $u \in \mathrm{Occ}_{a,i}$. Thus, $\xi[u] = \langle a, \pi i \rangle$ and, therefore, there is an edge from $\langle a, \pi i \rangle$ to $\langle c, \rho \rangle$ in g which proves Claim 3(a). Again let us prove that $u = v$. The case $v \in \mathrm{Occ}(\zeta) - \mathrm{Occ}_{a,i}$ is analogous to the second case of Case (i). If $v \in \mathrm{Occ}_{a,i}$, then $\xi[u] = \xi[v] = \langle a, \pi i \rangle$ implies $u = v$ by the sur property of A, which proves Claim 3(b).

We now show that M is ssuri. Let $\sigma \in \Sigma_{\mathrm{root}}^{(k)}$, $q_1, q_2 \in Q$, $p_1, ..., p_k \in P$, $\zeta_1 = \mathrm{rhs}_M(q_1, \sigma, \langle p_1, ..., p_k \rangle)$, and $\zeta_2 = \mathrm{rhs}_M(q_2, \sigma, \langle p_1, ..., p_k \rangle)$. Let $u \in \mathrm{Occ}(\zeta_1)$ and $v \in \mathrm{Occ}(\zeta_2)$ such that $\zeta_1[u] = \zeta_2[v] = \langle (a, I), x_i \rangle \in \langle Q, X_k \rangle$. If $q_1 = q_2$ then, by Claim 3(b), $u = v$. If $q_1 \neq q_2$, then by Claim 3(a), there is a path in $D_\sigma(p_1, ..., p_k)$ from $\langle a, \pi i \rangle$ to $\langle a_1, \pi \rangle$ and from $\langle a, \pi i \rangle$ to $\langle a_2, \pi \rangle$, where $q_1 = (a_1, I_1)$ and $q_2 = (a_2, I_2)$ with $a_1 \neq a_2$ (because otherwise, one is a dummy rule). This contradicts Lemma 5.3, i.e., that $D_\sigma(p_1, ..., p_k)$ is a forest.

The surp property of M can be proved similarly. It follows immediately from (b) of the following claim, which is similar to Claim 3. Let $\sigma \in \Sigma_{\mathrm{root}}^{(k)}$, $p_1, ..., p_k \in P$, $I = \{b_{v_1}, ..., b_{v_m}\}$ with $v_1 < \cdots < v_m$, $\langle c, \rho \rangle \in \mathrm{ins}_\sigma$, and $j \in [m]$.

Claim 4 (a) If y_j occurs in $\text{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle c, \rho \rangle)$, then there is a path from $\langle b_{v_j}, \pi \rangle$ to $\langle c, \rho \rangle$ in $D_\sigma(p_1, ..., p_k)$.

(b) y_j occurs in $\text{trans}_{\sigma, I, \langle p_1, ..., p_k \rangle}(\langle c, \rho \rangle)$ at most once. ∎

Consider the "single-used" MTTs of [Küh97]. The model of a macro tree transducer of Kühnemann has a "built-in" root symbol, just like our ATTs. Hence, $MT_{\text{su}} = \{\{(s, t) \mid (\text{root}(s), t) \in \tau\} \mid \tau \in MTT_{\text{ssuri}}\}$. By the proof of Lemma 5.11, this means that $ATT_{\text{sur}} \subseteq MT^R_{\text{su}}$. Hence, in the presence of regular look-ahead for the MTT, it answers the question whether $ATT_{\text{sur}} \subseteq MT_{\text{su}}$ which is mentioned as an open problem in [Küh97].

It is well known that MTTs are more powerful than ATTs, and thus, in general for an MTT M, there is no ATT A such that $\tau_A = \tau_M$. Classes of MTTs for which an equivalent ATT exists are considered in [CF82, FV97]. Let us discuss why a construction does not work in which the synthesized attributes of A are the states of M and the parameters of M correspond to the inherited attributes of A. Since each state of M has its own set of parameters, we need for each state q of rank m, m inherited attributes $(q, 1), ..., (q, m)$. If $\langle q, x_i \rangle (t_1, ..., t_m)$ occurs in the right-hand side of a (q', σ)-rule, then t_j defines the value for the inherited attribute (q, j). Hence, the $(\langle (q, j), \pi i \rangle, \sigma)$-rule of A is constructed from t_j. Clearly, if there is more than one occurrence of $\langle q, x_i \rangle$ in the σ-rules of M, then this construction only works if each occurrence has the same trees $t_1, ..., t_m$ as parameters (or if different parameters will be deleted during the derivation of M; in [FV97] a characterization of ATT in terms of MTTs is given which is based on this observation). If M is ssuri, then there is at most one occurrence of $\langle q, x_i \rangle$ in the σ-rules of M, and hence, we can construct A in the way as described above. Moreover, if M is also surp, then A is sur. The construction is a special case of the one in [CF82] (and hence, even produces an absolutely noncircular ATT). We repeat the construction here for completeness sake, and to prove the sur property of the ATT. A different proof of the inclusion $MTT_{\text{ssuri}} \subseteq ATT$ is given in Theorem 6.12 of [Küh97].

LEMMA 5.12. $MTT_{\text{ssuri}} \subseteq ATT$ and $MTT_{\text{ssuri, surp}} \subseteq ATT_{\text{sur}}$.

Proof. Let $M = (Q, \Sigma, \Delta, q_0, R)$ be an $\text{MTT}_{\text{ssuri}}$. Before we construct an ATT A which realizes the same translation as M, let us first define some auxiliary notions.

Let $\sigma \in \Sigma^{(k)}$, $q \in Q^{(m)}$, and $i \in [k]$. Since M is ssuri, there is at most one $q' \in Q$ and one $u \in \text{Occ}(\text{rhs}_M(q', \sigma))$ such that $\langle q, x_i \rangle$ occurs in $\text{rhs}_M(q', \sigma)$ at u. We denote q' by $r(\langle q, x_i \rangle, \sigma)$. For $j \in [m]$ the tree $\text{rhs}_M(q', \sigma)/uj$ is called the *jth parameter tree of* $\langle q, x_i \rangle$ *for* σ and is denoted by $p(\langle q, x_i \rangle, \sigma, j)$.

Let $q \in Q^{(m)}$. Then we define the substitution $\Theta_q = \Theta' \Theta''_q$, where $\Theta' = [\![\langle q', x_i \rangle \leftarrow \langle q', \pi i \rangle \mid \langle q', x_i \rangle \in \langle Q, X \rangle]\!]$ and $\Theta''_q = [y_j \leftarrow \langle (q, j), \pi \rangle \mid j \in [m]]$. Note that, in the substitution Θ', $\langle q', x_i \rangle$ is of rank $\text{rank}_Q(q')$ and $\langle q', \pi i \rangle$ is of rank zero (thus, e.g., $\langle q', x_i \rangle (t_1, ..., t_m)\, \Theta'$ equals $\langle q', \pi i \rangle$).

Let us now construct the ATT $A = (\text{Syn}, \text{Inh}, \Sigma, \Delta, \text{root}, a_0, R')$ as

- $\text{Syn} = Q$ and $a_0 = q_0$
- $\text{Inh} = \{(q, j) \mid q \in Q, j \in [\text{rank}_Q(q)]\}$

- Let $\sigma \in \Sigma^{(k)}$. For every $q \in \mathrm{Syn}$ let the rule

$$\langle q, \pi\rangle \to \mathrm{rhs}_M(q, \sigma)\, \Theta_q$$

be in R'_σ and for every $(q, j) \in \mathrm{Inh}$ and $i \in [k]$ let the rule

$$\langle (q, j), \pi i\rangle \to \zeta$$

be in R'_σ, where $\zeta = p(\langle q, x_i\rangle, \sigma, j)\, \Theta_{r(\langle q, x_i\rangle, \sigma)}$ if $r(\langle q, x_i\rangle, \sigma)$ exists and otherwise $\zeta = \mathrm{dummy}$ for an arbitrary symbol $\mathrm{dummy} \in \Delta^{(0)}$. Let $R'_{\mathrm{root}} = \{\langle q_0, \pi\rangle \to \langle q_0, \pi 1\rangle\} \cup \{\langle b, \pi 1\rangle \to \mathrm{dummy} \in \Delta^{(0)} \mid b \in \mathrm{Inh}\}$.

The synthesized attributes of A are the states of M and for every parameter y_j of a state $q \in Q$ there is an inherited attribute (q, j) in A. Let $q \in Q$ and $\sigma \in \Sigma$. If a state $\langle q', x_i\rangle$ occurs in $\mathrm{rhs}(q, \sigma)$ in a non-parameter position (i.e., in $\mathrm{rhs}(q, \sigma)$ only symbols in Δ occur on the path from the root to $\langle q', x_i\rangle$), then this state corresponds to the synthesized attribute q' at the ith child of σ, i.e., to the attribute $\langle q', \pi i\rangle$. If $\langle q', x_i\rangle(t_1, ..., t_m)$ occurs in $\mathrm{rhs}(q, \sigma)$, then $t_1, ..., t_m$ define the inherited attributes $(q', 1), ..., (q', m)$ at the ith child of σ, i.e., the inherited attributes $\langle (q', 1), \pi i\rangle, ..., \langle (q', m), \pi i\rangle$. If there are no such state calls for q' in any $\mathrm{rhs}(q, \sigma)$, then dummy-rules are added. For an example see Example 5.13.

We now show the correctness of the construction of A. For $s \in T_\Sigma$ we say that A is *noncircular on* s if and only if the dependency graph $D(s)$ of s is acyclic; obviously, A is noncircular if it is noncircular on every $s \in T_\Sigma$. For $s \in T_\Sigma$, $\mathrm{nf}(\Rightarrow_{A, \mathrm{root}(s)}, \langle q_0, \varepsilon\rangle)$ is equal to $\mathrm{nf}(\Rightarrow_{A, s}, \langle q_0, \varepsilon\rangle)[\langle b, \varepsilon\rangle \leftarrow \mathrm{dummy} \mid b \in \mathrm{Inh}]$, by the definition of the root-rules. By Claim 1 this is equal to $M_{q_0}(s)$, because $\mathrm{rank}_Q(q_0) = 0$.

Claim 1. Let $q \in Q^{(m)}$ and $s \in T_\Sigma$. Then

(a) A is noncircular on s, and

(b) $\mathrm{nf}(\Rightarrow_{A, s}, \langle q, \varepsilon\rangle) = M_q(s)\Psi_q$, where $\Psi_q = [y_j \leftarrow \langle (q, j), \varepsilon\rangle \mid j \in [m]]$.

Proof. By induction on the structure of s. Let $s = \sigma(s_1, ..., s_k)$, $k \geqslant 0$, $\sigma \in \Sigma^{(k)}$, and $s_1, ..., s_k \in T_\Sigma$. The induction hypothesis is denoted by IH1. To prove part (a), assume that A is circular on s. By IH1(b), $\mathrm{nf}(\Rightarrow_{A, s_i}, \langle q, \varepsilon\rangle)$ is in $T_\Delta(\{\langle (q, j), \varepsilon\rangle \mid j \in [m]\})$. Thus, by Lemma 3.14, there are $(q_1, j_1), ..., (q_n, j_n) \in \mathrm{Inh}$ and $i_1, ..., i_n \in [k]$, such that

(i) for every $v \in [n]$, $\langle (q_v, j_v), \varepsilon\rangle$ occurs in $\mathrm{nf}(\Rightarrow_{A, s_{i_v}}, \langle q_v, \varepsilon\rangle)$,

(ii) for every $v \in [n-1]$, $\langle q_{v+1}, \pi i_{v+1}\rangle$ occurs in $\mathrm{rhs}_A(\langle (q_v, j_v), \pi i_v\rangle, \sigma)$, and

(iii) $\langle q_1, \pi i_1\rangle$ occurs in $\mathrm{rhs}_A(\langle (q_n, j_n), \pi i_n\rangle, \sigma)$.

But in terms of M this means that for every $v \in [n-1]$, $\langle q_{v+1}, x_{i_{v+1}}\rangle$ occurs in the j_vth parameter tree of $\langle q_v, x_{i_v}\rangle$ for σ and $\langle q_1, x_{i_1}\rangle$ occurs in the j_nth parameter tree of $\langle q_n, x_{i_n}\rangle$. Since M is ssuri, there is at most one occurrence of $\langle q_v, x_{i_v}\rangle$ in the set of right-hand sides of σ-rules. This means that there is a σ-rule, the right-hand side ζ of which contains $\langle q_1, x_{i_1}\rangle(..., \langle q_2, x_{i_2}\rangle(...\langle q_n, x_{i_n}\rangle(..., \langle q_1, x_{i_1}\rangle(...), ...)...), ...)$. Hence there are at least two occurrences of $\langle q_1, x_{i_1}\rangle$ in ζ which contradicts ssuri of M and proves part (a) of Claim 1. In fact, this even shows that A is absolutely noncircular, with the "worst case" assumption that each $q \in \mathrm{Syn}$ depends on all $(q, j) \in \mathrm{Inh}$.

Part (b) of Claim 1 follows from Claim 2 (and Lemma 3.4) by taking $t = \mathrm{rhs}_M(q, \sigma)$.

Claim 2. Let $t \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$. If t is a subtree of $\mathrm{rhs}_M(q, \sigma)$, then $\mathrm{nf}(\Rightarrow_{A,s}, t\Theta_q\Pi) = t[\![\cdots]\!]\,\Psi_q$, where $\Pi = [\langle c, \pi v\rangle \leftarrow \langle c, v\rangle \mid \langle c, \pi v\rangle \in \mathrm{outs}_\sigma]$ and $[\![\cdots]\!] = [\![\langle q', x_i\rangle \leftarrow M_{q'}(s_i) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!]$.

Proof. By induction on the structure of t. The induction hypothesis is denoted by IH2.

If $t = y_j \in Y_m$, then $\mathrm{nf}(\Rightarrow_{A,s}, t\Theta_q\Pi) = \mathrm{nf}(\Rightarrow_{A,s}, \langle (q, j), \varepsilon\rangle) = \langle (q, j), \varepsilon\rangle$ and $y_j[\![\cdots]\!]\,\Psi_q = y_j\Psi_q = \langle (q, j), \varepsilon\rangle$.

If $t = \delta(t_1, ..., t_l)$ with $\delta \in \Delta^{(l)}$, $l \geqslant 0$, and $t_1, ..., t_l \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$, then $\mathrm{nf}(\Rightarrow_{A,s}, \delta(t_1, ..., t_l)\,\Theta_q\Pi) = \delta(\mathrm{nf}(\Rightarrow_{A,s}, t_1\Theta_q\Pi), ..., \mathrm{nf}(\Rightarrow_{A,s}, t_l\Theta_q\Pi))$. Since $t_1, ..., t_l$ are subtrees of t and, hence, subtrees of $\mathrm{rhs}_M(q, \sigma)$, we can apply IH2 to get $\delta(t_1[\![\cdots]\!]\,\Psi_q, ..., t_l[\![\cdots]\!]\,\Psi_q)$ which is equal to $\delta(t_1, ..., t_l)[\![\cdots]\!]\,\Psi_q$.

If $t = \langle \bar{q}, x_n\rangle(t_1, ..., t_l)$ with $\langle \bar{q}, x_n\rangle \in \langle Q, X_k\rangle^{(l)}$, $l \geqslant 0$, and $t_1, ..., t_l \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$, then $\langle \bar{q}, x_n\rangle(t_1, ..., t_l)[\![\cdots]\!]\,\Psi_q$ is equal to $M_{\bar{q}}(s_n)[y_j \leftarrow t_j[\![\cdots]\!]\,\Psi_q \mid j \in [l]]$. By adding an extra substitution $\Psi_{\bar{q}}$ we get

$$M_{\bar{q}}(s_n)\,\Psi_{\bar{q}}[\langle (\bar{q}, j), \varepsilon\rangle \leftarrow t_j[\![\cdots]\!]\,\Psi_q \mid j \in [l]].$$

Since t appears in $\mathrm{rhs}_M(q, \sigma)$ for each $j \in [l]$ the rule $\langle (\bar{q}, j), \pi n\rangle \to t_j\Theta_q$ is in R'_σ. By IH2, we can replace $t_j[\![\cdots]\!]\,\Psi_q$ by $\mathrm{nf}(\Rightarrow_{A,s}, t_j\Theta_q\Pi) = \mathrm{nf}(\Rightarrow_{A,s}, \langle (\bar{q}, j), n\rangle)$. We get

$$M_{\bar{q}}(s_n)\,\Psi_{\bar{q}}[\langle (\bar{q}, j), \varepsilon\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle (\bar{q}, j), n\rangle) \mid j \in [l]].$$

The substitution of $\langle (\bar{q}, j), \varepsilon\rangle$ with $j \in [l]$ can be extended to all $\langle b, \varepsilon\rangle$ with $b \in \mathrm{Inh}$, because only attributes of the form $\langle (\bar{q}, j), \varepsilon\rangle$ occur in $M_{\bar{q}}(s_n)\,\Psi_{\bar{q}}$. Applying IH1 we get $\mathrm{nf}(\Rightarrow_{A,s_n}, \langle \bar{q}, \varepsilon\rangle)[\langle b, \varepsilon\rangle \leftarrow \mathrm{nf}(\Rightarrow_{A,s}, \langle b, n\rangle) \mid b \in \mathrm{Inh}]$ which is by Lemma 3.12 equal to $\mathrm{nf}(\Rightarrow_{A,s}, \langle \bar{q}, n\rangle) = \mathrm{nf}(\Rightarrow_{A,s}, \langle \bar{q}, \pi n\rangle\,\Pi) = \mathrm{nf}(\Rightarrow_{A,s}, t\Theta_q\Pi)$. This finishes the proof of Claim 2.

It remains to show that if M is surp, then A is sur. Assume that A is not sur. Then there are $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, $\langle c, \rho\rangle \in \mathrm{outs}_\sigma$, and $\langle d_1, \rho_1\rangle, \langle d_2, \rho_2\rangle \in \mathrm{ins}_\sigma$ such that $\langle c, \rho\rangle$ occurs in both $\mathrm{rhs}_A(\langle d_1, \rho_1\rangle, \sigma)$ and $\mathrm{rhs}_A(\langle d_2, \rho_2\rangle, \sigma)$, and either (i) $\langle d_1, \rho_1\rangle \neq \langle d_2, \rho_2\rangle$ or (ii) there are two occurrences of $\langle c, \rho\rangle$ in $\mathrm{rhs}_A(\langle d_1, \rho_1\rangle, \sigma)$.

Let us first consider the case that $\langle c, \rho\rangle = \langle q, \pi i\rangle$ with $i \in [k]$. If $d_1, d_2 \in \mathrm{Syn}$, then by the definition of the rules of A, $\langle q, x_i\rangle$ occurs in both $\mathrm{rhs}_M(d_1, \sigma)$ and $\mathrm{rhs}_M(d_2, \sigma)$ which contradicts ssuri of M for case (i). For case (ii) it means that there are two occurrences of $\langle q, x_i\rangle$ in $\mathrm{rhs}_M(d_1, \sigma)$ which also contradicts ssuri of M. If $d_1 \in \mathrm{Syn}$ and $d_2 \in \mathrm{Inh}$, then $\langle q, x_i\rangle$ occurs in $\mathrm{rhs}_M(d_1, \sigma)$ in a non-parameter position, and it occurs in a parameter tree, which contradicts ssuri of M. If $d_1, d_2 \in \mathrm{Inh}$, then (i) $\langle q, x_i\rangle$ occurs in two distinct parameter trees or (ii) there are two distinct occurrences of $\langle q, x_i\rangle$ in one parameter tree, which both contradict ssuri of M.

We now consider the case that $\langle c, \rho\rangle = \langle (q, j), \pi\rangle$ with $(q, j) \in \mathrm{Inh}$. If $d_1, d_2 \in \mathrm{Syn}$, then by the definition of the rules of A this means that $d_1 = d_2 = q$; hence case (i) cannot occur and case (ii) means that there are two distinct occurrences of y_j in $\mathrm{rhs}_M(q, \sigma)$ which contradicts surp of M. If $d_1 \in \mathrm{Syn}$ and $d_2 \in \mathrm{Inh}$, then y_j occurs in a non-parameter position in $\mathrm{rhs}_M(q, \sigma)$ and in a parameter tree in $\mathrm{rhs}_M(q, \sigma)$ which contradicts surp of M. If $d_1, d_2 \in \mathrm{Inh}$, then (i) y_j occurs in two distinct parameter trees in $\mathrm{rhs}_M(q, \sigma)$ or (ii) there are two distinct occurrences of y_j in one parameter tree of $\mathrm{rhs}_M(q, \sigma)$, which both contradict surp of M. ∎

The construction in the proof of Lemma 5.12 is illustrated by the following very simple example.

EXAMPLE 5.13. Let $M = (Q, \Sigma, \Delta, q_0, R)$ be the MTT with $Q = \{q_0^{(0)}, q^{(1)}, q'^{(1)}\}$, $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$, $\Delta = \{\alpha^{(0)}\}$; R contains the rules (we do not show the q_0-rules):

$$\langle q, \sigma(x_1, x_2)\rangle(y) \to \langle q', x_1\rangle(\langle q, x_2\rangle(y))$$

$$\langle q', \sigma(x_1, x_2)\rangle(y) \to \langle q, x_1\rangle(y)$$

$$\langle q, \alpha\rangle(y) \to y$$

$$\langle q', \alpha\rangle(y) \to y.$$

By the construction in the proof of Lemma 5.12, $A = (Q, \{(q, 1), (q', 1)\}, \Sigma, \Delta, \mathrm{root}, q_0, R')$ with

$$\begin{aligned} R'_\sigma = \{&\langle q, \pi\rangle \to \langle q', \pi 1\rangle \\ &\langle q', \pi\rangle \to \langle q, \pi 1\rangle \\ &\langle (q, 1), \pi 1\rangle \to \langle (q', 1), \pi\rangle \\ &\langle (q', 1), \pi 1\rangle \to \langle q, \pi 2\rangle \\ &\langle (q, 1), \pi 2\rangle \to \langle (q, 1), \pi\rangle \\ &\langle (q', 1), \pi 2\rangle \to \mathrm{dummy}\} \end{aligned} \qquad \begin{aligned} R'_\alpha = \{&\langle q, \pi\rangle \to \langle (q, 1), \pi\rangle \\ &\langle q', \pi\rangle \to \langle (q', 1), \pi\rangle\} \end{aligned}$$

To illustrate Claim 1 (in the proof of Lemma 5.12), let $s = \sigma(\alpha, \alpha)$. Then $\langle q', \sigma(\alpha, \alpha)\rangle(y) \Rightarrow_M \langle q, \alpha\rangle(y) \Rightarrow_M y = M_{q'}(s)$. The corresponding derivation by A is $\langle q', \varepsilon\rangle \Rightarrow_{A,s} \langle q, 1\rangle \Rightarrow_{A,s} \langle (q, 1), 1\rangle \Rightarrow_{A,s} \langle (q', 1), \varepsilon\rangle$. And $\langle q, \sigma(\alpha, \alpha)\rangle(y) \Rightarrow_M \langle q', \alpha\rangle(\langle q, \alpha\rangle(y)) \Rightarrow_M \langle q', \alpha\rangle(y) \Rightarrow_M y$ corresponds to the derivation $\langle q, \varepsilon\rangle \Rightarrow_{A,s} \langle q', 1\rangle \Rightarrow_{A,s} \langle (q', 1), 1\rangle \Rightarrow_{A,s} \langle q, 2\rangle \Rightarrow_{A,s} \langle (q, 1), 2\rangle \Rightarrow_{A,s} \langle (q, 1), \varepsilon\rangle$.

Altogether we have shown in this and the previous section that, in the presence of regular look-ahead for sur MTTs and look-ahead for sur ATTs, the corresponding classes of translations coincide. This is our first main result.

THEOREM 5.14. $ATT^R_{\text{sur}} = MTT^R_{\text{sur}}$.

Proof. By Theorem 5.10 and Lemma 5.12, $MTT^R_{\text{sur}} = T^R\text{-}REL \circ MTT_{\text{ssur}} \subseteq T^R\text{-}REL \circ ATT_{\text{sur}}$. By Theorem 4.4 and the definition of ATT^R_{sur} this proves that $MTT^R_{\text{sur}} \subseteq ATT^R_{\text{sur}}$. By Theorem 4.4 and Lemma 5.11, $ATT^R_{\text{sur}} \subseteq T^R\text{-}REL \circ MTT^R_{\text{sur}}$ which equals MTT^R_{sur} because by Theorem 5.10 and Lemma 4.5, MTT^R_{sur} is closed under left composition with $T^R\text{-}REL$. ▮

Note that, by Theorem 5.10, an alternative way of expressing this result is that $T^R\text{-}REL \circ ATT_{\text{sur}} = T^R\text{-}REL \circ MTT_{\text{ssur}}$.

6. FINITE COPYING MTTs

In the previous section we have investigated single use restricted MTT$^{\text{R}}$s. The distinction between the copying done by states and that done by parameters led to the notions of suri and surp, which together form the single use restriction for MTT$^{\text{R}}$s. In this section we want to introduce a more liberal, dynamic way of restricting the copying power of MTT$^{\text{R}}$s, yet obtaining the same class MTT^R_{sur} of translations realized by sur MTT$^{\text{R}}$s.

The notion of finite copying was introduced by Aho and Ullmann [AU71] for generalized syntax-directed translation schemes, which are closely related to top-down tree transducers. Finite copying top-down tree transducers were further investigated in, e.g., [ERS80]. Intuitively a top-down tree transducer is finite copying, if every input subtree s/u is processed only a bounded number of times. Since the state sequence of s at u contains precisely the states that process the tree s/u, this means that the lengths of the state sequences are bounded. For macro tree transducers we take this definition over and call it *finite copying in the input* (for short, fci). But a macro tree transducer can also copy by means of its parameters (cf. also the discussion at the end of Section 3.1). Thus, we also define a notion of *finite copying in the parameters* (for short, fcp). Intuitively it means that each parameter may only be copied a bounded number of times in the q-translation of an input tree. For the notion of state sequence of an MTT$^{\text{R}}$, see Definition 3.7.

DEFINITION 6.1 (Finite copying in the input). Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT$^{\text{R}}$. Then M is *finite copying in the input* (for short, fci), if there is an $N \in \mathbb{N}$ such that for every $s \in T_\Sigma$ and $u \in \text{Occ}(s)$: $|\text{sts}_M(s, u)| \leqslant N$. The number N is called an *input copying bound for M*.

For a ranked alphabet Σ, a tree $t \in T_\Sigma(Y)$, and $j \geqslant 1$, we denote by $c_j(t)$ the number of occurrences of y_j in t, i.e., $c_j(t) = |\{u \in \text{Occ}(t) \mid t[u] = y_j\}|$.

DEFINITION 6.2 (Finite copying in the parameters). Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT$^{\text{R}}$. Then M is *finite copying in the parameters* (for short, fcp), if there is an $N \in \mathbb{N}$ such that for every $q \in Q^{(m)}$, $s \in T_\Sigma$, and $j \in [m]$, $c_j(M_q(s)) \leqslant N$. The number N is called a *parameter copying bound for M*.

The class of translations which can be realized by $\mathrm{MTT^R}$s which are fci is denoted by MTT^R_{fci}, and analogously for fcp. We say that an $\mathrm{MTT^R}$ is *finite copying* (for short, fc), if it is both, fci and fcp. The class $MTT^R_{\mathrm{fci,\,fcp}}$ of translations realized by fc $\mathrm{MTT^R}$s is also denoted by MTT^R_{fc}. The class $T_{\mathrm{fc}} = T_{\mathrm{fci}}$ of translations realized by finite copying top-down tree transducers is the one known from the literature (e.g., [ERS80]).

A rather obvious consequence of the finite copying property is that there is a bound on the number of translations $M_q(s/u)$ of an input subtree s/u that occur as part of the output tree $\tau_M(s)$. In fact, if M has input copying bound I and parameter copying bound N, then this bound is $I \cdot N^{I-1}$. To see this, recall Lemma 3.6. The number of $\langle\!\langle q, p\rangle\!\rangle$'s in $\hat{M}_{q_0}(s[u \leftarrow p])$ is bounded by I, and every $M_q(s/u)$ is copied at most N^{I-1} times by the others. We note that this property could have been taken as alternative definition of fci, for fcp $\mathrm{MTT^R}$s.

It will be shown in this section that $MTT^R_{\mathrm{fc}} = MTT^R_{\mathrm{sur}}$. We have already observed (without proof, after Definitions 5.4 and 5.7) that every $\mathrm{MTT^R_{sur}}$ is finite copying: the parameter copying bound is 1 and the input copying bound is $|Q|$, the number of states. Thus, as will be proved in detail in Theorem 6.12, $MTT^R_{\mathrm{sur}} \subseteq MTT^R_{\mathrm{fc}}$. We now turn to the proof of the other inclusion.

If an $\mathrm{MTT^R}$ M is fcp, then it can be turned into a surp $\mathrm{MTT^R}$ M'; the look-ahead of M' is used to determine how many copies of each parameter a state needs. The construction preserves the fci property.

LEMMA 6.3. $MTT^R_{\mathrm{fcp}} \subseteq MTT^R_{\mathrm{surp}}$ and $MTT^R_{\mathrm{fci,\,fcp}} \subseteq MTT^R_{\mathrm{fci,\,surp}}$.

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\mathrm{MTT^R_{fcp}}$ and let N be a parameter copying bound for M.

Thus, for every $s \in T_\Sigma$, $q \in Q^{(m)}$, and $j \in [m]$, $c_j(M_q(s)) \leqslant N$. We will construct a surp MTT M' from M in such a way that for every $q \in Q^{(m)}$ the numbers $c_j(M_q(s))$, $j \in [m]$, of occurrences of y_j in $M_q(s)$ are determined by the look-ahead automaton of M'. This is possible because the set $\{t \in T_\Delta(Y_m) \mid c_j(t) = (f(q))(j)\}$ for a fixed mapping $f(q)\colon [m] \to \{0, ..., N\}$ is a regular tree language, together with the fact that regular tree languages are preserved by inverse macro tree transductions (Theorem 7.4(1) of [EV85]) this means that we can determine the numbers $c_j(M_q(s))$ (i.e., the mappings $f(q)$) by regular look-ahead; indeed, below we give a construction of such a look-ahead automaton. Then, in the rules of M' we simply provide the correct amount of copies of each parameter; i.e., we replace every occurrence of $\langle q, x_i\rangle(t_1, ..., t_m)$ by $\langle (q, w), x_i\rangle(\underbrace{t_1, ..., t_1}_{w(1)\text{ times}}, ..., \underbrace{t_m, ..., t_m}_{w(m)\text{ times}})$, where (q, w) is a new state of rank $c_1(M_q(s_i)) + \cdots + c_m(M_q(s_i))$ and $w(j) = c_j(M_q(s_i))$ is the number of copies of y_j in $M_q(s_i)$ which is determined by look-ahead.

We now turn to the formal construction. Let $M' = (Q', P', \Sigma, \Delta, (q_0, \varnothing), R', h')$ be the $\mathrm{MTT^R}$ with $Q' = \{(q, w) \mid q \in Q^{(m)}, w\colon [m] \to \{0, ..., N\}\}$, $\mathrm{rank}_{Q'}((q, w)) = w(1) + \cdots + w(m)$, and $P' = P \times F$, where F is the set of all functions f which associate with each $q \in Q^{(m)}$ a function $f(q)$ from $[m]$ to $\{0, ..., N\}$. For a function $w\colon [m] \to \{0, ..., N\}$ we define the function $W\colon [m] \to \{0, ..., N\}$ such that $W(j) = w(1) + \cdots + w(j)$ for $j \in [m]$. Thus, $\mathrm{rank}_{Q'}((q, w)) = W(m)$ for $q \in Q^{(m)}$.

The look-ahead automaton of M' is constructed in such a way that for $s \in T_\Sigma$ it arrives in state (p, f), where $p = h(s)$ and f is a mapping with $(f(q))(j) = c_j(M_q(s))$ for every $q \in Q^{(m)}$ and $j \in [m]$. Formally, for $\sigma \in \Sigma^{(k)}$ with $k \geqslant 0$ and $(p_1, f_1), ..., (p_k, f_k) \in P'$, $h'_\sigma((p_1, f_1), ..., (p_k, f_k)) = (p_0, f_0)$, where $p_0 = h_\sigma(p_1, ..., p_k)$ and for every $q \in Q^{(m)}$ and $j \in [m]$, $(f_0(q))(j) = \text{copy}_j(\text{rhs}_M(q, \sigma, \langle p_1, ..., p_k \rangle))$. For $t \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, $\text{copy}_j(t)$ is recursively defined as follows. (Note that $\text{copy}_j(t)$ depends on $f_1, ..., f_k$.) If $t \in Y_m$, then $\text{copy}_j(t) = c_j(t)$. If $t = \delta(t_1, ..., t_l)$ with $\delta \in \Delta^{(l)}$, $l \geqslant 0$, and $t_1, ..., t_l \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, then $\text{copy}_j(t) = \text{copy}_j(t_1) + \cdots + \text{copy}_j(t_l)$. If $t = \langle r, x_i \rangle(t_1, ..., t_l)$ with $\langle r, x_i \rangle \in \langle Q, X_k \rangle^{(l)}$, $l \geqslant 0$, and $t_1, ..., t_l \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, then $\text{copy}_j(t) = (f_i(r))(1) \cdot \text{copy}_j(t_1) + (f_i(r))(2) \cdot \text{copy}_j(t_2) + \cdots + (f_i(r))(l) \cdot \text{copy}_j(t_l)$.

For every $(q, w) \in Q'$ with $q \in Q^{(m)}$, $\sigma \in \Sigma^{(k)}$, $m, k \geqslant 0$, and $(p_1, f_1), ..., (p_k, f_k) \in P'$, let the rule

$$\langle (q, w), \sigma(x_1, ..., x_k) \rangle (y_1, ..., y_{W(m)}) \to \zeta \qquad \langle (p_1, f_1), ..., (p_k, f_k) \rangle \qquad (*)$$

be in R', such that $\zeta = \text{dummy} \in \Delta^{(0)}$ if $f_0(q) \neq w$, where $(p_0, f_0) = h'_\sigma((p_1, f_1), ..., (p_k, f_k))$, and otherwise $\zeta = \text{rhs}_M(q, \sigma, \langle p_1, ..., p_k \rangle)\, R_1 R_2$, where R_1 denotes the substitution

$$[\![\langle r, x_i \rangle \leftarrow \langle (r, w_r), x_i \rangle (\underbrace{y_1, ..., y_1}_{w_r(1) \text{ times}}, ..., \underbrace{y_n, ..., y_n}_{w_r(n) \text{ times}}) \mid \langle r, x_i \rangle \in \langle Q, X_k \rangle^{(n)}, w_r = f_i(r)]\!]$$

and R_2 denotes the replacement for every $j \in [m]$, of the vth occurrence of y_j (with respect to pre-order) by y_{new}, where $\text{new} = W(j-1) + v$. This ends the definition of M'. An example is given in Example 6.4.

It is straightforward to show (by induction on the structure of s) that the look-ahead automaton of M' is defined in such a way that for every $s \in T_\Sigma$, if $h'(s) = (p_0, f_0)$, then $p_0 = h(s)$ and for every $q \in Q^{(m)}$ and $j \in [m]$, $f_0(q)(j) = c_j(M_q(s))$. In the induction step this follows from Lemma 3.4 and a proof of the fact that for $t \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$ and $j \in [m]$, $\text{copy}_j(t) = c_j(t[\![\cdots]\!])$, where $[\![\cdots]\!] = [\![\langle r, x_i \rangle \leftarrow M_r(s_i) \mid \langle r, x_i \rangle \in \langle Q, X_k \rangle]\!]$. This also shows that M' is well defined: the numbers computed by its look-ahead automaton are indeed in $\{0, ..., N\}$.

Let us now show that M' is surp. Consider a rule in R' of the form $(*)$, such that $f_0(q) = w$. (Note that both R_1 and $\text{copy}_j(t)$ depend on $f_1, ..., f_k$.)

Claim 1. Let $t \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$. For every $j \in [m]$, $c_j(tR_1) = \text{copy}_j(t)$.

Proof. By induction on the structure of t. If $t = y_l \in Y_m$, then $c_j(y_l R_1) = c_j(y_l) = \text{copy}_j(y_l)$. If $t = \delta(t_1, ..., t_l)$ with $\delta \in \Delta^{(l)}$, $l \geqslant 0$, and $t_1, ..., t_l \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, then $c_j(\delta(t_1, ..., t_l)\, R_1) = c_j(t_1 R_1) + \cdots + c_j(t_l R_1)$. By the induction hypothesis this equals $\text{copy}_j(t_1) + \cdots + \text{copy}_j(t_l) = \text{copy}_j(\delta(t_1, ..., t_l))$. If $t = \langle r, x_i \rangle(t_1, ..., t_l)$ with $\langle r, x_i \rangle \in \langle Q, X_k \rangle^{(l)}$, $l \geqslant 0$, and $t_1, ..., t_l \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, then $c_j(\langle r, x_i \rangle (t_1, ..., t_l)\, R_1)$ equals

$$c_j(\langle (r, w_r), x_i \rangle (\underbrace{t_1 R_1, ..., t_1 R_1}_{w_r(1) \text{ times}}, ..., \underbrace{t_l R_1, ..., t_l R_1}_{w_r(l) \text{ times}})),$$

where $w_r = f_i(r)$. This equals $w_r(1) \cdot c_j(t_1 R_1) + w_r(2) \cdot c_j(t_2 R_1) + \cdots + w_r(l) \cdot c_j(t_l R_1)$. By applying the induction hypothesis we get $w_r(1) \cdot \text{copy}_j(t_1) + \cdots + w_r(l) \cdot \text{copy}_j(t_l) = \text{copy}_j(\langle r, x_i\rangle(t_1, ..., t_l))$, which finishes the proof of Claim 1.

By Claim 1 and the definition of f_0, $c_j(\xi R_1) = w(j)$ for every $j \in [m]$, where $\xi = \text{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$. By the definition of R_2 it follows immediately that $c_v(\zeta) = c_v(\xi R_1 R_2) = 1$ for every $v \in [W(m)]$; hence M' is surp.

The correctness of M' follows from Claim 2 by taking $q = q_0$ and $w = \varnothing$.

Claim 2. Let $q \in Q^{(m)}$ and $s \in T_\Sigma$, and let $w = f_0(q)$, where $(p_0, f_0) = h'(s)$. Then $M'_{(q,w)}(s)\, \Theta_w = M_q(s)$, where $\Theta_w = [y_v \leftarrow y_j \mid j \in [m], W(j-1)+1 \leqslant v \leqslant W(j)]$.

Proof. By induction on the structure of s. Let $s = \sigma(s_1, ..., s_k)$ with $\sigma \in \Sigma^{(k)}$ and $s_1, ..., s_k \in T_\Sigma$, and let $h'(s_i) = (p_i, f_i)$ for $i \in [k]$. By the definition of the rule $(*)$ of M', $M'_{(q,w)}(\sigma(s_1, ..., s_k))\, \Theta_w$ equals $\xi R_1 R_2 [\![\cdots]\!]\, \Theta_w$, where $\xi = \text{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$ and $[\![\cdots]\!] = [\![\langle (r, w_r), x_i\rangle \leftarrow M'_{(r, w_r)}(s_i) \mid \langle (r, w_r), x_i\rangle \in \langle Q', X_k\rangle]\!]$. Clearly, since $[\![\cdots]\!]$ does not introduce parameters, the substitutions $[\![\cdots]\!]$ and Θ_w can be interchanged, and since $R_2 \Theta_w$ is the identity on ξR_1 because (as shown above) $c_j(\xi R_1) = w(j)$ for every $j \in [m]$, we get $\xi R_1 [\![\cdots]\!]$. Since second-order substitution is associative we can combine R_1 and $[\![\cdots]\!]$ to get

$$\xi [\![\langle r, x_i\rangle \leftarrow M'_{(r, w_r)}(s_i)\, \Theta_{w_r} \mid \langle r, x_i\rangle \in \langle Q, X_k\rangle^{(n)}, w_r = f_i(r)]\!],$$

where Θ_{w_r} is the substitution $[y_v \leftarrow y_j \mid j \in [n], W_r(j-1)+1 \leqslant v \leqslant W_r(j)]$. By the induction hypothesis, $M'_{(r, w_r)}(s_i)\, \Theta_{w_r} = M_r(s_i)$. Thus we get $\xi [\![\langle r, x_i\rangle \leftarrow M_r(s_i) \mid \langle r, x_i\rangle \in \langle Q, X_k\rangle]\!]$ which, by the correctness of the first part of the look-ahead of M', equals $M_q(s)$ and finishes the proof of Claim 2.

It remains to show that if M is fci, then so is M'. If M is fci, then there is an $I \geqslant 0$, such that I is an input copying bound for M. Let $\hat{M}$ and $\hat{M}'$ be the extensions of M and M', respectively (see Definition 3.5). Let $(p, f) \in P'$ and let $s \in T_\Sigma(\{(p, f)\})$ such that (p, f) occurs at most once in s. The following claim is the "extended" version of Claim 2. Let $q \in Q^{(m)}$, $m \geqslant 0$, $(p_0, f_0) = \hat{h}'(s)$, and $w = f_0(q)$.

Claim 3. $\hat{M}'_{(q,w)}(s)\, \Theta_w = \hat{M}_q(s[(p, f) \leftarrow p])\, \Psi$, where Θ_w is as in Claim 2 and $\Psi = [\![\langle\!\langle r, p\rangle\!\rangle \leftarrow \langle\!\langle (r, w_r), (p, f)\rangle\!\rangle(\underbrace{y_1, ..., y_1}_{w_r(1)\text{ times}}, ..., \underbrace{y_n, ..., y_n}_{w_r(n)\text{ times}}) \mid r \in Q^{(n)}, w_r = f(r)]\!]$.

Proof. By induction on the structure of s. If $s = (p, f)$, then $\hat{M}'_{(q,w)}((p, f))\, \Theta_w$ equals $\langle\!\langle (q, w), (p, f)\rangle\!\rangle(y_1, ..., y_{W(m)})\, \Theta_w$. By applying Θ_w we get $\langle\!\langle (q, w), (p, f)\rangle\!\rangle(\underbrace{y_1, ..., y_1}_{w(1)\text{ times}}, ..., \underbrace{y_m, ..., y_m}_{w(m)\text{ times}})$. Since $f_0 = f$, this equals $\langle\!\langle q, p\rangle\!\rangle(y_1, ..., y_m)\, \Psi = \hat{M}_q(p)\, \Psi = \hat{M}_q(s[(p, f) \leftarrow p])\, \Psi$. The proof for $s = \sigma(s_1, ..., s_k)$ is as in Claim 2, except that the induction hypothesis is $\hat{M}'_{(r, w_r)}(s_i)\, \Theta_{w_r} = \hat{M}_r(s_i[(p, f) \leftarrow p])\, \Psi$. Then $\xi [\![\langle r, x_i\rangle \leftarrow \hat{M}_r(s_i[(p, f) \leftarrow p])\, \Psi \mid \langle r, x_i\rangle \in \langle Q, X_k\rangle]\!]$ equals $\hat{M}_q(s[(p, f) \leftarrow p])\, \Psi$ because $\xi = \text{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$ contains no elements of $\langle\!\langle Q, \{p\}\rangle\!\rangle$.

Let $s \in T_\Sigma$ and $u \in \mathrm{Occ}(s)$. Then $\mathrm{sts}_{M'}(s, u)$ is the sequence of states which occur in $\hat{M}'_{(q_0, \varnothing)}(s[u \leftarrow (p, f)])$, where $(p, f) = h'(s/u)$. By Claim 3, $\hat{M}'_{(q_0, \varnothing)}(s[u \leftarrow (p, f)])$ equals $\hat{M}_{q_0}(s[u \leftarrow p])\,\Psi$. The number of occurrences of $\langle\!\langle q, p \rangle\!\rangle$ with $q \in Q$ in $\hat{M}_{q_0}(s[u \leftarrow p])$ is bounded by I. Hence the number of occurrences of elements of $\langle\!\langle Q', \{(p, f)\} \rangle\!\rangle$ in $\hat{M}'_{(q_0, \varnothing)}(s[u \leftarrow (p, f)])$ is also bounded. In particular, the number $I \cdot N^{I-1}$ is an input copying bound for M'. This is true because at most I occurrences of elements in $\langle\!\langle Q, \{p\} \rangle\!\rangle$ are present in $\hat{M}_{q_0}(s[u \leftarrow p])$, and the substitution of an occurrence v of $\langle\!\langle r, p \rangle\!\rangle$ by Ψ produces at most N copies of each subtree of v. ▮

Let us consider a simple example illustrating the construction in the proof of Lemma 6.3.

EXAMPLE 6.4. Let $M = (Q, \{p\}, \Sigma, \Delta, q_0, R, h)$ be the $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{fcp}}$ with $Q = \{q_0^{(0)}, q^{(3)}\}$, $\Sigma = \{\gamma^{(1)}, \alpha^{(0)}, \beta^{(0)}\}$, $\Delta = \{\delta^{(5)}, \sigma^{(3)}, \alpha^{(0)}\}$, $h_\sigma(p, ..., p) = p$ for every $\sigma \in \Sigma$ and R consisting of the rules (we omit the q_0-rules).

$$\langle q, \gamma(x_1)\rangle(y_1, y_2, y_3) \to \langle q, x_1\rangle(y_3, y_2, y_1) \qquad \langle p\rangle$$

$$\langle q, \alpha\rangle(y_1, y_2, y_3) \to \delta(y_3, y_2, y_2, y_3, y_2)$$

$$\langle q, \beta\rangle(y_1, y_2, y_3) \to \sigma(y_1, y_1, y_2)$$

Clearly, $N = 3$ is a parameter copying bound for M. We now construct the $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{surp}}$ M', following the construction in the proof of Lemma 6.3. For convenience, we denote a function $w\colon [m] \to \{0, ..., 3\}$ by the string $w(1) \cdots w(m)$; in particular $\varnothing$ is denoted by ε. Let $M' = (Q', P', \Sigma, \Delta, (q_0, \varepsilon), R', h')$. The set Q' of states of M' equals $\{(q_0, \varepsilon)^{(0)}, (q, 000)^{(0)}, (q, 001)^{(1)}, ..., (q, 333)^{(9)}\}$. Since P is a singleton we simply let $P' = F = \{f_{000}, f_{001}, ..., f_{333}\}$, where $f_{ijk}(q_0) = \varepsilon$ and $f_{ijk}(q) = ijk$ for $i, j, k \in \{0, ..., 3\}$. Then $h_\alpha() = f$, where $(f(q))(j) = \mathrm{copy}_j(\mathrm{rhs}_M(q, \alpha, \langle\,\rangle)) = \mathrm{copy}_j(\delta(y_3, y_2, y_2, y_3, y_2)) = \mathrm{copy}_j(y_3) + \mathrm{copy}_j(y_2) + \cdots + \mathrm{copy}_j(y_2)$. For $j = 1, 2, 3$ this equals 0, 3, and 2, respectively. Thus $h_\alpha(\,) = f_{032}$. For β we get $h_\beta() = f_{210}$. For γ let us only consider those look-ahead states which actually occur in computations; then $h_\gamma(f_{032}) = f_{230}$, $h_\gamma(f_{210}) = f_{012}$, $h_\gamma(f_{230}) = f_{032}$, and $h_\gamma(f_{012}) = f_{210}$.

Let us now construct the rules of M'. Again we consider only those rules that will actually occur in derivations of M'. For the $((q, 032), \gamma, \langle f_{230}\rangle)$-rule we show in detail how the right-hand side is obtained. By definition, we get $\langle q, x_1\rangle(y_3, y_2, y_1)\, R_1 R_2 = \langle (q, 230), x_1\rangle(y_3, y_3, y_2, y_2, y_2)\, R_2 = \langle (q, 230), x_1\rangle(y_4, y_5, y_1, y_2, y_3)$. The rules are

$$\langle (q, 032), \gamma(x_1)\rangle(y_1, ..., y_5) \to \langle (q, 230), x_1\rangle(y_4, y_5, y_1, y_2, y_3) \qquad \langle f_{230}\rangle$$

$$\langle (q, 230), \gamma(x_1)\rangle(y_1, ..., y_5) \to \langle (q, 032), x_1\rangle(y_3, y_4, y_5, y_1, y_2) \qquad \langle f_{032}\rangle$$

$$\langle (q, 210), \gamma(x_1)\rangle(y_1, y_2, y_3) \to \langle (q, 012), x_1\rangle(y_3, y_1, y_2) \qquad \langle f_{012}\rangle$$

$$\langle (q, 012), \gamma(x_1)\rangle(y_1, y_2, y_3) \to \langle (q, 210), x_1\rangle(y_2, y_3, y_1) \qquad \langle f_{210}\rangle$$

$$\langle (q, 032), \alpha\rangle(y_1, ..., y_5) \to \delta(y_4, y_1, y_2, y_5, y_3)$$

$$\langle (q, 210), \beta\rangle(y_1, y_2, y_3) \to \sigma(y_1, y_2, y_3)$$

Let us now verify (see Claim 2) that for the input $s=\gamma(\alpha)$, $M'_{(q,230)}(s)\,\Theta_{230}=M_q(s)$. We get $\langle (q,230), \gamma(\alpha)\rangle(y_1, ..., y_5) \Rightarrow_{M'} \langle (q,032), \alpha)\rangle(y_3, y_4, y_5, y_1, y_2) \Rightarrow_{M'} \delta(y_1, y_3, y_4, y_2, y_5)$. The substitution Θ_{230} replaces both y_1 and y_2 by y_1 and it replaces y_3, y_4, and y_5 by y_2. Thus $\delta(y_1, y_3, y_4, y_2, y_5)\,\Theta_{230}=\delta(y_1, y_2, y_2, y_1, y_2)$. For M, $\langle q, \gamma(\alpha)\rangle(y_1, y_2, y_3) \Rightarrow_M \langle q, \alpha\rangle(y_3, y_2, y_1) \Rightarrow_M \delta(y_1, y_2, y_2, y_1, y_2)$.

Before we show that every fci $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{surp}}$ can be turned into an equivalent one which is suri, we need the following normal form.

Nondeleting normal form. In proofs it is sometimes useful to know that all parameters which occur in the rules of an $\mathrm{MTT}^{\mathrm{R}}$ M are actually used to generate output, i.e., that for a state q of rank m all parameters in Y_m occur in the right-hand side of each q-rule. We call an MTT with this property *nondeleting* (*in the parameters*). The nondeleting property is comparable with the reducedness of context-free grammars. For the IO macro grammars, which can be seen as MTTs without input and with strings in the right-hand sides of the rules, Fischer proves a nondeleting ("argument-preserving") normal form in Theorem 3.1.10 of [Fis68]. Our proof will be essentially the same, but it will need regular look-ahead to preserve the determinism of the MTT. Given an arbitrary $\mathrm{MTT}^{\mathrm{R}}$, how can we construct a nondeleting $\mathrm{MTT}^{\mathrm{R}}$ M' which realizes the same translation as M? The set of parameters occurring in the right-hand side of a q-rule can be any subset of Y_m. Thus, by taking states of the form (q, I) with $I\subseteq[m]$ we can code the information, which parameters are needed, into the states of M'. However, how do we know which parameters, occurring in the right-hand side of a (q,σ)-rule will be deleted during the computation of M? Similar to the discussion in the proof of Lemma 6.3, this is a regular property and, thus can be determined by regular look-ahead. Indeed, this follows by the facts that $\{t\in T_\Delta(Y_m) \mid \text{there is an } i\in I \text{ such that } y_i \text{ occurs in } t\}$ is regular for every $I\subseteq[m]$ and that regular tree languages are closed under inverse macro tree transductions (cf. Theorem 7.4 of [EV85]); just consider τ_{M_q}, where M_q is equal to M but with initial state q. However, below we give a concrete construction. We now define the notion of nondeleting for $\mathrm{MTT}^{\mathrm{R}}$'s.

DEFINITION 6.5 (Nondeleting). Let $M=(Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\mathrm{MTT}^{\mathrm{R}}$. If for every $q\in Q^{(m)}$, $\sigma\in\Sigma^{(k)}$, $p_1, ..., p_k\in P$, and $j\in[m]$, y_j occurs in $\mathrm{rhs}(q, \sigma, \langle p_1, ..., p_k\rangle)$, then M is *nondeleting*.

Let us now show that the nondeleting property is a normal form for $\mathrm{MTT}^{\mathrm{R}}$'s.

LEMMA 6.6. For every $\mathrm{MTT}^{\mathrm{R}}$ M there is a nondeleting $\mathrm{MTT}^{\mathrm{R}}$ M' such that $\tau_{M'}=\tau_M$. The construction involved preserves fci, suri, ssuri, fcp, and surp.

Proof. Let $M=(Q, P, \Sigma, \Delta, q_0, R, h)$. The construction of M' is similar to the construction in the proof of Lemma 6.3. In fact, instead of determining by regular look-ahead the precise number $c_j(M_q(s))$ of occurrences of y_j in $M_q(s)$, we now only need to determine whether y_j occurs in $M_q(s)$ or not. Again, this is done by regular look-ahead. We denote by F the set of all functions which associate with every $q\in Q^{(m)}$ a subset of $[m]$. For a subset I of $\mathbb{N}$ we denote by $I(j)$ the jth element of I with respect to $<$. Let $M'=(Q', P', \Sigma, \Delta\cup\{d^{(2)}\}, (q_0, \varnothing), R', h')$, where $Q'=\{(q, I)^{(|I|)} \mid q\in Q^{(m)}, I\subseteq[m]\}$ and $P'=P\times F$.

The look-ahead automaton of M' is defined as follows: For $\sigma \in \Sigma^{(k)}$ and $(p_1, f_1), ..., (p_k, f_k) \in P'$ let $h'_\sigma((p_1, f_1), ..., (p_k, f_k)) = (p_0, f_0)$, where $p_0 = h_\sigma(p_1, ..., p_k)$ and for every $q \in Q^{(m)}$, $f_0(q) = \mathrm{oc}(\mathrm{rhs}_M(q, \sigma, \langle p_1, ..., p_k \rangle))$. For every $t \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, $\mathrm{oc}(t) \subseteq [m]$ is recursively defined as follows. (Note that $\mathrm{oc}(t)$ depends on $f_1, ..., f_k$.) If $t = y_j \in Y_m$, then $\mathrm{oc}(t) = \{j\}$. If $t = \delta(t_1, ..., t_l)$ with $\delta \in \Delta^{(l)}$, $l \geqslant 0$, and $t_1, ..., t_l \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, then $\mathrm{oc}(t) = \mathrm{oc}(t_1) \cup \cdots \cup \mathrm{oc}(t_l)$. If $t = \langle r, x_i \rangle (t_1, ..., t_l)$ with $\langle r, x_i \rangle \in \langle Q, X_k \rangle^{(l)}$, $l \geqslant 0$, and $t_1, ..., t_l \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, then $\mathrm{oc}(t) = \bigcup \{\mathrm{oc}(t_j) \mid j \in f_i(r)\}$.

For every $(q, I) \in Q'$, $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $(p_1, f_1), ..., (p_k, f_k) \in P'$, let the rule

$$\langle (q, I), \sigma(x_1, ..., x_k) \rangle (y_1, ..., y_{|I|}) \to \zeta \qquad \langle (p_1, f_1), ..., (p_k, f_k) \rangle \tag{$*$}$$

be in R', such that $\zeta = d(y_1, d(y_2, ..., d(y_{|I|-1}, y_{|I|})))$ if $I \neq f_0(q)$, where $(p_0, f_0) = h'_\sigma((p_1, f_1), ..., (p_k, f_k))$; and otherwise $\zeta = \mathrm{rhs}_M(q, \sigma, \langle p_1, ..., p_k \rangle)\, R_1 R_2$, where R_1 denotes the substitution

$$[\![\langle r, x_i \rangle \leftarrow \langle (r, I_r), x_i \rangle (y_{I_r(1)}, ..., y_{I_r(n)}) \mid \langle r, x_i \rangle \in \langle Q, X_k \rangle, I_r = f_i(r), n = |I_r|]\!]$$

and $R_2 = [y_{I(j)} \leftarrow y_j \mid j \in [|I|]]$. This ends the construction of M'.

It is straightforward to show (by induction on the structure of s) that the look-ahead automaton of M' is defined in such a way that for every $s \in T_\Sigma$, if $h'(s) = (p_0, f_0)$, then $p_0 = h(s)$ and $f_0(q) = \{j \mid y_j \text{ occurs in } M_q(s)\}$ for every $q \in Q$. In the induction step it can be shown (using Lemma 3.4) that for $t \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, $\mathrm{oc}(t) = \mathrm{par}(t[\![\cdots]\!])$, where $[\![\cdots]\!] = [\![\langle r, x_i \rangle \leftarrow M_r(s_i) \mid \langle r, x_i \rangle \in \langle Q, X_k \rangle]\!]$ and $\mathrm{par}(t) = \{j \in [m] \mid y_j \text{ occurs in } t\}$.

Let us now show that M' is nondeleting. Consider a rule $(*)$ in R'. (Note that both R_1 and $\mathrm{oc}(t)$ depend on $f_1, ..., f_k$.)

Claim 1. For $t \in T_{\langle Q, X_k \rangle \cup \Delta}(Y_m)$, $\mathrm{par}(t R_1) = \mathrm{oc}(t)$.

This claim can be proved by induction on the structure of t. Since it is very similar to the proof of Claim 1 in the proof of Lemma 6.3 we omit it.

If the $((q, I), \sigma, \langle (p_1, f_1), ..., (p_k, f_k) \rangle)$-rule does not have a dummy right-hand side $d(y_1, ..., d(y_{|I|}))$, then $I = f_0(q) = \mathrm{oc}(\xi)$, where $\xi = \mathrm{rhs}_M(q, \sigma, \langle p_1, ..., p_k \rangle)$. By Claim 1, $\mathrm{par}(\xi R_1) = I$ and, hence, by the definition of R_2, $\mathrm{par}(\zeta) = \mathrm{par}(\xi R_1 R_2) = [m]$ and, thus, M' is nondeleting.

The correctness of the construction follows from Claim 2 by taking $q = q_0$ and $I = \varnothing$.

Claim 2. Let $q \in Q^{(m)}$ and $s \in T_\Sigma$, and let $I = f_0(q)$, where $(p_0, f_0) = h'(s)$. Then $M'_{(q, I)}(s)\, \Theta_I = M_q(s)$, where $\Theta_I = [y_j \leftarrow y_{I(j)} \mid j \in [|I|]]$.

This claim can be proved by induction on the structure of s (similar to the proof of Claim 2 in the proof of Lemma 6.3). We now show that if M is fci, then so is M'. If M is fci, then there is an $N \geqslant 0$ such that N is an input copying bound for M. Let $\hat{M}$ and $\hat{M}'$ be the extensions of M and M', respectively. Let $(p, f) \in P'$ and let $s \in T_\Sigma(\{(p, f)\})$ such that (p, f) occurs at most once in s. The following claim is the "extended" version of Claim 2. Let $q \in Q^{(m)}$, $(p_0, f_0) = \hat{h}'(s)$, and $I = f_0(q)$.

Claim 3. $\hat{M}'_{(q,I)}(s)\,\Theta_I = \hat{M}(s[(p,f)\leftarrow p])\,\Psi$, where Θ_I is as in Claim 2 and $\Psi = [\![\langle\!\langle r, p\rangle\!\rangle \leftarrow \langle\!\langle (r, I_r), (p, f)\rangle\!\rangle(y_{I_r(1)}, ..., y_{I_r(n)}) \mid r \in Q, I_r = f(r), n = |I_r|]\!]$.

This claim can be proved by induction on the structure of s (similar to the proof of Claim 3 in the proof of Lemma 6.3).

Let $s \in T_\Sigma$ and $u \in \mathrm{Occ}(s)$. Then $\mathrm{sts}_{M'}(s, u)$ is the sequence of states which occur in $\hat{M}'_{(q_0,\varnothing)}(s[u\leftarrow(p,f)])$, where $(p,f) = h'(s/u)$. By Claim 3, $\hat{M}'_{(q_0,\varnothing)}(s[u\leftarrow (p,f)])$ equals $\hat{M}_{q_0}(s[u\leftarrow p])\,\Psi$. Since each I_r is a subset of $[m]$, where $m = \mathrm{rank}_Q(r)$, the substitution Ψ can only delete and, hence, the number of occurrences of states in $\hat{M}'_{(q_0,\varnothing)}(s[u\leftarrow(p,f)])$ is less than or equal to the one in $\hat{M}_{q_0}(s[u\leftarrow p])$. Hence, M' has the same input copying bound N as M.

Assume now that M is suri. Thus there is a sur partition $\Pi = \{Q_1, ..., Q_n\}$ and a collection of sur mappings $\mathscr{T}$ for M. Then it is easy to verify that $\Pi' = \{Q'_1, ..., Q'_n\}$ with $Q'_i = \{(q, I) \in Q' \mid q \in Q_i\}$ is a sur partition for M' and that $\mathscr{T}'$ with $\mathscr{T}'_{\sigma,\langle(p_1,f_1),...,(p_k,f_k)\rangle}(Q'_j, i) = Q'_v$ for every $\sigma \in \Sigma^{(k)}$, $j \in [n]$, $i \in [k]$, and (p_1, f_1), ..., $(p_k, f_k) \in P$ with $\mathscr{T}_{\sigma,\langle p_1,...,p_k\rangle}(Q_j, i) = Q_v$ is a collection of sur mappings for M'. In particular, if M is ssuri (i.e., $n = 1$), then so is M'.

If M is fcp then there is a parameter copying bound N' for M. For $(q, I) \in Q'^{(m)}$ and $j \in [m]$, $c_j(M'_{(q,I)}(s)) = c_{I(j)}(M_q(s))$ by Claim 2. Hence, M' has the same parameter copying bound N' as M. Clearly, if M is surp, then so is M'. ∎

We now want to prove that each fci $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{surp}}$ M can be turned into a suri $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{surp}}$ M' which realizes the same translation as M. By Lemma 6.6 we may assume that M is nondeleting. It turns out that the type of second-order substitution inherent in the derivation of an MTT M which is both surp and nondeleting, is rather restricted and as a result the state sequences of M are easy to compute (in a way that is known for top-down tree transducers). In fact, in a derivation step $\langle q, \sigma(s_1, ..., s_k)\rangle(t_1, ..., t_m) \Rightarrow_M \xi$ all the trees $t_1, ..., t_m$ each appear exactly once in ξ (only the order may change). Now consider an input tree s, a node u of s, and the state sequence $\mathrm{sts}_M(s, u) = q_1 \cdots q_n$. Recall that this is the sequence of states which appear in $\hat{M}_{q_0}(s[u\leftarrow h(s/u)])$ (in preorder). Then the state sequence of s at a child ui of u can be computed in the following way: Simply consider all occurrences of elements of $\langle Q, \{x_i\}\rangle$ in $\zeta_v = \mathrm{rhs}(q_v, \sigma, \langle h(s_1), ..., h(s_k)\rangle)$ for all $v \in [n]$. Since M is surp and nondeleting, we know that all these occurrences will appear in $\hat{M}_{q_0}(s[ui\leftarrow h(s/ui)])$; of course, we do not know their precise order. Hence, from $\zeta_1, ..., \zeta_n$ we can compute a permutation of $\mathrm{sts}_M(s, ui)$. We will now formalize this.

First we need an easy lemma. Recall that $c_j(t)$ is the number of occurrences of y_j in t, cf. Definition 6.2.

LEMMA 6.7. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be a nondeleting $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{surp}}$. For every $s \in T_\Sigma$, $q \in Q^{(m)}$, $m \geqslant 1$, and $j \in [m]$, $c_j(M_q(s)) = 1$.

Proof. By induction on the structure of s. Let $s = \sigma(s_1, ..., s_k)$, $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, $s_1, ..., s_k \in T_\Sigma$, and $p_i = h(s_i)$ for $i \in [k]$. Then $c_j(M_q(s)) = c_j(\zeta[\![\cdots]\!])$, where $\zeta = \mathrm{rhs}(q, \sigma, \langle p_1, ..., p_k\rangle)$ and $[\![\cdots]\!] = [\![\langle r, x_i\rangle \leftarrow M_r(s_i) \mid \langle r, x_i\rangle \in \langle Q, X_k\rangle]\!]$. By induction $M_r(s_i)$ contains each parameter exactly once, and so the substitution $[\![\cdots]\!]$ changes only the order of the parameters which occur in ζ. Thus $c_j(\zeta[\![\cdots]\!]) = c_j(\zeta)$, and $c_j(\zeta) = 1$ because M is nondeleting and surp. ∎

We now turn to the computation of state sequences. We first define a syntactic variation of the notion of state sequence (overloading the notation sts_M, cf. Definition 3.7).

DEFINITION 6.8. Let M be an MTT^{R} and Q its set of states. Let $k \geqslant 0$, $\zeta \in T_{\langle Q, X_k \rangle}(Y)$, and $i \in [k]$. The *state sequence of* x_i *in* ζ, denoted by $\text{sts}_M(\zeta, i)$ is the sequence $q_1 \cdots q_n$ of all states in Q such that $\langle q_1, x_i \rangle, ..., \langle q_n, x_i \rangle$ occur in ζ at some occurrences $u_1, ..., u_n$, respectively, and $u_1 < \cdots < u_n$.

LEMMA 6.9. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be a nondeleting MTT^R_{surp}. Let $s \in T_\Sigma$, $u \in \text{Occ}(s)$, $\text{sts}_M(s, u) = q_1 \cdots q_n$, and $s[u] = \sigma \in \Sigma^{(k)}$. Let, for every $j \in [k]$, $p_j = h(s/uj)$ and for every $v \in [n]$, $\zeta_v = \text{rhs}_M(q_v, \sigma, \langle p_1, ..., p_k \rangle)$. Then, for every $i \in [k]$, $\text{sts}_M(\zeta_1, i) \cdots sts_M(\zeta_n, i)$ is a permutation of $\text{sts}_M(s, ui)$.*

Proof. Let $\hat{M}$ be the extension of M. By the definition of state sequences, $\text{sts}_M(s, ui)$ consists of the states $r_1, ..., r_m$ such that $\langle\langle r_1, p_i \rangle\rangle, ..., \langle\langle r_m, p_i \rangle\rangle$ are all elements of $\langle\langle Q, \{p_i\} \rangle\rangle$ that occur in $\hat{M}_{q_0}(s[ui \leftarrow p_i])$. Let us apply Lemma 3.6 to $s[ui \leftarrow p_i]$. We get

$$\hat{M}_{q_0}(s[ui \leftarrow p_i]) = \hat{M}_{q_0}(s[ui \leftarrow p_i][u \leftarrow p]) [\![\langle\langle q', p \rangle\rangle \leftarrow \hat{M}_{q'}(s[ui \leftarrow p_i]/u) \mid q' \in Q]\!]$$

which equals $\hat{M}_{q_0}(s[u \leftarrow p])[\![-]\!]$, where $[\![-]\!]$ is the substitution

$$[\![\langle\langle q', p \rangle\rangle \leftarrow \hat{M}_{q'}(\underbrace{\sigma(s_1, ..., s_{i-1}, p_i, s_{i+1}, ..., s_k)}_{\tilde{s}}) \mid q' \in Q]\!]$$

with $s_j = s/uj$ for $j \in [k] - \{i\}$. We know that $\text{sts}_M(s, u) = q_1 \cdots q_n$ and, hence, that $\langle\langle q_1, p \rangle\rangle, ..., \langle\langle q_n, p \rangle\rangle$ are all elements of $\langle\langle Q, \{p\} \rangle\rangle$ that occur in $\hat{M}_{q_0}(s[u \leftarrow p])$. Moreover, $\hat{M}_{q_0}(s[u \leftarrow p]) \in T_\Delta(\langle\langle Q, \{p\} \rangle\rangle)$. Thus, all elements of $\langle\langle Q, \{p_i\} \rangle\rangle$ in $\hat{M}_{q_0}(s[u \leftarrow p])[\![-]\!]$ can only stem from the substitution $[\![-]\!]$. Since $\hat{M}$ is surp and nondeleting, it follows from Lemma 6.7 that the replacement of an occurrence of $\langle\langle q', p \rangle\rangle$ by $\hat{M}_{q'}(\tilde{s})$ does not delete or introduce new elements of $\langle\langle Q, \{p\} \rangle\rangle$. Therefore, all elements of $\langle\langle Q, \{p_i\} \rangle\rangle$ that occur in $\hat{M}(s[u \leftarrow p])[\![-]\!]$ occur in the trees $\hat{M}_{q_1}(\tilde{s}), ..., \hat{M}_{q_n}(\tilde{s})$. For $v \in [n]$, $\hat{M}_{q_v}(\tilde{s}) = \zeta_v [\![\langle q', x_\mu \rangle \leftarrow \hat{M}_{q'}(s_\mu) \mid \mu \in [k] - \{i\}]\!] \, \Theta$, where $\Theta = [\langle r, x_i \rangle \leftarrow \langle\langle r, p_i \rangle\rangle \mid r \in Q]$. And hence, since $\hat{M}_{q'}(s_\mu) = M_q(s_\mu) \in T_\Delta(Y)$, the sequence of all states that occur in $\hat{M}_{q_v}(\tilde{s})$ is $\text{sts}_M(\zeta_v, i)$. Altogether, this means that $\text{sts}_M(\zeta_1, i) \cdots \text{sts}_M(\zeta_n, i)$ is a permutation of the elements of $\langle\langle Q, \{p_i\} \rangle\rangle$ that appear in $\hat{M}_{q_0}(s[ui \leftarrow p_i])$, i.e., of the elements in $\text{sts}_M(s, ui)$. ∎

In Lemma 5.3 of [vV96] it is proved that $yT_{\text{fc}}(REGT) \subseteq yT_{\text{ssur}}(REGT)$. The idea is to use as states of the ssur top-down tree transducer M' the state sequences of the finite copying top-down tree transducer M (with a bar on one state of the state sequence). We now use the same idea to prove the following more general lemma (see also Corollary 7.5).

LEMMA 6.10. $MTT^R_{\text{fci, surp}} \subseteq MTT^R_{\text{suri, surp}}$.

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\text{MTT}^{\text{R}}_{\text{fci, surp}}$. By Lemma 6.6 we may assume that M is nondeleting. Let N be an input copying bound for M. We want to construct an $\text{MTT}^{\text{R}}_{\text{suri, surp}}$ M' such that $\tau_{M'} = \tau_M$.

Intuitively, in the states of M' we compute the state sequences of M (modulo a permutation); if the state sequence of s at u (of M) is $w = q_1 \cdots q_n$, then the corresponding state sequence of M' is $q'_1 \cdots q'_n$ with $q'_j = (q_1 \cdots \bar{q}_j \cdots q_n)$. Thus, the particular state q_j of M processing s/u is marked by a bar in the corresponding state $q_1 \cdots \bar{q}_j \cdots q_n$ of M'. In this way, no state of M' appears more than once in a state sequence of M (cf. the remark following Definition 5.7). Since M' can compute the state sequences of M in a top-down fashion as shown in Lemma 6.9, M' will be suri.

For a string $w = q_1 \cdots q_n \in Q^*$ and $j \in [n]$, let w^j denote the string $q_1 \cdots q_{j-1} \bar{q}_j q_{j+1} \cdots q_n$ obtained from w by "marking" the jth element q_j, and let $\text{Marked}(w) = \{w^j \mid j \in [n]\}$. For $v \in [n]$, we write $\text{sts}^v_M(\zeta, i)$ instead of $\text{sts}_M(\zeta, i)^v$.

We define $M' = (Q', P, \Sigma, \Delta, \bar{q}_0, R', h)$, where $Q' = \{q_1 \cdots \bar{q}_j \cdots q_n \mid q_1, ..., q_n \in Q, n \in [N], j \in [n]\}$ with $\text{rank}_{Q'}(q_1 \cdots \bar{q}_j \cdots q_n) = \text{rank}_Q(q_j)$. For $q_1 \cdots \bar{q}_j \cdots q_n \in Q'^{(m)}$, $\sigma \in \Sigma^{(k)}$, and $p_1, ..., p_k \in P$, let the rule

$$\langle q_1 \cdots \bar{q}_j \cdots q_n, \sigma(x_1, ..., x_k)\rangle(y_1, ..., y_m) \to \zeta \qquad \langle p_1, ..., p_k\rangle$$

be in R', where $\zeta = \text{dummy} \in \Delta^{(0)}$, if there is an $i \in [k]$ such that the length of $\text{sts}_M(\zeta_1, i) \cdots \text{sts}_M(\zeta_n, i)$ is greater than N, where $\zeta_\mu = \text{rhs}_M(q_\mu, \sigma, \langle p_1, ..., p_k\rangle)$ for $\mu \in [n]$; otherwise ζ is obtained from $\text{rhs}_M(q_j, \sigma, \langle p_1, ..., p_k\rangle)$ by replacing, for every $i \in [k]$, the vth occurrence of $\langle q, x_i\rangle$ (with $q \in Q$) by $\langle \text{sts}_M(\zeta_1, i) \cdots \text{sts}^v_M(\zeta_j, i) \cdots \text{sts}_M(\zeta_n, i), x_i\rangle$. Note that this string $\text{sts}_M(\zeta_1, i) \cdots \text{sts}^v_M(\zeta_j, i) \cdots \text{sts}_M(\zeta_n, i)$ is of the form $w\bar{q}w'$ with $w, w' \in Q^*$.

An example of this construction is given in Example 6.11. Let us now prove the correctness of M'. For a sequence $q_1 \cdots q_n$ of states in Q and $s \in T_\Sigma$ we say that $q_1 \cdots q_n$ *occurs on* s, if there is an $\tilde{s} \in T_\Sigma$ and a $u \in \text{Occ}(\tilde{s})$ such that $\tilde{s}/u = s$ and $q_1 \cdots q_n$ is a permutation of $\text{sts}_M(\tilde{s}, u)$. Since the sequence q_0 occurs on s for every $s \in T_\Sigma$, the following claim proves the correctness of M'.

Claim. Let $q_1, ..., q_n \in Q$ and $s \in T_\Sigma$ such that $q_1 \cdots q_n$ occurs on s. Then for every $j \in [n]$, $M'_{q_1 \cdots \bar{q}_j \cdots q_n}(s) = M_{q_j}(s)$.

Proof. By induction on the structure of s. Let $s = \sigma(s_1, ..., s_k)$, $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_\Sigma$. For $i \in [k]$ let $p_i = h(s_i)$, and for $j \in [n]$ let $\zeta_j = \text{rhs}_M(q_j, \sigma, \langle p_1, ..., p_k\rangle)$. By Lemma 3.4, $M'_{q_1 \cdots \bar{q}_j \cdots q_n}(s) = \text{rhs}_{M'}(q_1 \cdots \bar{q}_j \cdots q_n, \sigma, \langle p_1, ..., p_k\rangle)[\![\cdots]\!]$, where $[\![\cdots]\!]$ denotes the substitution $[\![\langle w, x_i\rangle \leftarrow M'_w(s_i) \mid \langle w, x_i\rangle \in \langle Q', X_k\rangle]\!]$. Since $q_1 \cdots q_n$ occurs on s and M is nondeleting and surp, we know by Lemma 6.9 that for every $i \in [k]$, $\text{sts}_M(\zeta_1, i) \cdots \text{sts}_M(\zeta_n, i)$ occurs on s_i. Therefore, the $(q_1 \cdots \bar{q}_j \cdots q_n, \sigma, \langle p_1, ..., p_k\rangle)$-rule has a non-dummy right-hand side. Hence, $M'_{q_1 \cdots \bar{q}_j \cdots q_n}(s)$ equals $\xi\Theta[\![\cdots]\!]$, where $\xi = \text{rhs}_M(q_j, \sigma, \langle p_1, ..., p_k\rangle)$ and Θ denotes the replacement of the vth occurrence of $\langle q, x_i\rangle$ by $\langle \text{sts}_M(\zeta_1, i) \cdots \text{sts}^v_M(\zeta_j, i) \cdots \text{sts}_M(\zeta_n, i), x_i\rangle$. We can apply the induction hypothesis and obtain that $M'_{w\bar{q}w'}(s_i)$ with $wqw' = \text{sts}_M(\zeta_1, i) \cdots \text{sts}_M(\zeta_n, i)$ equals $M_q(s_i)$, because $\text{sts}_M(\zeta_1, i) \cdots \text{sts}_M(\zeta_n, i)$ occurs on s_i. If we now combine the substitutions Θ and $[\![\cdots]\!]$, then we get $\xi\Theta'$, where Θ' denotes the second-order substitution of replacing every occurrence of $\langle q, x_i\rangle$ by $M_q(s_i)$ for $\langle q, x_i\rangle \in \langle Q, X_k\rangle$. This equals $\xi[\![\langle q, x_i\rangle \leftarrow M_q(s_i) \mid \langle q, x_i\rangle \in \langle Q, X_k\rangle]\!] = M_{q_j}(s)$.

Since R' consists of dummy-rules and rules which are obtained from rules of M by renaming of states, M' is surp. It remains to show that M' is suri. Let $\Pi = \{\text{Marked}(w) \mid w \in Q^*,\ |w| \in [N]\}$ and for $q_1 \cdots q_n \in Q^*$ with $n \in [N]$, $\sigma \in \Sigma^{(k)}$, $p_1, ..., p_k \in P$, and $i \in [k]$, let

$$\mathscr{T}_{\sigma, \langle p_1, ..., p_k \rangle}(\text{Marked}(q_1 \cdots q_n), i) = \text{Marked}(\text{sts}_M(\zeta_1, i) \cdots \text{sts}_M(\zeta_n, i)),$$

where $\zeta_\mu = \text{rhs}_M(q_\mu, \sigma, \langle p_1, ..., p_k \rangle)$ for $\mu \in [n]$. It follows directly from the definition of the rules of M' that Π is a sur partition for M' and that $\mathscr{T}$ is a collection of sur mappings for M'. ∎

The following example illustrates the construction in the proof of Lemma 6.10.

EXAMPLE 6.11. We consider a finite copying top-down tree transducer, without regular look-ahead. Let $M = (Q, \Sigma, \Delta, q_0, R)$ with $Q = Q^{(0)} = \{q_0, q_1, q_2\}$, $\Sigma = \{\gamma^{(1)}, \gamma'^{(1)}, \alpha^{(0)}\}$, $\Delta = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}\}$, and R consisting of the rules:

$$\begin{array}{lll}
\langle q_\nu, \gamma(x_1)\rangle \to \gamma(\langle q_\nu, x_1\rangle) & \text{for} & \nu \in \{0, 2\} \\
\langle q_1, \gamma(x_1)\rangle \to \sigma(\langle q_2, x_1\rangle, \langle q_2, x_1\rangle) & & \\
\langle q_0, \gamma'(x_1)\rangle \to \sigma(\langle q_1, x_1\rangle, \langle q_0, x_1\rangle) & & \\
\langle q_\nu, \gamma'(x_1)\rangle \to \alpha & \text{for} & \nu \in \{1, 2\} \\
\langle q_\nu, \alpha\rangle \to \alpha & \text{for} & \nu \in \{0, 1, 2\}
\end{array}$$

Using Lemma 6.9, it is straightforward to verify that M is fci and $N = 3$ is an input copying bound for M. However, M is not suri (just consider the (q_1, γ)-rule).

We now construct a suri top-down tree transducer M' which realizes the same translation as M, following the construction in the proof of Lemma 6.10. Let $M' = (Q', \Sigma, \Delta, \bar{q}_0, R')$ with $Q' = Q'^{(0)} = \{r_1 \cdots \bar{r}_j \cdots r_n \mid r_1, ..., r_n \in Q,\ n \in [3],\ j \in [n]\}$. We only show the rules in R' which will actually be used in derivations by M':

$$\begin{array}{l}
\langle \bar{q}_0, \gamma(x_1)\rangle \to \gamma(\langle \bar{q}_0, x_1\rangle) \\
\langle \bar{q}_1 q_0, \gamma(x_1)\rangle \to \sigma(\langle \bar{q}_2 q_2 q_0, x_1\rangle, \langle q_2 \bar{q}_2 q_0, x_1\rangle) \\
\langle q_1 \bar{q}_0, \gamma(x_1)\rangle \to \gamma(\langle q_2 q_2 \bar{q}_0, x_1\rangle) \\
\langle w, \gamma(x_1)\rangle \to \gamma(\langle w, x_1\rangle) \quad \text{for} \quad w \in \text{Marked}(q_2 q_2 q_0) \\
\langle \bar{q}_0, \gamma'(x_1)\rangle \to \sigma(\langle \bar{q}_1 q_0, x_1\rangle, \langle q_1 \bar{q}_0, x_1\rangle) \\
\langle \bar{q}_1 q_0, \gamma'(x_1)\rangle \to \alpha \\
\langle q_1 \bar{q}_0, \gamma'(x_1)\rangle \to \sigma(\langle \bar{q}_1 q_0, x_1\rangle, \langle q_1 \bar{q}_0, x_1\rangle) \\
\langle \bar{q}_2 q_2 q_0, \gamma'(x_1)\rangle \to \alpha \\
\langle q_2 \bar{q}_2 q_0, \gamma'(x_1)\rangle \to \alpha \\
\langle q_2 q_2 \bar{q}_0, \gamma'(x_1)\rangle \to \sigma(\langle \bar{q}_1 q_0, x_1\rangle, \langle q_1 \bar{q}_0, x_1\rangle)
\end{array}$$

and the right-hand side of each α-rule of M' equals α. Obviously, M' is suri and its sur partition contains Marked(q_0), Marked(q_1q_0), and Marked($q_2q_2q_0$). Consider the input tree $s=\gamma'\gamma\gamma\gamma'\alpha$. As the reader may verify, $t=\tau_M(s)=\sigma(\sigma(\gamma\alpha, \gamma\alpha), \gamma\gamma\sigma(\alpha, \alpha))$. The state sequences of M are $\mathrm{sts}_M(s, \varepsilon)=q_0$, $\mathrm{sts}_M(s, 1)=q_1q_0$, $\mathrm{sts}_M(s, 11)=q_2q_2q_0$, $\mathrm{sts}_M(s, 111)=q_2q_2q_0$, and $\mathrm{sts}_M(s, 1111)=q_1q_0$. To illustrate the claim in the proof of Lemma 6.10, consider the corresponding derivation of M':

$$\begin{aligned}\langle \bar{q}_0, s\rangle &\Rightarrow_{M'} \sigma(\langle \bar{q}_1q_0, \gamma\gamma\gamma'\alpha\rangle, \langle q_1\bar{q}_0, \gamma\gamma\gamma'\alpha\rangle)\\ &\Rightarrow^*_{M'} \sigma(\sigma(\langle \bar{q}_2q_2q_0, \gamma\gamma'\alpha\rangle, \langle q_2\bar{q}_2q_0, \gamma\gamma'\alpha\rangle), \gamma(\langle q_2q_2\bar{q}_0, \gamma\gamma'\alpha\rangle))\\ &\Rightarrow^*_{M'} \sigma(\sigma(\gamma(\langle \bar{q}_2q_2q_0, \gamma'\alpha\rangle), \gamma(\langle q_2\bar{q}_2q_0, \gamma'\alpha\rangle)), \gamma\gamma(\langle q_2q_2\bar{q}_0, \gamma'\alpha\rangle))\\ &\Rightarrow^*_{M'} \sigma(\sigma(\gamma\alpha, \gamma\alpha), \gamma\gamma\sigma(\langle \bar{q}_1q_0, \alpha\rangle, \langle q_1\bar{q}_0, \alpha\rangle))\\ &\Rightarrow^*_{M'} t.\end{aligned}$$

We are now ready to prove our second main result.

THEOREM 6.12. $MTT^R_{\mathrm{fc}} = MTT^R_{\mathrm{sur}}$.

Proof. $MTT^R_{\mathrm{fc}} \subseteq MTT^R_{\mathrm{sur}}$ holds because $MTT^R_{\mathrm{fc}} = MTT^R_{\mathrm{fci,\,fcp}} \subseteq MTT^R_{\mathrm{fci,\,surp}}$ (by Lemma 6.3) and $MTT^R_{\mathrm{fci,\,surp}} \subseteq MTT^R_{\mathrm{suri,\,surp}} = MTT^R_{\mathrm{sur}}$ (by Lemma 6.10).

Hence it remains to show that $MTT^R_{\mathrm{sur}} \subseteq MTT^R_{\mathrm{fc}}$. In fact we will show that every $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{sur}}$ is an $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{fc}}$. Let $M=(Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{suri,\,surp}}$. By Lemma 6.6 we may assume that M is nondeleting. Let Π be a sur partition for M.

We first show that M is fci. From the next claim it follows that $N=\max(|\bar{Q}| \mid \bar{Q}\in\Pi)$ is an input copying bound for M.

Claim. Let $s\in T_\Sigma$ and $u\in \mathrm{Occ}(s)$. Then

(i) $\{q\in Q \mid q$ occurs in $\mathrm{sts}_M(s, u)\} \subseteq \bar{Q}$ for some $\bar{Q}\in\Pi$ and

(ii) no state appears more than once in $\mathrm{sts}_M(s, u)$.

Proof. By induction on u. For $u=\varepsilon$, $\mathrm{sts}_M(s, \varepsilon)=q_0$ and thus (i) and (ii) hold. Let $\mathrm{sts}_M(s, u)=q_1\cdots q_n$. By the induction hypothesis, $q_1, ..., q_n\in\bar{Q}$ for some $\bar{Q}\in\Pi$ and $q_1, ..., q_n$ are pairwise different. By Lemma 6.9, $\mathrm{sts}_M(s, ui)$ is a permutation of $w=\mathrm{sts}_M(\zeta_1, i)\cdots \mathrm{sts}_M(\zeta_n, i)$, where $\zeta_\nu=\mathrm{rhs}_M(q_\nu, \sigma, \langle p_1, ..., p_k\rangle)$ for $\nu\in[n]$, $s[u]=\sigma\in\Sigma^{(k)}$, and $p_j=h(s/uj)$ for $j\in[k]$. By the definition of suri, all states in w are in $\mathscr{T}_{\sigma, \langle p_1, ..., p_k\rangle}(\bar{Q}, i)\in\Pi$ which proves (i) for ui. Since M is ssuri with respect to $\bar{Q}$, (ii) holds for ui.

Since M is nondeleting and surp, it follows from Lemma 6.3 that M is fcp and that $N'=1$ is a parameter copying bound for M. ∎

7. MAIN RESULT AND CONSEQUENCES

In [BE98] it is shown that the class $MSOTT$ of tree translations definable by monadic second-order logic is equal to the class ATT^R_{sur} (cf. the Introduction). By the results presented in this paper we can now give a characterization of the class $MSOTT$ in terms of macro tree transducers. By the results of Section 5, the class $MSOTT$ is equal to the class $MTT^{\mathrm{R}}_{\mathrm{sur}}$ of translations realized by single use restricted macro tree transducers with regular look-ahead. By the results of Section 6, this class is equal to the class $MTT^{\mathrm{R}}_{\mathrm{fc}}$ of translations realized by finite copying macro tree transducers with regular look-ahead.

From Theorems 5.14 and 6.12 we obtain the main result.

THEOREM 7.1. $MSOTT = ATT^{R}_{\mathrm{sur}} = MTT^{R}_{\mathrm{sur}} = MTT^{R}_{\mathrm{fc}}$.

Every MSO definable tree translation can be realized by an MTT, because MTTs are closed under regular look-ahead (Theorem 4.21 of [EV85]); i.e., $MTT^R \subseteq MTT$. Note, however, that the construction of $MTT^R \subseteq MTT$ does not preserve the finite copying property (cf. also the part on "weak finite copying" later in this section).

COROLLARY 7.2. $MSOTT \subseteq MTT$.

The inclusion is proper because for an MSO definable tree translation the size of the output tree is linear in the size of the input tree (and already a top-down tree transducer can translate a monadic tree of size n into the full binary tree of size $2^n - 1$).

As discussed in the Introduction, it is known from [EvO97, CE95] that the class of context-free graph languages (see, e.g., [Eng97]) can be obtained by applying MSO graph transductions to regular tree languages. Hence, the class of tree languages generated by context-free graph grammars equals the class $MSOTT$ $(REGT)$ of MSO definable tree translations applied to regular tree languages. In fact, this holds for both well-known types of context-free graph grammars, namely, hyperedge replacement (HR) and node replacement (NR), because the classes of tree languages they generate coincide (cf. Section 6 of [Eng97]). Since $REGT$ is closed under T^R-REL (cf. Corollary IV.6.7 in [GS84]) it follows from Theorem 5.10 that $MTT^R_{\mathrm{sur}}(REGT) = MTT_{\mathrm{ssur}}(REGT)$. It is also well known (see, e.g., Proposition 18.1 of [GS97]) that regular look-ahead can be simulated by a relabeling of the input tree, and hence, $MTT^R_{\mathrm{fc}}(REGT) = MTT_{\mathrm{fc}}(REGT)$. Thus, applying Theorem 7.1 we obtain the next corollary (cf. Corollary 19 of [BE98]).

COROLLARY 7.3. *$MTT_{\mathrm{ssur}}(REGT) = MTT_{\mathrm{fc}}(REGT)$ is the class of tree languages generated by* (HR *or* NR) *context-free graph grammars.*

In [Dre97] Drewes shows that the class of tree languages generated by context-free graph grammars can be obtained by evaluating the output tree languages of finite-copying top-down tree transducers in an algebra of (hyper)graphs in which each operation is a substitution into a tree graph. In [EM] it is shown that this type of evaluation can be carried out by macro tree transducers which are simple (i.e., linear and nondeleting) both in the input variables and in the parameters. This means that Drewes' result is equivalent with a similar characterization as Corollary 7.3 of the class of tree languages generated by context-free graph grammars, namely, that it is the class $MTT_{\mathrm{fci,\,sp}}(REGT)$ of output tree languages of macro tree transducers which are fci and simple in the parameters (i.e., surp and nondeleting), taking regular tree languages as input. This characterization follows from Corollary 7.3 by Lemmas 6.3 and 6.6.

From [BE98] it is known that the class $MSOTT_{\mathrm{dir}}$ of so-called direction-preserving MSO tree translations is equal to the class $ATT^R_{\mathrm{os,\,sur}}$ of tree translations which can be realized by sur ATTs that have only synthesized attributes. It is well known that ATTs with synthesized attributes only correspond to top-down tree transducers (see, e.g., [CF82, Fül81]). This means that $ATT^R_{\mathrm{os}} = T^R$. We now consider the influence of the sur property.

THEOREM 7.4. $MSOTT_{\mathrm{dir}} = ATT^R_{\mathrm{os,\,sur}} = T^R_{\mathrm{sur}} = T^R_{\mathrm{fc}}$.

Proof. $MSOTT_{\mathrm{dir}} = ATT^R_{\mathrm{os,\,sur}}$ by Theorem 17 of [BE98].

Let us now show that $ATT^R_{\mathrm{os,\,sur}} \subseteq T^R_{\mathrm{sur}}$. By definition and Theorem 4.4, $ATT^R_{\mathrm{os,\,sur}} = ATT\text{-}REL \circ ATT_{\mathrm{os,\,sur}} = T^R\text{-}REL \circ ATT_{\mathrm{os,\,sur}}$. By the proof of Lemma 5.1 it follows that $ATT_{\mathrm{os,\,sur}} \subseteq T^R_{\mathrm{sur}}$ (in fact, even T_{sur}, because the regular look-ahead is not needed). Hence, $ATT^R_{\mathrm{os,\,sur}} \subseteq T^R\text{-}REL \circ T^R_{\mathrm{sur}}$. The latter equals T^R_{sur} because T^R_{sur} is closed under left composition with $T^R\text{-}REL$, which follows from $T^R_{\mathrm{sur}} = T^R\text{-}REL \circ T_{\mathrm{ssur}}$, as mentioned at the end of Section 5.2, together with Lemma 4.5.

From the proof of Lemma 5.12 it follows that $T_{\mathrm{ssur}} \subseteq ATT_{\mathrm{os,\,sur}}$. Hence, together with $T^R_{\mathrm{sur}} = T^R\text{-}REL \circ T_{\mathrm{ssur}}$, we get $T^R_{\mathrm{sur}} \subseteq ATT^R_{\mathrm{os,\,sur}}$ and therefore $ATT^R_{\mathrm{os,\,sur}} = T^R_{\mathrm{sur}}$.

From the proofs of Lemma 6.10 and Theorem 6.12 it follows immediately that $T^R_{\mathrm{sur}} = T^R_{\mathrm{fc}}$. ∎

If we apply the classes in Theorem 7.4 to $REGT$, then we obtain the corollary.

COROLLARY 7.5. $MSOTT_{\mathrm{dir}}(REGT) = ATT_{\mathrm{os,\,sur}}(REGT) = T_{\mathrm{ssur}}(REGT) = T_{\mathrm{fc}}(REGT)$.

In [Rao97] a special type of tree grammar is investigated which generates tuples of trees. It is straightforward to see that these grammars correspond to sur ATTs which have synthesized attributes only (cf. [vV96]). In [Rao97] it is also proved that $T_{\mathrm{fc}}(REGT)$ is the class of tree languages which they generate, i.e., $ATT_{\mathrm{os,\,sur}}(REGT)$. For the corresponding classes of yield languages this equivalence is proved in Section 4 of [vV96] and also in [Wei92], using different formalisms. It is also easy to see (cf. [vV96]) that this is the class of multiple context-free languages of [SMFK91]. These equivalences are discussed in more detail in Section 6 of [Eng97] (where the tree grammars of [Rao97] are called multiple regular tree grammars).

Instead of taking regular tree languages as input, we can also consider the classes of output tree languages of these tree transducers, i.e., the special case of taking the regular tree language T_Σ of all trees over Σ as input. For a class X of tree transducers we denote by $OUT(X)$ the class of output tree languages generated by X. It is shown in Theorem 4.7 of [Man98] that every regular tree language is the output tree language of a "semi-relabeling." Since every semi-relabeling is a strongly sur top-down tree transducer (cf. the discussion in the beginning of Section 5 in

[Man98]), it follows that $REGT \subseteq OUT(T_{\text{ssur}})$. This implies that for every class X of translations which is closed under left composition with T_{ssur}, $OUT(X) = X(REGT)$. Examples of such classes are T_{ssur}, T_{fc}, MTT_{ssur}, and MTT_{fc} (for closure of T_{ssur} under composition see, e.g., Satz 6.5 in [Küh97]). Thus, Corollaries 7.3 and 7.5 also hold for the corresponding output tree languages.

Let $MSOTS$ denote the class of *MSO definable tree-to-string translations*. It is easy to see that there is an MSO transducer which translates a monadic tree $t = a_1(a_2(\cdots a_n(e)))$ into the string $a_1 \cdots a_n$. This injective translation is denoted by p and is, as for yield (cf. Section 2.3), extended to translations and classes of translations. Conversely, there is an MSO transducer which defines p^{-1}, i.e., translates a string $a_1 \cdots a_n$ into the monadic tree $a_1(\cdots a_n(e))$ (recall from Section 2.3 that e is the special symbol with $ye = \varepsilon$). Since MSO graph transductions are closed under composition we get that $MSOTS = pMSOTT_{\text{mon}}$, where $MSOTT_{\text{mon}}$ denotes the class of MSO definable tree translations with monadic output alphabet. Let us now prove that this class equals yT^R_{fc}. For a class X of translations we use the subscript "mon" to denote the restriction of X to translations with monadic output alphabets.

Note that an alternative proof of $pMTT^R_{\text{mon}} = yT^R$ can be found in Theorem 8.7(a) of [EV88].

LEMMA 7.6. *$pMTT^R_{\text{mon}} = yT^R$ and $pMTT^R_{\text{fc, mon}} = yT^R_{\text{fc}}$.*

Proof. Let M be a $\mathrm{T^R}$ and let M' be an $\mathrm{MTT^R_{mon}}$. We may assume that q_0 does not appear in any right-hand side of a rule of M. Furthermore, we may assume that M' is nondeleting and has no state of rank greater than one (to see this, just apply Lemma 6.6 and remove the output symbol d and all states of rank greater than one). Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ and $M' = (Q', P', \Sigma, \Delta', q_0', R', h')$, with $(\Delta')^{(0)} = \{e\}$. We say that M *is related to* M', if $Q = \{q^{(0)} \mid q \in Q'\}$, $P' = P$, $q_0' = q_0$, $h' = h$, and for every $q \in Q$, $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $p_1, ..., p_k \in P$,

$$y(\text{rhs}_M(q, \sigma, \langle p_1, ..., p_k \rangle)) = \pi(\text{rhs}_{M'}(q, \sigma, \langle p_1, ..., p_k \rangle)),$$

where for every $t \in T_{\langle Q', X_k \rangle \cup \Delta'}(Y_1)$, $\pi(t)$ is recursively defined as follows. If $t = y_1$ or $t = e$, then $\pi(t) = \varepsilon$. If $t \in \langle Q', X_k \rangle^{(0)}$, then $\pi(t) = t$. If $t = \delta(t_1)$ with $\delta \in (\langle Q', X_k \rangle \cup \Delta')^{(1)}$ and $t_1 \in T_{\langle Q', X_k \rangle \cup \Delta'}(Y_1)$, then $\pi(t)$ is the string $\delta\pi(t_1)$. Note that $\pi t = pt$ for $t \in T_{\Delta'}$.

Let M be related to M'. Then $y\tau_M(s) = p\tau_{M'}(s)$ for every $s \in T_\Sigma$. This follows from the following claim (with $q = q_0$).

Claim. For every $q \in Q$ and $s \in T_\Sigma$, $yM_q(s) = pM'_q(s)$.

Proof. By induction on s. Let $s = \sigma(s_1, ..., s_k)$ with $\sigma \in \Sigma^{(k)}$, $k \geqslant 0$, and $s_1, ..., s_k \in T_\Sigma$. Then $pM'_q(s) = p(\zeta[\![\cdots]\!])$ with $\zeta = \text{rhs}_{M'}(q, \sigma, \langle p_1, ..., p_k \rangle)$, $p_i = h(s_i)$, and $[\![\cdots]\!] = [\![\langle q', x_i \rangle \leftarrow M'_{q'}(s_i) \mid \langle q', x_i \rangle \in \langle Q', X_k \rangle]\!]$. If we move p inside the substitution $[\![\cdots]\!]$ and apply the induction hypothesis, then we get $\pi(\zeta)[-]$ with $[-] = [\langle q', x_i \rangle \leftarrow yM_q(s_i) \mid \langle q', x_i \rangle \in \langle Q, X_k \rangle]$, because $Q = \{q^{(0)} \mid q \in Q'\}$ and second-order substitution of monadic trees corresponds to ordinary substitution of their

π-translations. Since M is related to M' this equals $y\xi[-]$ with $\xi = \text{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$. By moving the yield outside of $[\![-]\!]$ we obtain $yM_q(s)$.

It is easy to see that the extended version of this claim, i.e., $y\hat{M}_q(s[u \leftarrow p]) = p\hat{M}'_q(s[u \leftarrow p])$ can be proved similarly (where p is extended to $T_{\langle\!\langle Q, \{p\}\rangle\!\rangle \cup \Delta'}$ in the obvious way) and hence, M is fc if and only if M' is fc.

We now show that for every T^{R} M there is a related MTT^{R} M' and vice versa. Let M and M' be as above. If M is given, define $Q' = \{q_0{}^{(0)}\} \cup \{q^{(1)} \mid q \in Q - \{q_0\}\}$, $\Delta' = \{\delta^{(1)} \mid \delta \in \Delta^{(0)}\} \cup \{e^{(0)}\}$, and for every $q \in Q$, $\sigma \in \Sigma^{(k)}$, and $p_1, ..., p_k \in P$, let $\text{rhs}_{M'}(q, \sigma, \langle p_1, ..., p_k\rangle)$ equal $p^{-1}(y\zeta)[e \leftarrow y_1]$ if $q \neq q_0$ and, otherwise, it equals $p^{-1}(y\zeta)$, where $\zeta = \text{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$. Conversely, if M' is given, define $Q = \{q^{(0)} \mid q \in Q'\}$, $\Delta = \{d^{(2)}, e^{(0)}\} \cup \{\delta^{(0)} \mid \delta \in \Delta'^{(1)}\}$, and for every $q \in Q$, $\sigma \in \Sigma^{(k)}$, and $p_1, ..., p_k \in P$, let $\text{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$ be any tree $t \in T_{\langle Q, X_k\rangle \cup \Delta}$ with $yt = \pi(\zeta)$, where $\zeta = \text{rhs}_{M'}(q, \sigma, \langle p_1, ..., p_k\rangle)$. Obviously, M is related to M' for both definitions. ▮

Since $MSOTS = pMSOTT_{\text{mon}}$ and, by Theorem 7.1, $MSOTT = MTT^R_{\text{fc}}$, we get $MSOTS = pMTT^R_{\text{fc, mon}}$. Together with Lemma 7.6, we obtain the characterization of the class of MSO definable tree-to-string translations as a corollary.

THEOREM 7.7. $MSOTS = yT^R_{\text{fc}}$.

It follows from [EvO97, CE95] that the class of string languages generated by context-free graph grammars equals the class $MSOTS(REGT)$ of MSO definable tree-to-string translations applied to regular tree languages. As a corollary we get a result known from [EH91] (cf. Section 6 of [Eng97]).

COROLLARY 7.8. $yT_{\text{fc}}(REGT)$ is the class of string languages generated by (HR or NR) context-free graph grammars.

It is easy to see (cf. Example 1(6, yield) of [BE98]), that there is an MSO transducer which translates a tree t into its yield yt. It is also easy to see, that there is an MSO transducer which translates a string w into a tree t such that the yield of t equals w. It simply translates a string $a_1 \cdots a_n$ into the tree $\sigma(a_1, \sigma(a_2, ..., \sigma(a_n, e)))$. Since MSO transductions are closed under composition, this means that $MSOTS = yMSOTT$. By Theorems 7.1 and 7.7 we obtain the corollary.

COROLLARY 7.9. $yMTT^R_{\text{fc}} = yT^R_{\text{fc}}$.

If we consider "attributed tree-to-string transducers" (ATS transducers), i.e., attribute grammars in which all attribute values are strings and the only semantic operation is concatenation of strings, then, in view of Theorems 7.1 and 7.4, Corollary 7.9 can be formulated as $ATS^R_{\text{sur}} = ATS^R_{\text{os, sur}}$, where ATS denotes the class of tree-to-string translations realized by ATS transducers. Hence, for sur ATS with look-ahead, the presence of inherited attributes has no influence on the translational power.

COROLLARY 7.10. $yMTT_{\text{fc}}(REGT) = yT_{\text{fc}}(REGT)$.

Note that $T_{\text{fc}}(REGT)$ is properly included in $MTT_{\text{fc}}(REGT)$, e.g., the monadic tree language $\{a^n(b^n(e)) \mid n \geqslant 0\}$ over the ranked alphabet $\{a^{(1)}, b^{(1)}, e^{(0)}\}$ is in

$MTT_{\mathrm{fc}}(REGT)$ but not even in $T(REGT)$ because the monadic tree languages in $T(REGT)$ are regular; this was mentioned by mistake as an open problem in Section 6 of [Eng97]. Note also that $yT(REGT)$ is properly included in $yMTT(REGT)$; as an example, the language $\{(a^n b)^{2^n} \mid n \geqslant 1\}$ is obviously in $yMTT(REGT)$, but not in $yT(REGT)$ as shown in Theorem 3.16 of [Eng82].

Weak finite copying. Our definition of finite copying for MTT$^{\mathrm{R}}$s is a generalization of the one for top-down tree transducers. It does not distinguish states which contribute to the output from those which do not (due to deletion or erasing). A more appropriate, but rather technical notion of finite copying does not consider states which do not contribute to the output. We now briefly discuss this possibility and show that the corresponding translations can still be realized be finite copying MTT$^{\mathrm{R}}$s.

Consider an MTT$^{\mathrm{R}}$ $M=(Q, P, \Sigma, \Delta, q_0, R, h)$ and a tree $s \in T_\Sigma$ with $u \in \mathrm{Occ}(s)$ and $p=h(s/u)$. Let $\xi = \hat{M}_{q_0}(s[u \leftarrow p])$ and $v \in \mathrm{Occ}(\xi)$ with $\xi[v] = \langle\!\langle q, p\rangle\!\rangle$ for some $q \in Q$. There are two ways in which this occurrence of $\langle\!\langle q, p\rangle\!\rangle$ will not contribute to the output: (i) v is *deleted*, i.e., v has a prefix $v'i$ such that $\xi[v'] = \langle\!\langle q', p\rangle\!\rangle$ and y_i does not occur in $M_{q'}(s/u)$ or (ii) v is *erased*, i.e., $M_q(s/u) \in Y$. We say that v is *productive*, if v is neither deleted nor erased. We now define a notion of finite copying which distinguishes productive from non-productive occurrences of $\langle\!\langle q, p\rangle\!\rangle$. An MTT$^{\mathrm{R}}$ M is *weak finite copying in the input* (for short, wfci) if there is an $N \in \mathbb{N}$ such that for every input tree s and $u \in \mathrm{Occ}(s)$, $|\mathrm{psts}_M(s, u)| \leqslant N$, where $\mathrm{psts}_M(s, u)$ equals $\mathrm{sts}_M(s, u)$ restricted to states which occur at productive nodes of $\hat{M}_{q_0}(s[u \leftarrow h(s/u)])$. The MTT$^{\mathrm{R}}$ M is *weak finite copying* (for short, wfc) if it is wfci and fcp. We use the abbreviations wfci and wfc as subscripts for classes of translations realized by the corresponding MTTs.

Consider now a wfc MTT$^{\mathrm{R}}$. Is there a finite copying MTT$^{\mathrm{R}}$ which realizes the same translation as M? In other words, is it possible to remove all nonproductive occurrences? It is straightforward to prove that the construction of a nondeleting MTT$^{\mathrm{R}}$ (Lemma 6.6) preserves wfci. Hence, we can remove deleted occurrences. It remains to remove erasing occurrences. This can be done quite similarly to the nondeleting normal form. An MTT$^{\mathrm{R}}$ M is *nonerasing* if for every $q \in Q$, $\sigma \in \Sigma^{(k)}$, and $p_1, ..., p_k \in P$, $\mathrm{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle) \notin Y$. Clearly, if M is nonerasing then $M_q(s) \notin Y$ for all $q \in Q$ and $s \in T_\Sigma$.

LEMMA 7.11. *For every* MTT^R *M there is a nonerasing nondeleting* MTT^R *M′ such that* $\tau_{M'} = \tau_M$. *The construction involved preserves fcp, and if M is wfci then M′ is fci.*

Proof. Let $M=(Q, P, \Sigma, \Delta, q_0, R, h)$. By Lemma 6.6 (and the above remark) we may assume that M is nondeleting. Hence, if the right-hand side ζ of some q-rule is in Y, then $q \in Q^{(1)}$ and $\zeta = y_1$. Define $M' = (Q, P', \Sigma, \Delta, q_0, R', h')$, where $P' = \{(p, S) \mid p \in P, S \subseteq Q^{(1)}\}$. For every $q \in Q^{(m)}$, $\sigma \in \Sigma^{(k)}$, $k, m \geqslant 0$, and $(p_1, S_1), ..., (p_k, S_k) \in P'$ let $\xi_q = \mathrm{rhs}_M(q, \sigma, \langle p_1, ..., p_k\rangle)$ and $\Theta = [\![\langle q', x_i\rangle \leftarrow y_1 \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle,\ q' \in S_i]\!]$. Then $h'_\sigma((p_1, S_1), ..., (p_k, S_k)) = (p, S)$, where $p = h_\sigma(p_1, ..., p_k)$ and $S = \{q \in Q \mid \xi_q \Theta = y_1\}$. Let the rule

$$\langle q, \sigma(x_1, ..., x_k)\rangle(y_1, ..., y_m) \to \zeta \qquad \langle (p_1, S_1), ..., (p_k, S_k)\rangle$$

be in R', where $\zeta = \xi_q \Theta$ if $\xi_q \Theta \neq y_1$ and, otherwise, $\zeta = \text{dummy}(y_1)$ with $\text{dummy} \in \Delta^{(1)}$.

Intuitively, the look-ahead automaton of M' determines all erased occurrences and the substitution Θ removes them from the right-hand sides. By construction, M' is nonerasing and nondeleting. Let $s \in T_\Sigma$. It should be clear that $h'(s) = (p, S)$ with $p = h(s)$ and $S = \{q \in Q \mid M_q(s) = y_1\}$ and that for every $q \in Q$, if $M_q(s) \notin Y$ then $M'_q(s) = M_q(s)$. Hence $\tau_{M'} = \tau_M$ and the construction of M' preserves fcp. Now let M be wfci. To prove that M' is fci, it can be shown that for every $s \in T_\Sigma$ and every $u \in \text{Occ}(s)$ with $u \neq \varepsilon$,

$$\begin{aligned}&\hat{M}'_q(s[u \leftarrow (p, S)])\\ &\qquad = \hat{M}_q(s[u \leftarrow p])[\langle\!\langle r, p \rangle\!\rangle \leftarrow \langle\!\langle r, (p, S) \rangle\!\rangle \mid r \notin S][\![\langle\!\langle r, p \rangle\!\rangle \leftarrow y_1 \mid r \in S]\!],\end{aligned}$$

where $(p, S) = h'(s/u)$. This implies that $\text{sts}_{M'}(s, u)$ is equal to $\text{sts}_M(s, u)$ restricted to states that occur at nonerased nodes, which is equal to $\text{psts}_M(s, u)$; thus, $\text{sts}_{M'}(s, u) = \text{psts}_M(s, u)$. ∎

From Lemma 7.11 we obtain the theorem.

THEOREM 7.12. $MTT^R_{\text{wfc}} = MTT^R_{\text{fc}}$.

We will now show that MTT_{wfc} is closed under regular look-ahead.

LEMMA 7.13. $MTT^R_{\text{wfc}} = MTT_{\text{wfc}}$.

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\text{MTT}^{\text{R}}_{\text{wfc}}$ with $P = \{p_1, ..., p_n\}$. For $k \geqslant 0$ let $\text{bal}(k)$ be the balanced n-ary tree of height k which contains, for every $i < k$, at the ith level only the symbol $\langle \tau, x_i \rangle$ (of rank n) and at the leaf $i_1 \cdots i_k$ with $i_1, ..., i_k \in [n]$ the symbol $\langle p_{i_1}, ..., p_{i_k} \rangle$. Thus, $\text{bal}(0) = \langle\ \rangle$, $\text{bal}(1) = \langle \tau, x_1 \rangle(\langle p_1 \rangle, ..., \langle p_n \rangle)$, $\text{bal}(2) = \langle \tau, x_1 \rangle(\langle \tau, x_2 \rangle(\langle p_1, p_1 \rangle, ..., \langle p_1, p_n \rangle), \langle \tau, x_2 \rangle(\langle p_2, p_1 \rangle, ..., \langle p_2, p_n \rangle), ..., \langle \tau, x_2 \rangle(\langle p_n, p_1 \rangle, ..., \langle p_n, p_n \rangle))$, etc.

The construction of M' is similar to the one in Theorem 4.21 of [EV85], where it is proved that $MTT^R = MTT$. It has a special "test state" τ which simulates the look-ahead automaton by deleting all its parameters, except the one corresponding to the correct look-ahead state. Define $M' = (Q', \Sigma, \Delta, q_0, R')$ with $Q' = \{\tau^{(n)}\} \cup Q$. For every $q \in Q^{(m)}$ and $\sigma \in \Sigma^{(k)}$ let

$$\langle q, \sigma(x_1, ..., x_k) \rangle(y_1, ..., y_m) \to \zeta$$

be in R', where $\zeta = \text{bal}(k)[\langle r_1, ..., r_k \rangle \leftarrow \text{rhs}_M(q, \sigma, \langle r_1, ..., r_k \rangle) \mid r_1, ..., r_k \in P]$. For every $\sigma \in \Sigma^{(k)}$ let the rule

$$\langle \tau, \sigma(x_1, ..., x_k) \rangle(y_1, ..., y_n) \to \xi$$

be in R', where $\xi = \text{bal}(k)[\langle r_1, ..., r_k \rangle \leftarrow y_j \mid r_1, ..., r_k \in P, h_\sigma(r_1, ..., r_k) = p_j]$.

Let $s \in T_\Sigma$. Since $M'_\tau(s) = y_i$ if $h(s) = p_i$, it is straightforward to see that for every $q \in Q$, $M'_q(s) = M_q(s)$. Viewing M' as an MTT^{R} with a single (dummy) look-ahead state p, it can be shown that for every $u \in \mathrm{Occ}(s)$, if $h(s/u) = p_i$, then $\hat{M}'_\tau(s[u \leftarrow p]) = \langle\!\langle \tau, p \rangle\!\rangle(y_{j_1}, ..., y_{j_n})$, where $\hat{h}(s[u \leftarrow p_i]) = p_{j_v}$ for $v \in [n]$, and $\hat{M}'_q(s[u \leftarrow p]) [\![\langle\!\langle \tau, p \rangle\!\rangle \leftarrow y_i]\!] [\langle\!\langle q, p \rangle\!\rangle \leftarrow \langle\!\langle q, p_i \rangle\!\rangle \mid q \in Q] = \hat{M}_q(s[u \leftarrow p_i])$. Hence, $\mathrm{psts}_{M'}(s, u) = \mathrm{psts}_M(s, u)$ and hence, if M is wfc then so is M'. ▮

Altogether we obtain the theorem.

THEOREM 7.14. $MSOTT = MTT_{\mathrm{wfc}}$.

Future work. Let us discuss some questions which could be subject to future research. Let us consider Theorem 6.12 again. What is the correspondence between the input copying bound N of an $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{fc}}$ and the sur partition of the corresponding $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{sur}}$? For surp $\mathrm{MTT}^{\mathrm{R}}$s we know the answer. For $N \geqslant 1$, let $MTT^{\mathrm{R}}_{\mathrm{fci}(N),\,\mathrm{surp}}$ denote the class of translations realized by $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{surp}}$s with input copying bound less than or equal to N, and let $MTT^{\mathrm{R}}_{\mathrm{suri}(N),\,\mathrm{surp}}$ denote the class of translations realized by $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{suri},\,\mathrm{surp}}$s for which the size of each $\bar{Q}$ in the sur partition Π is less than or equal to N. Then it can be shown that $MTT^{\mathrm{R}}_{\mathrm{fci}(N),\,\mathrm{surp}} = MTT^{\mathrm{R}}_{\mathrm{suri}(N),\,\mathrm{surp}}$, as follows. Let M be an $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{fci},\,\mathrm{surp}}$ with input copying bound N. Then, by the proof of Lemma 6.10 (and Lemma 6.6), there is an $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{suri},\,\mathrm{surp}}$ M' such that for each $\bar{Q}$ in the sur partition of M', $|\bar{Q}| = |\mathrm{Marked}(w)|$ and $|w| \leqslant N$. Hence, $MTT^{\mathrm{R}}_{\mathrm{fci}(N),\,\mathrm{surp}} \subseteq MTT^{\mathrm{R}}_{\mathrm{suri}(N),\,\mathrm{surp}}$. In the proof of Theorem 6.12 it is shown that $MTT^{\mathrm{R}}_{\mathrm{sur}} \subseteq MTT^{\mathrm{R}}_{\mathrm{fci},\,\mathrm{surp}}$. Let M be an $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{suri}(N),\,\mathrm{surp}}$ and let Π be a sur partition for M. It is shown in the proof of Theorem 6.12 that $\max(|\bar{Q}| \mid \bar{Q} \in \Pi)$ is an input copying bound for M. However, there it is assumed that M is nondeleting and the construction of a nondeleting MTT in the proof of Lemma 6.6 does not preserve the sizes of the sets in the sur partition. But clearly, if $\mathrm{sts}_M(s, u) = q_1 \cdots q_n$ then $\mathrm{sts}_M(s, ui)$ contains the same or less states as $\mathrm{sts}_M(\zeta_1) \cdots \mathrm{sts}_M(\zeta_n)$, where $\zeta_v = \mathrm{rhs}_M(q_v, \sigma, \langle p_1, ..., p_k \rangle)$; cf. Lemma 6.9. Hence, the claim in the proof of Theorem 6.2 can also be proved for deleting $\mathrm{MTT}_{\mathrm{sur}}$s, and hence, $\max(|\bar{Q}| \mid \bar{Q} \in \Pi)$ is an input copying bound for M.

There seems to be a relation between the number of synthesized attributes of an $\mathrm{ATT}^{\mathrm{R}}_{\mathrm{sur}}$ and the maximal size of the elements of the sur partition of an $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{sur}}$. However, from our constructions this relationship is not immediate.

From Corollary 7.2 we know that every MSO definable tree translation can be realized by an MTT. Is it decidable for an MTT M, whether τ_M is MSO definable? What classes of macro tree transducers that are larger than those of fc $\mathrm{MTT}^{\mathrm{R}}$s and wfc MTTs can be defined such that the translations realized by them are MSO definable?

It is shown in Theorem 4.2.1 of [EV85] that $MTT^R = MTT$, but the construction does not preserve the fc and sur properties. For weak fc we have shown in Theorem 7.14 that $MTT^R_{\mathrm{wfc}} = MTT_{\mathrm{wfc}}$. Is MTT_{fc} properly included in MTT^R_{fc}, and the analogous question for sur.

ACKNOWLEDGMENTS

We thank the referees for their critical reading and many constructive remarks. We thank, in particular, one of the referees for pointing out a misleading statement in Section 7.

Received September 10, 1998; final manuscript received February 20, 1999

REFERENCES

[AU71] Aho, A. V., and Ullman, J. D. (1971), Translations on a context-free grammar, *Inform. and Control* **19**, 439–475.

[BE98] Bloem, R., and Engelfriet, J. (1998), "A Comparison of Tree Transductions Defined by Monadic Second Order Logic and by Attribute Grammars," Technical Report 98-02, Leiden University. [To appear in *J. Comp. Syst. Sci*].

[Boc76] Bochmann, G. V. (1976), Semantic evaluation from left to right, *Comm. ACM* **19**, 55–62.

[CE95] Courcelle, B., and Engelfriet, J. (1995), A logical characterization of the sets of hypergraphs defined by hyperedge replacement grammars, *Math. Systems Theory* **28**, 515–552.

[CF82] Courcelle, B., and Franchi-Zannettacci, P. (1982), Attribute grammars and recursive program schemes, *Theoret. Comput. Sci.* **17**, 163–191, 235–257.

[Cou83] Courcelle, B. (1983), Fundamental properties of infinite trees, *Theoret. Comput. Sci.* **25**, 95–169.

[Cou94] Courcelle, B. (1994), Monadic second-order definable graph transductions: A survey, *Theoret. Comput. Sci.* **126**, 53–75.

[DKH97] Drewes, F., Kreowski, H.-J., and Habel, A. (1997), Hyperedge replacement graph grammars, *in* "Handbook of Graph Grammars and Computing by Graph Transformation," (G. Rozenberg, Ed.), Vol. 1, Chap. 2, p. 95–162, World Scientific, Singapore.

[Dre97] Drewes, F. (1997), "A Characterization of the Sets of Hypertrees Generated by Hyperedge-Replacement Graph Grammars," Technical Report 3/97, Universität Bremen, [To appear in *Theory Comput. Systems*]

[EF81] Engelfriet, J., and Filé, G. (1981), The formal power of one-visit attribute grammars, *Acta Inform.* **16**, 275–302.

[EH91] Engelfriet, J., and Heyker, L. (1991), The string-generating power of context-free hypergraph grammars, *J. Comp. Syst. Sci.* **43**, 328–360.

[EM] Engelfriet, J., and Maneth, S., Tree languages generated by context-free graph grammars. [To appear *in* "Proc. TAGT '98," Lect. Notes in Comput. Sci., Springer-Verlag, New York/Berlin].

[Eng75] Engelfriet, J. (1975), Bottom-up and top-down tree transformations—A comparison, *Math. Systems Theory* **9**(3), 198–231.

[Eng77] Engelfriet, J. (1977), Top-down tree transducers with regular look-ahead, *Math. Systems Theory* **10**, 289–303.

[Eng80] Engelfriet, J. (1980), Some open questions and recent results on tree transducers and tree languages, *in* "Formal Language Theory; Perspectives and Open Problems" (R. V. Book, Ed.), Academic Press, New York.

[Eng81] Engelfriet, J. (1981), "Tree Transducers and Syntax-Directed Semantics," Technical Report Memorandum 363, Technische Hogeschool Twente. [Also *in* "Proceedings of 7th Colloquium on Trees in Algebra and Programming (CAAP 1982), Lille, France, 1982."]

[Eng82] Engelfriet, J. (1982), Three hierarchies of transducers, *Math. Systems Theory* **15**, 95–125.

[Eng97] Engelfriet, J. (1997), Context-free graph grammars, *in* "Handbook of Formal Languages" (G. Rozenberg and A. Salomaa, Eds.), Vol. 3, Chap. 3, Springer-Verlag, New York/Berlin.

[ER97] Engelfriet, J., and Rozenberg, G. (1997), Node replacement graph grammars, *in* "Handbook of Graph Grammars and Computing by Graph Transformation" (G. Rozenberg, Ed.), Vol. 1, Chap. 1, pp. 1–94, World Scientific, Singapore.

[ERS80] Engelfriet, J., Rozenberg, G., and Slutzki, G. (1980), Tree transducers, L systems, and two-way machines, *J. Comput. System Sci.* **20**, 150–202.

[EV85] Engelfriet, J., and Vogler, H. (1985), Macro tree transducers, *J. Comput. System Sci.* **31**, 71–146.

[EV88] Engelfriet, J., and Vogler, H. (1988), High level tree transducers and iterated pushdown tree transducers, *Acta Inform.* **26**, 131–192.

[EV94] Engelfriet, J., and Vogler, H. (1994), The translation power of top-down tree-to-graph transducers, *J. Comput. System Sci.* **49**, 258–305.

[EvO97] Engelfriet, J., and van Oostrom, V. (1997), Logical description of context-free graph languages, *J. Comput. System Sci.* **55**(3), 489–503.

[FHVV93] Fülöp, Z., Herrmann, F., Vágvölgyi, S., and Vogler, H. (1993), Tree transducers with external functions, *Theoret. Comput. Sci.* **108**, 185–236.

[Fis68] Fischer, M. J. (1968), "Grammars with Macro-like Productions," Ph.D. thesis, Harvard University, Cambridge, MA.

[Fra82] Franchi-Zannettacci, P. (1982), "Attributes Semantiques et Schemas de Programmes," Ph.D. thesis, Université de Bordeaux I. [Thèse d'Etat]

[Fül81] Fülöp, Z. (1981), On attributed tree transducers, *Acta Cybern.* **5**, 261–279.

[FV95] Fülöp, Z., and Vágvölgyi, S. (1995), Attributed tree transducers cannot induce all deterministic bottom-up tree transformations, *Inform. and Comput.* **116**(2), 231–240.

[FV97] Fülöp, Z., and Vogler, H. (1997), "A Characterization of Attributed Tree Transformations," Technical Report TUD/FI97/15, Technical University of Dresden. [To appear in *Theory Comput. Systems*]

[FV98] Fülöp, Z., and Vogler, H. (1998), "Syntax-Directed Semantics—Formal Models Based on Tree Transducers," EATCS Monographs on Theoretical Computer Science, Springer-Verlag, New York/Berlin.

[Gan83] Ganzinger, H. (1983), Increasing modularity and language-independency in automatically generated compilers, *Sci. Comput. Programming* **3**, 223–278.

[Gie88] Giegerich, R. (1988), Composition and evaluation of attribute coupled grammars, *Acta Informatica* **25**, 355–423.

[GS84] Gécseg, F., and Steinby, M. (1984), "Tree Automata," Akadémiai Kiadó, Budapest.

[GS97] Gécseg, F., and Steinby, M. (1997), Tree automata, *in* "Handbook of Formal Languages" (G. Rozenberg and A. Salomaa, Eds.), Vol. 3, Chap. 1, Springer-Verlag, New York/Berlin.

[Hab92] Habel, A. (1992), "Hyperedge Replacement: Grammars and Languages," Lect. Notes in Comput. Sci., Vol. 643, Springer-Verlag, New York/Berlin.

[Knu68] Knuth, D. E. (1968), Semantics of context-free languages, *Math. Systems Theory* **2**, 127–145. [Corrections in *Math. Systems Theory* **5**, 95–96. (1971)]

[Küh97] Kühnemann, A. (1997), "Berechnungsstärken von Teilklassen primitiv-rekursiver Programmschemata," Ph.D. thesis, Technical University Dresden.

[Küh98] Kühnemann, A. (1998), Benefits of tree transducers for optimizing functional programs, *in* "Foundations of Software Technology & Theoretical Computer Science—FST &TCS'98, 18th International Conference, Proceedings," Lect. Notes in Comput. Sci., Vol. 1530 (V. Arvind and R. Ramanujam, Eds.), pp. 146–157, Springer-Verlag, New York/Berlin.

[KW76] Kennedy, K., and Warren, S. K. (1976), Automatic generation of efficient evaluators for attribute grammars, *in* "Conf. Rec. of 3rd Symp. on Principles of Programming Languages," pp. 32–49.

[Man98] Maneth, S. (1998), The generating power of total deterministic tree transducers, *Inform. and Comput.* **147**, 111–144.

[Rao97] Raoult, J.-C. (1997), Rational tree relations, *Bull. Belg. Math. Soc.* **4**, 149–176. [See also Recursively defined tree transductions, *in* "Proc. RTA'93," Lect. Notes in Comput. Sci., Vol. 690, pp. 343–357, Springer-Verlag, New York/Berlin]

[SMFK91] Seki, H., Matsumura, T., Fujii, M., and Kasami, T (1991), On multiple context-free grammars, *Theoret. Comput. Sci.* **88**, 191–229.

[vV96] van Vugt, N. (1996), Generalized context-free grammars, Master's thesis, Internal Report 96-12, Department of Computer Science, Leiden University.

[Wei92] Weir, D. J. (1992), Linear context-free rewriting systems and deterministic tree-walking transducers, *in* "Proc. 30th Annual Meeting of the Association for Computational Linguists."

Printed in Belgium