SIAM J. COMPUT.
Vol. 32, No. 4, pp. 950–1006

MACRO TREE TRANSLATIONS OF LINEAR SIZE INCREASE ARE MSO DEFINABLE*

JOOST ENGELFRIET† AND SEBASTIAN MANETH†‡

Abstract. The first main result is that if a macro tree translation is of linear size increase, i.e., if the size of every output tree is linearly bounded by the size of the corresponding input tree, then the translation is MSO definable (i.e., definable in monadic second-order logic). This gives a new characterization of the MSO definable tree translations in terms of macro tree transducers: they are exactly the macro tree translations of linear size increase. The second main result is that given a macro tree transducer, it can be decided whether or not its translation is MSO definable, and if it is, then an equivalent MSO transducer can be constructed. Similar results hold for attribute grammars, which define a subclass of the macro tree translations.

Key words. tree transducers, monadic second-order logic, decidability

AMS subject classifications. 68Q45, 68Q42, 03D05

DOI. 10.1137/S0097539701394511

1. Introduction.

Very often a complex object has a structure that shows how it is composed from smaller objects by the application of certain operations. The smaller objects may themselves be composed of other objects. Such a structure can naturally be described as a tree, and hence the objects are "tree-structured." Examples of tree-structured objects are the words of a context-free language (with derivation trees as structure) or the graphs of bounded tree-width (with tree decompositions as structure). Now consider the transformation of a tree-structured object, based on its structure and independent of the interpretation of the operations, i.e., a tree-to-tree transformation. We are interested in models of such transformations: tree transducers. Well-known examples of tree transducers are top-down tree transducers [46, 48, 1, 32] and attribute grammars [17, 27, 28] (motivated by syntax-directed semantics and compilers; cf. [35, 37, 40, 53]), unranked tree transducers [43, 2] and pebble tree transducers [45] (motivated by the transformation of XML documents; cf. [51]), and macro tree transducers [15, 8, 9, 24, 28] (motivated by syntax-directed and denotational semantics [35, 47], and used as a model in, e.g., functional programming [52, 39, 41], language prototyping [49], and linguistics [38, 44]). Motivated by model theory is the idea of "interpretation," meaning the definition of a (logical) structure in terms of logical formulas over another structure (cf. Chapter 10 of [13]). For monadic second-order (MSO) logic, such MSO interpretations have recently been used to characterize the generation of graphs by context-free graph grammars [5, 7, 23, 16] (see also [36]). Taking trees as a logical structure, another type of tree transducer is obtained: the MSO tree transducer, studied in [3, 19] (for strings, see [18]). An important part of tree transducer theory is comparing the formal power of these different models of transformation of tree-structured objects and providing effective translations between these models. This paper compares the power of macro tree transducers and MSO tree transducers.

*Received by the editors August 30, 2001; accepted for publication (in revised form) March 12, 2003; published electronically June 25, 2003.
http://www.siam.org/journals/sicomp/32-4/39451.html

†Leiden University, LIACS, P.O. Box 9512, 2300 RA Leiden, The Netherlands (engelfri@liacs.nl).
‡Present address: Swiss Institute of Technology Lausanne, Programming Methods Laboratory (LAMP), 1015 Lausanne, Switzerland (sebastian.maneth@epfl.ch).

The macro tree transducer (MTT) is a finite state device that translates, in a recursive top-down fashion, an input tree into an output tree, handling context information by the use of parameters. The states of the MTT can be viewed as functions that call each other recursively; the initial state is the main function. The (tree-to-tree) translations of MTTs form a large class containing the translations of top-down tree transducers and attribute grammars. In order to prove our results, we add the feature of regular look-ahead (see, e.g., section 18 of [32]) to top-down tree transducers, attribute grammars, and MTTs. Note that in the case of MTTs this has no influence on the translations: the classes of translations realized by MTTs with and without regular look-ahead are the same [24].

The MSO tree transducer uses formulas in monadic second-order logic to define tree-to-tree translations. This provides a declarative way of defining a tree translation, as opposed to the operational way of an MTT. The idea is to define the nodes and edges of the output tree in terms of MSO formulas that are interpreted in the input tree, or, more precisely, in a fixed number of disjoint copies of the input tree. Tree translations definable in MSO logic have nice properties, comparable to those of finite state transductions on strings. In particular, they are closed under composition and they can be computed in linear time. Macro tree translations do not possess these properties.

The question arises, What is the precise relationship between these two different models? From [3, 19] it is known that every MSO definable tree translation can be realized by an MTT. However, the converse does not hold for obvious reasons: by definition, MSO definable tree translations are of linear size increase: the size of the output tree is at most k times the size of the input tree, where k is the number of disjoint copies of the input tree, used to define the output tree. On the other hand, the translations realized by MTTs can be of double exponential size increase (cf. Lemma 4.22 of [28]). Our first main result is that if we restrict ourselves to translations of linear size increase, then the two formalisms, MSO tree transducers and MTTs, have exactly the same power, i.e., the respective classes of translations coincide.

Let us briefly discuss the proof of the first main result. As mentioned before, our MTTs are equipped with regular look-ahead. In [19] a characterization of the MSO definable tree translations in terms of MTTs is given: they are the translations realized by "finite copying" MTTs. The notion of finite copying was introduced in [1] for generalized syntax-directed translation schemes, which are closely related to top-down tree transducers. It requires that there be a bound on the number of occurrences of states that translate a given node of the input tree. For MTTs this requirement is called "finite copying in the input," and an MTT is finite copying [19] if it is both finite copying in the input and "finite copying in the parameters"; the latter means that there is a bound on the number of copies made of a parameter. We want to prove that if the translation realized by an MTT is of linear size increase, then it is MSO definable. By the above this is equivalent to showing that for every MTT M that is of linear size increase (i.e., which realizes a translation of linear size increase), there is an equivalent MTT M' that is finite copying. How can we construct M', given M? The idea is that every MTT M can be transformed into a normal form M', called the "proper normal form" of M, such that if M is of linear size increase, then M' is finite copying. Roughly speaking this normal form requires that all states and parameters of M' are really "needed"; more precisely, each state generates infinitely many output trees (considering all possible input trees), and for each parameter y there are infinitely many actual parameter trees being substituted for y (for all possible input trees). Then for a proper MTT M' it can be shown that (i) if M' is of linear size increase, then it is finite copying in the parameters, and (ii) if M' is finite copying in the parameters and of linear size increase, then it is finite copying in the input. Both (i) and (ii) are proved by a pumping argument; i.e., it is shown that if M' is not finite copying in the parameters, then it is not of linear size increase, and similarly for (ii).

Our second main result concerns decidability. Given an MTT it can be decided whether or not its translation is MSO definable, and if so, an equivalent MSO tree transducer can be constructed. The proof is based on the following results: (1) the translation realized by an MTT M is MSO definable—i.e., of linear size increase—if and only if its proper normal form M' is finite copying (by the proof of our first main result, as discussed above); (2) for an MTT it is decidable whether or not it is finite copying (the proof is based on the fact that the finiteness of ranges of MTTs is decidable [12]); and (3) from [19, 3] it follows that given a finite copying MTT, an equivalent MSO tree transducer can be constructed.

Note that very often membership in a subclass is undecidable (such as regularity of a context-free language). In cases of decidability there is often a characterization of the subclass that is independent of the device that defines the whole class, i.e., a "semantic" rather than "syntactic" characterization, such as our linear size increase characterization. As another example, in [6] it is shown that an NR (node replacement) context-free graph language can be generated by an HR (hyperedge replacement) context-free graph grammar if and only if the number of edges of its graphs is linearly bounded by the number of nodes.

The idea for our main results stems from [1]; there it is shown that a generalized syntax-directed translation (gsdt) scheme can be realized by a tree-walking transducer if and only if it is of linear size increase. Since gsdt schemes are a variation of top-down tree transducers, and tree-walking transducers are closely related to finite copying top-down tree transducers [22], our result can be viewed as a generalization of the result of [1], from top-down tree transducers to MTTs. In fact, since the proper normal form of a top-down tree transducer is again a top-down tree transducer, we reobtain their result (in our formalism): the top-down tree translations of linear size increase are exactly the translations realized by finite copying top-down tree transducers. Moreover, they are exactly the MSO definable top-down tree translations.

The main result of [19], on which this paper is based, is in turn based on the main result of [3], which states that the MSO definable tree translations can be characterized by attribute grammars (more precisely, by attributed tree transducers with look-ahead) that are single-use restricted. The single-use restriction [33, 30, 39, 41] is interesting, because attribute grammars are closed under left-composition with single-use restricted attribute grammars. Our results now imply that given an attributed tree transducer (with look-ahead) it can be decided whether or not there exists an equivalent one that is single-use restricted, and furthermore that the linear size increase attributed tree translations are precisely the MSO definable tree translations.

This paper is structured as follows. In section 2 trees and tree substitutions are defined. In particular, the definition of second-order tree substitution is given, which is the type of substitution that MTTs are based on. Various results about these substitutions are proved, for example, how to compute the number of occurrences of a particular symbol in a tree to which a second-order tree substitution is applied. Then tree languages and tree translations are defined, and the notion of MSO definable tree translation is recalled briefly. Section 3 defines MTTs, which are total deterministic and equipped with regular look-ahead. Some basic results needed in the paper are recalled, and two subclasses defined by restrictions on the parameters are considered. Section 4 recalls the notion of finite copying, which consists of two parts: finite copying in the input and finite copying in the parameters. It is proved that it is decidable for an MTT whether or not it is finite copying. Moreover, although this is already known from the result of [19], it is proved for the sake of completeness that if an MTT is finite copying, then it is of linear size increase. The proof is based on an intermediate, very natural notion of bounded copying: "finite contribution." An MTT is of finite contribution if there is a bound on the number of output nodes that are contributed by a given node of the input tree. Also in this section the notion of "finite nested copying in the input" is introduced; it requires a bound on the amount of nesting of the states that translate a given node of the input tree. In section 5 the proper normal form is introduced, and it is shown how to construct, given an MTT, an equivalent one in proper normal form. Section 6 proves our main results: if the translation realized by a proper MTT M is of linear size increase (for short, "M is lsi"), then M is finite copying. The proof goes in three stages: (I) If M is lsi, then it is finite nested copying in the input. (II) If M is lsi and finite nested copying in the input, then it is finite copying in the parameters. And finally, (III) if M is lsi, finite nested copying in the input, and finite copying in the parameters, then it is finite copying in the input. Section 7 presents the main results and their consequences for top-down tree transducers, attribute grammars, and context-free graph grammars. At last, some open problems and further research topics are mentioned.

We note that technically this paper is concerned with MTTs only. The links to MSO tree transducers were established in [3, 19].

2. Preliminaries. The set $\{0, 1, \dots\}$ of natural numbers is denoted by $\mathbb{N}$. The empty set is denoted by $\varnothing$. For $k \in \mathbb{N}$, $[k]$ denotes the set $\{1, \dots, k\}$; thus $[0] = \varnothing$. For a set A, $|A|$ is the cardinality of A, and A^* is the set of all strings over A. The empty string is denoted by ε. The length of a string w is denoted $|w|$, and the number of occurrences of the symbol a in w is denoted by $\#_a(w)$. For a set $B \subseteq A$, $\#_B(w) = \sum\{\#_a(w) \mid a \in B\}$. For strings $v, w_1, \dots, w_n \in A^*$ and distinct $a_1, \dots, a_n \in A$, we denote by $v[a_1 \leftarrow w_1, \dots, a_n \leftarrow w_n]$ the result of (simultaneously) substituting w_i for every occurrence of a_i in v. Note that the substitution $[a_1 \leftarrow w_1, \dots, a_n \leftarrow w_n]$ 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 $a \leftarrow w$ for which condition P holds.

2.1. Trees. A set Σ together with a mapping $\text{rank}_\Sigma \colon \Sigma \to \mathbb{N}$ is called a *ranked set*. For $k \geq 0$, $\Sigma^{(k)}$ is the set $\{\sigma \in \Sigma \mid \text{rank}_\Sigma(\sigma) = k\}$; we also write $\sigma^{(k)}$ to indicate that $\text{rank}_\Sigma(\sigma) = k$. For sets Σ and A, $\langle \Sigma, A\rangle = \Sigma \times A$; if Σ is ranked, then so is $\langle \Sigma, A\rangle$, with $\text{rank}_{\langle \Sigma, A\rangle}(\langle \sigma, a\rangle) = \text{rank}_\Sigma(\sigma)$ 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 $X = \{x_1, x_2, \dots\}$ and the *set of parameters* to be $Y = \{y_1, y_2, \dots\}$. For $k \geq 0$, $X_k = \{x_1, \dots, x_k\}$ and $Y_k = \{y_1, \dots, y_k\}$. In this paper (as opposed to other papers) we allow a ranked set to contain parameters. However, by convention, if a parameter is an element of a ranked set Σ, then we require that it have rank zero; i.e., we require that $\Sigma \cap Y \subseteq \Sigma^{(0)}$. Thus, *parameters have rank zero*.

Let Σ be a ranked set. The *set of trees over* Σ, denoted by T_Σ, is the smallest set of strings $T \subseteq \Sigma^*$ such that if $\sigma \in \Sigma^{(k)}$, $k \geq 0$, and $t_1, \dots, t_k \in T$, then $\sigma\, t_1 \cdots t_k \in T$.

For better readability we will usually write $\sigma(t_1,\dots,t_k)$ for $\sigma\, t_1\cdots t_k$. For a set of parameters $Y' \subseteq Y$ we will also use $T_\Sigma(Y')$ to denote the set of trees $T_{\Sigma\cup Y'}$, where $\Sigma\cup Y'$ is the ranked set with $\mathrm{rank}_{\Sigma\cup Y'}(\sigma) = \mathrm{rank}_\Sigma(\sigma)$ for $\sigma\in\Sigma$, and $\mathrm{rank}_{\Sigma\cup Y'}(y) = 0$ for $y \in Y'$.

For every tree $t \in T_\Sigma$, the *set of nodes of* t, denoted by $V(t)$, is a subset of $\mathbb{N}^*$ which is inductively defined as follows: if $t = \sigma(t_1,\dots,t_k)$ with $\sigma \in \Sigma^{(k)}$, $k \geq 0$, and for all $i \in [k], t_i \in T_\Sigma$, then $V(t) = \{\varepsilon\} \cup \{iu \mid u \in V(t_i), i \in [k]\}$. Thus, ε represents the root of a tree, and for a node u the ith child of u is represented by ui. A *leaf* is a node without children. If $u = vw$ with $w \in \mathbb{N}^*$, then v is an *ancestor of* u and u is a *descendant of* v; if $w \neq \varepsilon$, then v is a *proper ancestor of* u, and u is a *proper descendant of* v. The *label of* t *at node* u is denoted by $t[u]$; we also say that $t[u]$ occurs in t (at u). The *subtree of* t *at node* u is denoted by t/u. The *substitution of the tree* $s \in T_\Sigma$ *at node* 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 $t = \sigma(t_1,\dots,t_k)$, $i \in [k]$, and $u \in V(t_i)$, then $t[iu] = t_i[u]$, $t/iu = t_i/u$, and $t[iu \leftarrow s] = \sigma(t_1,\dots,t_i[u\leftarrow s],\dots,t_k)$.

The usual preorder of the nodes of t (which, in fact, is the lexicographical order on $\mathbb{N}^*$) is denoted $<$; thus, $\varepsilon < iu$ (for $i \geq 1$), if $u < v$, then $iu < iv$, and if $i < j$, then $iu < jv$.

The *size* of a tree t, denoted by size(t), is the number $|V(t)|$ of nodes of t. For $t = \sigma(t_1,\dots,t_k)$, size$(t)$ equals $1 + \mathrm{size}(t_1) + \cdots + \mathrm{size}(t_k)$; note that $\mathrm{size}(t) = \sum_{\sigma\in\Sigma} \#_\sigma(t) = |t|$. For $\sigma \in \Sigma$, $V_\sigma(t)$ denotes the set of nodes of t which are labeled by σ, i.e., $\{u \in V(t) \mid t[u] = \sigma\}$; note that $|V_\sigma(t)| = \#_\sigma(t)$: the number of occurrences of σ in t. For a set $S \subseteq \Sigma$, $V_S(t) = \bigcup_{\sigma\in S} V_\sigma(t)$. The *height* of t is denoted by height(t); for $t = \sigma(t_1,\dots,t_k)$ it equals $1 + \max\{\mathrm{height}(t_i) \mid i \in [k]\}$.

2.2. Tree substitution. In the previous subsection on trees we already defined a particular tree substitution: for trees t, s and a node u of t, $t[u \leftarrow s]$ is the result of replacing in t the subtree t/u by s. Now we want to consider replacing in t all occurrences of a symbol σ.

Trees are particular strings and therefore string substitution as defined in the beginning of these preliminaries is applicable to a tree. In order to guarantee that the resulting string is again a tree, we require that only symbols of rank zero, i.e., leaves, may be replaced by trees; we refer to this type of substitution as "first-order tree substitution." Note that top-down tree transducers are based on first-order tree substitution. In contrast to this, "second-order tree substitution" means that symbols of arbitrary rank can be replaced by a tree. This is the type of substitution MTTs are based on. Consider the replacement of a symbol σ of rank k by a tree s in which the parameter symbols $y_1,\dots,y_k$ occur to indicate where the subtrees of σ have to be inserted. Now, if σ occurs at a node u of the tree t, then replacing it by s means to replace the subtree t/u of t by the tree $s[y_1 \leftarrow t/u1,\dots,y_k \leftarrow t/uk]$. (Hence the first-order tree substitution is used to define the second-order one.) Now we define second-order tree substitution formally. Since all occurrences of σ have to be replaced simultaneously, an inductive definition is appropriate.

Let Σ be a ranked set and let $\sigma_1,\dots,\sigma_n$ be distinct elements of $\Sigma - Y$, $n \geq 1$, and for each $i \in [n]$ let s_i be a tree in $T_{\Sigma - Y}(Y_{\mathrm{rank}_\Sigma(\sigma_i)})$. For $t \in T_\Sigma$, the *second-order tree substitution of* σ_i *by* s_i *in* t, denoted by $t[\![\sigma_1 \leftarrow s_1,\dots,\sigma_n \leftarrow s_n]\!]$, is inductively defined as follows (abbreviating $[\![\sigma_1 \leftarrow s_1,\dots,\sigma_n \leftarrow s_n]\!]$ by $[\![\dots]\!]$): For $t = \sigma(t_1,\dots,t_k)$ with $\sigma \in \Sigma^{(k)}$, $k \geq 0$, and $t_1,\dots,t_k \in T_\Sigma$, (i) if $\sigma = \sigma_i$ for an $i \in [n]$, then $t[\![\dots]\!] = s_i[y_j \leftarrow t_j[\![\dots]\!] \mid j \in [k]]$ and (ii) otherwise $t[\![\dots]\!] = \sigma(t_1[\![\dots]\!],\dots,t_k[\![\dots]\!])$.

We will say that $[\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$ is a second-order tree substitution over Σ. Note that it is a mapping from T_Σ to T_Σ. In fact, it is a tree homomorphism [31]. Note also that (just as first-order tree substitution) second-order tree substitution is associative (by the closure of tree homomorphisms under composition; cf. Theorem IV.3.7 of [31]), i.e., $t[\![\sigma \leftarrow s]\!][\![\sigma \leftarrow s']\!] = t[\![\sigma \leftarrow s[\![\sigma \leftarrow s']\!]]\!]$, 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 [4]). 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 $\sigma \leftarrow s$ for which condition P holds. In second-order tree substitutions we use for the relabeling $\sigma \leftarrow \delta(y_1, \dots, y_k)$ of $\sigma^{(k)}$ by $\delta^{(k)}$ the abbreviation $\sigma \leftarrow \delta$; note that this is, in fact, a string substitution.

Note that the restrictions on σ_i and s_i in the first sentence of the previous paragraph are rather subtle. Recall from section 2.1 that Σ can contain parameters. However, for technical reasons, we do not want a second-order tree substitution to replace parameters (hence $\sigma_i \in \Sigma - Y$), and we do not want it to introduce parameters (hence $s_i \in T_{\Sigma-Y}(Y_{\mathrm{rank}_\Sigma(\sigma_i)})$, which means that $s_i \in T_{\Sigma\cup Y}$, and if y_j occurs in s_i, then $j \leq \mathrm{rank}_\Sigma(\sigma_i)$).

The second-order tree substitution $[\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$ is *nondeleting* if for every $i \in [n]$, $\#_{y_j}(s_i) \geq 1$ for all $j \in [\mathrm{rank}_\Sigma(\sigma_i)]$, and it is *nonerasing* if for every $i \in [n]$, $s_i \notin Y$. It is *productive* if it is both nondeleting and nonerasing.

LEMMA 2.1. *Let Σ be a ranked alphabet and $\Phi = [\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$ a nondeleting second-order tree substitution over Σ. For all $t, t' \in T_\Sigma$, if t' is a subtree of t, then $t'\Phi$ is a subtree of $t\Phi$. In particular, for $y \in Y$, if $\#_y(t) \geq 1$, then $\#_y(t\Phi) \geq 1$.*

Proof. For $t = \sigma(t_1, \dots, t_k)$, $t_j\Phi$ is a subtree of $t\Phi$. Hence the result follows immediately by induction on the structure of t.

If $\#_y(t) \geq 1$, then y is a subtree of t (because, by convention, the parameter y has rank 0). This means, by the first part of this lemma, that y is also a subtree of $t\Phi$, i.e., $\#_y(t\Phi) \geq 1$. Note that $y\Phi = y$ because, by the definition of second-order tree substitution, $\sigma_i \notin Y$ for all $i \in [n]$. ∎

LEMMA 2.2. *Let Σ be a ranked alphabet and $\Phi = [\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$ a nonerasing second-order tree substitution over Σ. For every $t \in T_\Sigma$, if $t \notin Y$, then $t\Phi \notin Y$.*

Proof. Let $t = \sigma(t_1, \dots, t_k)$ with $\sigma \in \Sigma^{(k)} - Y$. If $\sigma \notin \{\sigma_1, \dots, \sigma_n\}$, then $t\Phi = \sigma(t_1\Phi, \dots, t_k\Phi) \notin Y$. If $\sigma = \sigma_i$ for some $i \in [n]$, then $t\Phi = s_i[y_j \leftarrow t_j\Phi \mid j \in [k]] \notin Y$ (because $s_i \notin Y$). ∎

In order to calculate the number of times that a particular node u of a tree is copied by the application of a second-order tree substitution, we need to know which symbols occur at the ancestors of u. For this we define the string obtained by reading the labels of the ancestors of u in descending order, starting at the root; if u is labeled by a parameter, then we do not include its label in this string, because in trees of the form $t[\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$ the parameters present in the trees s_i do not appear (by the requirement that $s_i \in T_{\Sigma-Y}(Y_{\mathrm{rank}_\Sigma(\sigma_i)})$).

For a tree $t \in T_\Sigma$ and a node $u \in V(t)$, the *label path to u (in t)*, denoted by $\mathrm{lpath}(t, u)$, is the string in $(\Sigma - Y)^*$ defined recursively as follows: $\mathrm{lpath}(t, \varepsilon) = \varepsilon$ if $t \in Y$, and otherwise $\mathrm{lpath}(t, \varepsilon) = t[\varepsilon]$; for $i \geq 1$ and $u \in \mathbb{N}^*$, $\mathrm{lpath}(t, iu) = t[\varepsilon]\,\mathrm{lpath}(t/i, u)$. For example, let t be the tree $\gamma(\sigma(a, y_1))$; then $\mathrm{lpath}(t, 12) = \gamma\,\mathrm{lpath}(\sigma(a, y_1), 2) = \gamma\sigma\,\mathrm{lpath}(y_1, \varepsilon) = \gamma\sigma$, $\mathrm{lpath}(t, 1) = \gamma\sigma$, and $\mathrm{lpath}(t, 11) = \gamma\sigma a$.

The following lemma shows how a label path in t changes if a second-order tree substitution is applied to t.

LEMMA 2.3. *Let Σ be a ranked alphabet, Φ the second-order tree substitution $[\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$ over Σ, and $t \in T_\Sigma$.*

(i) *Every label path in $t\Phi$ is of the form $w_0 v_1 w_1 \cdots v_m w_m$, where $m \geq 0$, $w_0 \sigma_{i_1} w_1 \cdots \sigma_{i_m} w_m$ is a label path in t, $i_1, \dots, i_m \in [n]$, v_j is a label path in s_{i_j} for $j \in [m]$, and $w_0, \dots, w_m \in (\Sigma - \{\sigma_1, \dots, \sigma_n\})^*$.*

(ii) *If Φ is nondeleting, then for every $w, v \in \Sigma^*$ such that $w\sigma_i$ is a label path in t and v is a label path in s_i, there is a $w' \in \Sigma^*$ such that $w'v$ is a label path in $t\Phi$.*

2.3. Number of occurrences. Since this paper is about the size increase of MTTs, and they are based on second-order tree substitution, we need to know how the size of a tree t changes when a second-order tree substitution Φ is applied to t. Recall that size$(t\Phi)$ is the sum of the numbers $\#_\sigma(t\Phi)$ of occurrences of σ in $t\Phi$ for all symbols σ. Thus, we need to determine the number $\#_\sigma(t\Phi)$. Since second-order tree substitution is based on first-order tree substitution which is a particular string substitution, we first determine the number $\#_a(w[\dots])$, where w is a string and $[\dots]$ is a string substitution.

The following lemma can be proved by straightforward induction on the length of w.

LEMMA 2.4. *Let A be an alphabet. Let $w, v_1, \dots, v_n \in A^*$ and let $a_1, \dots, a_n$ be distinct elements of A. For every $a \in A$,*

$$\#_a(w[a_1 \leftarrow v_1, \dots, a_n \leftarrow v_n]) = S^a + \sum_{i \in [n]} \#_{a_i}(w) \cdot \#_a(v_i),$$

where $S^a = \#_a(w)$ if $a \notin \{a_1, \dots, a_n\}$, and otherwise $S^a = 0$.

In the next lemma we prove the generalization of Lemma 2.4 to second-order tree substitution. Intuitively we now have to take into account, for a node u of the tree t, how many times it is copied by the application of the second-order tree substitution $\Phi = [\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$. For each σ_i that occurs at a proper ancestor u' of u, u is in some subtree $t/u'j$ of u'; thus, replacing σ_i by s_i generates $\#_{y_j}(s_i)$ copies of $t/u'j$. Hence, the product of these numbers $\#_{y_j}(s_i)$, for all proper ancestors u', determines the number of copies of u in $t\Phi$. In the lemma this product is denoted $\prod F^\Phi_{t,u}$, where the family $F^\Phi_{t,u}$ of numbers is defined as follows.

DEFINITION 2.5 (the family $F^\Phi_{t,u}$). *Let Σ be a ranked alphabet and $\Phi = [\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$ a second-order tree substitution over Σ. For every $t \in T_\Sigma$ and $u \in V(t)$, define the family $F^\Phi_{t,u}$ as*

$$F^\Phi_{t,u} = \{f_{u'}\}_{u' \text{ proper ancestor of } u},$$

where

$$f_{u'} = \begin{cases} 1 & \text{if } t[u'] \notin \{\sigma_1, \dots, \sigma_n\}, \\ \#_{y_j}(s_i) & \text{if } t[u'] = \sigma_i, i \in [n], \text{ and } u = u'ju'' \text{ with } j \geq 1, u'' \in \mathbb{N}^*. \end{cases}$$

Note that if $u = \varepsilon$, i.e., $F^\Phi_{t,u}$ is empty, then $\prod F^\Phi_{t,u} = 1$.

LEMMA 2.6. *Let Σ be a ranked alphabet, $\Phi = [\![\sigma_1 \leftarrow s_1, \dots, \sigma_n \leftarrow s_n]\!]$ a second-order tree substitution over Σ, and $t \in T_\Sigma$. For every $\sigma \in \Sigma$,*

$$\#_\sigma(t\Phi) = S^\sigma_1 + S^\sigma_2,$$

where

$$S_1^\sigma = \sum_{u\in V_\sigma(t)} \prod F_{t,u}^\Phi \quad \textit{if } \sigma \notin \{\sigma_1,\dots,\sigma_n\}, \textit{ and otherwise } S_1^\sigma = 0,$$

$$S_2^\sigma = \sum_{u\in V_{\sigma_i}(t), i\in[n]} \#_\sigma(s_i)\cdot \prod F_{t,u}^\Phi \quad \textit{if } \sigma \notin Y, \textit{ and otherwise } S_2^\sigma = 0.$$

Proof. Denote $\{\sigma_1,\dots,\sigma_n\}$ by Σ_n. Let $O_\varepsilon = V_\sigma(t)\cap\{\varepsilon\}$, $O = V_\sigma(t) - \{\varepsilon\}$, and for $i\in[n]$, $O_{\varepsilon,i} = V_{\sigma_i}(t)\cap\{\varepsilon\}$ and $O_i = V_{\sigma_i}(t) - \{\varepsilon\}$. Clearly, $S_1^\sigma = T_1 + S_1$, where for $\sigma\notin\Sigma_n$, $T_1 = \sum_{u\in O_\varepsilon}\prod F_{t,u}^\Phi$ and $S_1 = \sum_{u\in O}\prod F_{t,u}^\Phi$, and otherwise $T_1 = 0$ and $S_1 = 0$. Similarly, $S_2^\sigma = T_2 + S_2$, where for $\sigma\notin Y$, $T_2 = \sum_{u\in O_{\varepsilon,i}, i\in[n]} \#_\sigma(s_i)\cdot\prod F_{t,u}^\Phi$ and $S_2 = \sum_{u\in O_i, i\in[n]}\#_\sigma(s_i)\cdot\prod F_{t,u}^\Phi$, and otherwise $T_2 = 0$ and $S_2 = 0$.

The proof that $S_1^\sigma + S_2^\sigma$ equals $\#_\sigma(t\Phi)$ is by induction on the structure of t. Let $t = \sigma'(t_1,\dots,t_k)$ with $\sigma'\in\Sigma^{(k)}$, $k\geq 0$, and $t_1,\dots,t_k\in T_\Sigma$.

Case 1. $\sigma'\in\Sigma-\Sigma_n$.

Then $t[\varepsilon]\notin\Sigma_n$ and hence, for every $j\in[k]$ and $v\in V(t_j)$, $\prod F_{t,jv}^\Phi = \prod F_{t_j,v}^\Phi$. Since O equals $\bigcup_{j\in[k]}\{jv \mid v\in V_\sigma(t_j)\}$, it follows that $\sum_{u\in O}\prod F_{t,u}^\Phi$ is equal to $\sum_{v\in V_\sigma(t_j), j\in[k]}\prod F_{t_j,v}^\Phi$, and similarly for O_i. We can apply the induction hypothesis for t_j to $S_{1,j}^\sigma + S_{2,j}^\sigma$, where $S_{1,j}^\sigma = \sum_{v\in V_\sigma(t_j)}\prod F_{t_j,v}^\Phi$ if $\sigma\notin\Sigma_n$, and otherwise $S_{1,j}^\sigma = 0$, and $S_{2,j}^\sigma = \sum_{v\in V_{\sigma_i}(t_j), i\in[n]}\#_\sigma(s_i)\cdot\prod F_{t_j,v}^\Phi$ if $\sigma\notin Y$, and otherwise $S_{2,j}^\sigma = 0$. Since $O_{\varepsilon,i} = \varnothing$ we get that $T_2 = 0$ and hence

$$S_1^\sigma + S_2^\sigma = T_1 + \sum_{j\in[k]}\#_\sigma(t_j\Phi).$$

Now T_1 equals 1 if $\sigma' = \sigma$, and 0 otherwise. By the definition of $\#_\sigma$ this means that the above is equal to $\#_\sigma(\sigma'(t_1\Phi,\dots,t_k\Phi))$. This equals $\#_\sigma(t\Phi)$ by the definition of second-order tree substitution.

Case 2. $\sigma' = \sigma_i$ for some $i\in[n]$.

For every $j\in[k]$ and $v\in V(t_j)$, $\prod F_{t,jv}^\Phi = \#_{y_j}(s_i)\cdot\prod F_{t_j,v}^\Phi$. Thus, $S_1 = \sum_{j\in[k]}\#_{y_j}(s_i)\cdot S_{1,j}^\sigma$ and $S_2 = \sum_{j\in[k]}\#_{y_j}(s_i)\cdot S_{2,j}^\sigma$. By induction, $S_{1,j}^\sigma + S_{2,j}^\sigma = \#_\sigma(t_j\Phi)$. Hence $S_1 + S_2 = \sum_{j\in[k]}\#_{y_j}(s_i)\cdot\#_\sigma(t_j\Phi)$. Now $T_1 = 0$, and if $\sigma\notin Y$, then $T_2 = \#_\sigma(s_i)$, and otherwise $T_2 = 0$. We can apply Lemma 2.4 to $T_1 + T_2 + S_1 + S_2$ (with $a = \sigma$ and $S^a = T_2$) to obtain $\#_\sigma(s_i[y_j\leftarrow t_j\Phi \mid j\in[k]])$, which equals $\#_\sigma(t\Phi)$ by the definition of second-order tree substitution. ∎

Recall from section 2.2 that the second-order tree substitution $\Phi = [\![\sigma_1 \leftarrow s_1,\dots,\sigma_n\leftarrow s_n]\!]$ is nondeleting if each s_i contains at least one occurrence of y_j for every $j\in[\mathrm{rank}_\Sigma(\sigma_i)]$, and nonerasing if no s_i is in Y. We can now use Lemma 2.6 to prove that if Φ is productive, i.e., both nondeleting and nonerasing, then its application does not decrease the size of a tree.

LEMMA 2.7. *Let Σ be a ranked alphabet and $\Phi = [\![\sigma_1\leftarrow s_1,\dots,\sigma_n\leftarrow s_n]\!]$ a second-order tree substitution over Σ. If Φ is productive, then size$(t\Phi)\geq$ size(t) for every $t\in T_\Sigma$.*

Proof. Let $\Sigma_n = \{\sigma_1,\dots,\sigma_n\}$. Since $\mathrm{size}(t\Phi) = \sum_{\sigma\in\Sigma}\#_\sigma(t\Phi)$, we can apply Lemma 2.6 to obtain $\sum_{\sigma\in\Sigma}S_1^\sigma + \sum_{\sigma\in\Sigma}S_2^\sigma$, where S_1^σ and S_2^σ are as in Lemma 2.6.

Since Φ is nondeleting, for every $u\in V_\sigma(t)$, $\prod F_{t,u}^\Phi\geq 1$. Thus

$$\mathrm{size}(t\Phi)\geq \sum_{\sigma\in\Sigma-\Sigma_n, u\in V_\sigma(t)} 1 + \sum_{u\in V_{\sigma_i}(t), i\in[n]}\ \sum_{\sigma\in\Sigma-Y}\#_\sigma(s_i).$$

Since Φ is nonerasing, each s_i contains at least one symbol in $\Sigma - Y$. Hence

$$\mathrm{size}(t\Phi) \geq \sum_{\sigma\in\Sigma-\Sigma_n, u\in V_\sigma(t)} 1 + \sum_{\sigma\in\Sigma_n, u\in V_\sigma(t)} 1 = \sum_{\sigma\in\Sigma, u\in V_\sigma(t)} 1 = \mathrm{size}(t). \qquad \square$$

2.4. Tree languages. A *tree language* is a subset of T_Σ for some ranked alphabet Σ.

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 $\tilde{h}(\sigma(s_1, \dots, s_k)) = h_\sigma(\tilde{h}(s_1), \dots, \tilde{h}(s_k))$ for every $\sigma \in \Sigma^{(k)}$, $k \geq 0$, and $s_1, \dots, s_k \in T_\Sigma$. 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 $L = \{s \in T_\Sigma \mid h(s) \in F\}$. For $p \in P$ the tree language $\{s \in T_\Sigma \mid h(s) = p\}$ is denoted by L_p. Note that, in particular, L_p is regular for every $p \in P$.

2.5. Tree translations. Let Σ and Δ be ranked alphabets. A (total) function $\tau : T_\Sigma \to T_\Delta$ is called a *tree translation* or simply a translation. For a tree language $L \subseteq T_\Sigma$, $\tau(L)$ denotes the set $\{t \in T_\Delta \mid t = \tau(s) \text{ for some } s \in L\}$, and for $L \subseteq T_\Delta$, $\tau^{-1}(L)$ denotes $\{s \in T_\Sigma \mid \tau(s) \in L\}$. For a class $\mathcal{T}$ of tree translations and a class $\mathcal{L}$ of tree languages, $\mathcal{T}(\mathcal{L})$ denotes the class of tree languages $\{\tau(L) \mid \tau \in \mathcal{T}, L \in \mathcal{L}\}$.

A tree translation $\tau : T_\Sigma \to T_\Delta$ is *of linear size increase* (for short, *lsi*) if there is a $c \in \mathbb{N}$ such that $\mathrm{size}(\tau(s)) \leq c \cdot \mathrm{size}(s)$ for all $s \in T_\Sigma$. The class of all tree translations of linear size increase is denoted *LSI*.

We will now shortly define MSO definability of a tree translation. This definition will, however, not be needed in the paper. Let k be the maximal rank of a symbol in Δ. The tree translation $\tau : T_\Sigma \to T_\Delta$ is *MSO definable* (i.e., definable in monadic second-order logic) if there is an *MSO tree transducer* which realizes τ, that is, if there exist a finite set C and MSO(Σ)-formulas $\nu_c(x)$, $\psi_{\delta,c}(x)$, and $\chi_{i,c,d}(x, y)$, with $c, d \in C$, $\delta \in \Delta$, and $1 \leq i \leq k$, such that for every $s \in T_\Sigma$, $\tau(s) \in T_\Delta$ is isomorphic to the tree t with set of nodes $\{(c, x) \in C \times V(s) \mid s \models \nu_c(x)\}$, node (c, x) has label δ if and only if $s \models \psi_{\delta,c}(x)$, and (d, y) is the ith child of (c, x) if and only if $s \models \chi_{i,c,d}(x, y)$. An MSO($\Sigma$)-formula is a formula of MSO logic that uses atomic formulas $\mathrm{lab}_\sigma(x)$ and $\mathrm{child}_i(x, y)$, with $\sigma \in \Sigma$ and $i \geq 1$, to express that x has label σ and y is the ith child of x, respectively. The class of all MSO definable tree translations is denoted *MSOTT*. For examples and more details, see, e.g., [5, 3]. Note that, by definition, every MSO definable tree translation τ is of linear size increase: $\mathrm{size}(\tau(s)) \leq |C| \cdot \mathrm{size}(s)$. Thus, *MSOTT* $\subseteq$ *LSI*.

3. Macro tree transducers. In this section we recall the definition of MTTs and some basic lemmas about them. Furthermore, we consider two subclasses of MTTs which are defined by certain (static) restrictions on the rules of the transducers.

3.1. Basic definitions and results. A macro tree transducer is a syntax-directed translation device in which the translation of an input tree may depend on its subtrees, represented by input variables $x_1, x_2, \dots$, and on its context, represented by parameters $y_1, y_2, \dots$. A rule of an MTT is of the form $\langle q, \sigma(x_1, \dots, x_k)\rangle(y_1, \dots, y_m) \to \zeta$, where q is a state of rank m and σ is an input symbol of rank k. The left-hand side will be viewed as a tree with a root, labeled $\langle q, \sigma(x_1, \dots, x_k)\rangle$, and m children, labeled

$y_1, \ldots, y_m$, respectively. The right-hand side ζ is a tree over output symbols, the parameters $y_1, \ldots, y_m$, and symbols $\langle q', x_i \rangle$, where q' is a (ranked) state and $1 \leq i \leq k$. Thus, the MTT generalizes the top-down tree transducer in the sense that states have parameters. Note that symbols $\langle q', x_i \rangle$ can occur at any node of ζ, not just at leaves as in the case of top-down tree transducers.

We consider only *total deterministic* MTTs. For technical reasons we add the feature of regular look-ahead to them (this does not change the class of translations; cf. Theorem 4.21 of [24]). Regular look-ahead means that in a rule as above the input variables $x_1, \ldots, x_k$ range over regular tree languages. Formally, these regular tree languages are combined into one finite state tree automaton.

DEFINITION 3.1 (MTT with regular look-ahead). *A* macro tree transducer with regular look-ahead *(for short, MTT$^{\mathrm{R}}$) is a tuple $M = (Q, P, \Sigma, \Delta, q_0, R, h)$, where Q is a ranked alphabet of* states, *Σ and Δ are ranked alphabets of* input *and* output symbols, *respectively, $\Delta \cap Y = \varnothing$, $q_0 \in Q^{(0)}$ is the* initial state, *(P, Σ, h) is a finite state tree automaton, called the* look-ahead automaton of *M, and R is a finite set of* rules *of the following form: For every $q \in Q^{(m)}$, $\sigma \in \Sigma^{(k)}$, and $p_1, \ldots, p_k \in P$ with $m, k \geq 0$ there is exactly one rule of the form*

$$(*) \qquad \langle q, \sigma(x_1, \ldots, x_k) \rangle (y_1, \ldots, y_m) \to \zeta \quad \langle p_1, \ldots, p_k \rangle$$

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

A rule r of the form $(*)$ is called the $(q, \sigma, \langle p_1, \ldots, p_k \rangle)$-rule and its right-hand side ζ is denoted by $\mathrm{rhs}(r)$ or by $\mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k \rangle)$; it is also called a q-rule, a σ-rule, or a (q, σ)-rule. A *top-down tree transducer with regular look-ahead* (for short, T^{R}) is an $\mathrm{MTT}^{\mathrm{R}}$ all states of which are of rank zero. If the look-ahead automaton is trivial, i.e., $P = \{p\}$ and $h_\sigma(p, \ldots, p) = p$ for all $\sigma \in \Sigma$, then M is called an *MTT*, and if M is a T^{R}, then M is called a *top-down tree transducer*. In such cases we omit the look-ahead automaton and simply denote M by $(Q, \Sigma, \Delta, q_0, R)$; we also omit the look-ahead part $\langle p_1, \ldots, p_k \rangle$ in every rule $(*)$.

We now define the derivation relation induced by an $\mathrm{MTT}^{\mathrm{R}}$ M. Recall from section 2.2 that in a second-order tree substitution $\langle q', x_i \rangle \leftarrow \langle q', s_i \rangle$ is shorthand for $\langle q', x_i \rangle \leftarrow \langle q', s_i \rangle (y_1, \ldots, y_n)$, where n is the rank of q'.

DEFINITION 3.2 (derivation relation). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT$^{\mathrm{R}}$. 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 $u \in V(\xi_1)$, $\sigma \in \Sigma^{(k)}$, $s_1, \ldots, s_k \in T_\Sigma$, $q \in Q^{(m)}$, and $t_1, \ldots, t_m \in T_{\langle Q, T_\Sigma \rangle \cup \Delta}(Y)$ such that $\xi_1/u = \langle q, \sigma(s_1, \ldots, s_k) \rangle (t_1, \ldots, t_m)$ and $\xi_2 = \xi_1[u \leftarrow \zeta]$, where ζ equals*

$$\mathrm{rhs}_M(q, \sigma, \langle h(s_1), \ldots, h(s_k) \rangle) [\![\langle q', x_i \rangle \leftarrow \langle q', s_i \rangle \mid \langle q', x_i \rangle \in \langle Q, X_k \rangle]\!] [y_j \leftarrow t_j \mid j \in [m]].$$

Since the derivation relation $\Rightarrow_M$ induced by M is confluent and terminating (cf., e.g., Chapter 4 of [28]) there is, for every tree $\xi \in T_{\langle Q, T_\Sigma \rangle \cup \Delta}(Y_m)$, a unique tree $t \in T_\Delta(Y_m)$ such that $\xi \Rightarrow_M^* t$ (in fact, t is the normal form of ξ with respect to $\Rightarrow_M$).

DEFINITION 3.3 (translation). *For every $q \in Q^{(m)}$ and $s \in T_\Sigma$ the* q-translation of *s, denoted by $M_q(s)$, is the unique tree $t \in T_\Delta(Y_m)$ such that $\langle q, s \rangle (y_1, \ldots, y_m) \Rightarrow_M^* t$. Thus, for $q \in Q^{(m)}$, M_q is a (total) function from T_Σ to $T_\Delta(Y_m)$. The* translation realized by *M, denoted by τ_M, is the (total) function $M_{q_0} : T_\Sigma \to T_\Delta$.*

Thus, $\tau_M = M_{q_0}$ and for $s \in T_\Sigma$, $\tau_M(s) = M_{q_0}(s)$ is the unique tree $t \in T_\Delta$ such that $\langle q_0, s \rangle \Rightarrow_M^* t$. If $\tau_M(s) = t$, then s is an *input tree* (of M) and t is the corresponding *output tree* (of M).

An $\mathrm{MTT}^{\mathrm{R}}$ is *of linear size increase* (for short, lsi) if τ_M is lsi (cf. section 2.5). Two $\mathrm{MTT}^{\mathrm{R}}$s M and M' are *equivalent* if $\tau_M = \tau_{M'}$. The class of all translations which can be realized by MTTs and $\mathrm{MTT}^{\mathrm{R}}$s is denoted by MTT and MTT^R, respectively. The class of all translations which can be realized by T^{R}s is denoted by T^R.

LEMMA 3.4 (Theorem 4.21 of [24]). *$MTT^R = MTT$ (effectively).*

The q-translations $M_q(s)$ of trees $s \in T_\Sigma$ can be characterized inductively as follows, using second-order tree substitution.

LEMMA 3.5 (Lemma 4.8 of [26]). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R. For every $q \in Q$, $\sigma \in \Sigma^{(k)}$, $k \geq 0$, and $s_1, \dots, s_k \in T_\Sigma$,*

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

As mentioned in the introduction, macro tree translations can be of double exponential size increase. This is shown in the following example.

Example 3.6. Let $M = (Q, \Sigma, \Delta, q_0, R)$ be the MTT with $Q = \{q_0^{(0)}, q^{(1)}\}$, $\Sigma = \{\sigma^{(1)}, \alpha^{(0)}\}$, $\Delta = \{\delta^{(2)}, \beta^{(0)}\}$, and R consisting of the following rules:

$$\begin{array}{lcl} \langle q_0, \sigma(x_1)\rangle & \to & \langle q, x_1\rangle(\beta), \\ \langle q_0, \alpha\rangle & \to & \beta, \\ \langle q, \sigma(x_1)\rangle(y_1) & \to & \langle q, x_1\rangle(\langle q, x_1\rangle(y_1)), \\ \langle q, \alpha\rangle(y_1) & \to & \delta(y_1, y_1). \end{array}$$

The MTT M translates α into β, and for $n \geq 0$ it translates the input tree $s_n = \sigma(\sigma^n(\alpha))$ into a full binary tree of height $2^n + 1$ (with 2^{2^n} leaves). Figure 3.1 shows a derivation of M: First $\langle q_0, s_n\rangle \Rightarrow_M \langle q, \sigma^n(\alpha)\rangle(\beta)$. Then, due to the copying of states of the (q,σ)-rule, $\langle q, \sigma^n(\alpha)\rangle(\beta)$ is translated into the monadic tree $\langle q,\alpha\rangle(\langle q,\alpha\rangle(\cdots\langle q,\alpha\rangle(\beta)\cdots))$ containing 2^n occurrences of $\langle q, \alpha\rangle$. At last, due to the copying of parameters of the (q,α)-rule, this monadic tree is translated into a full binary tree of height $2^n + 1$. Thus, the input tree s_n of size $n+2$ is translated into a tree of size $2^{2^n+1} - 1$, and hence the translation realized by M is of double exponential size increase. Note that the q-translation $M_q(\sigma^n(\alpha)) \in T_\Delta(\{y_1\})$ is the full binary tree of height 2^n+1 in which each leaf is labeled y_1, denoted t_n in what follows. In fact, a derivation $\langle q, \sigma^n(\alpha)\rangle(y_1) \Rightarrow_M \langle q, \sigma^{n-1}(\alpha)\rangle(\langle q, \sigma^{n-1}(\alpha)\rangle(y_1)) \Rightarrow_M^* t_n$ can be obtained from Figure 3.1 by removing the first derivation step and changing every β into y_1. Another way of showing that $M_q(\sigma^n(\alpha)) = t_n$ is by induction on n, using Lemma 3.5. For $n = 0$, $M_q(\alpha) = \mathrm{rhs}_M(q,\alpha) = \delta(y_1,y_1) = t_0$. The induction step is proved as follows: $M_q(\sigma(\sigma^n(\alpha))) = \mathrm{rhs}_M(q,\sigma)[\![\langle q,x_1\rangle \leftarrow M_q(\sigma^n(\alpha))]\!] = \langle q,x_1\rangle(\langle q,x_1\rangle(y_1))[\![\langle q,x_1\rangle \leftarrow t_n]\!] = t_n[y_1 \leftarrow t_n] = t_{n+1}$. This ends the example.

The following two results are often used in this paper.

LEMMA 3.7 (Lemma 7.4(1) of [24]). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R. For every $q \in Q^{(m)}$, $m \geq 0$, and regular tree language $L \subseteq T_\Delta(Y_m)$, $M_q^{-1}(L)$ is regular and can be defined effectively.*

Proof. In Lemma 7.4(1) of [24] the result is stated for the case $m = 0$. The general case can be reduced to this case as follows: For every $r \in Q$ let $\overline{r}$ be a symbol not in Σ. Define the $\mathrm{MTT}^{\mathrm{R}}$ $\overline{M} = (Q, P, \Sigma \cup \{\overline{r}^{(1)} \mid r \in Q\}, \Delta \cup \{\overline{y}_j^{(0)} \mid j \in [\overline{m}]\}, q_0, R \cup \overline{R}, h \cup \overline{h})$, where $\overline{m}$ is the maximal rank of a state of M. For every $r \in Q^{(n)}$, $n \geq 0$, and $p \in P$ let $\overline{h}_{\overline{r}}(p) = p$, and let the rule

$$\langle q_0, \overline{r}(x_1)\rangle \to \langle r, x_1\rangle(\overline{y}_1, \dots, \overline{y}_n) \quad \langle p\rangle$$

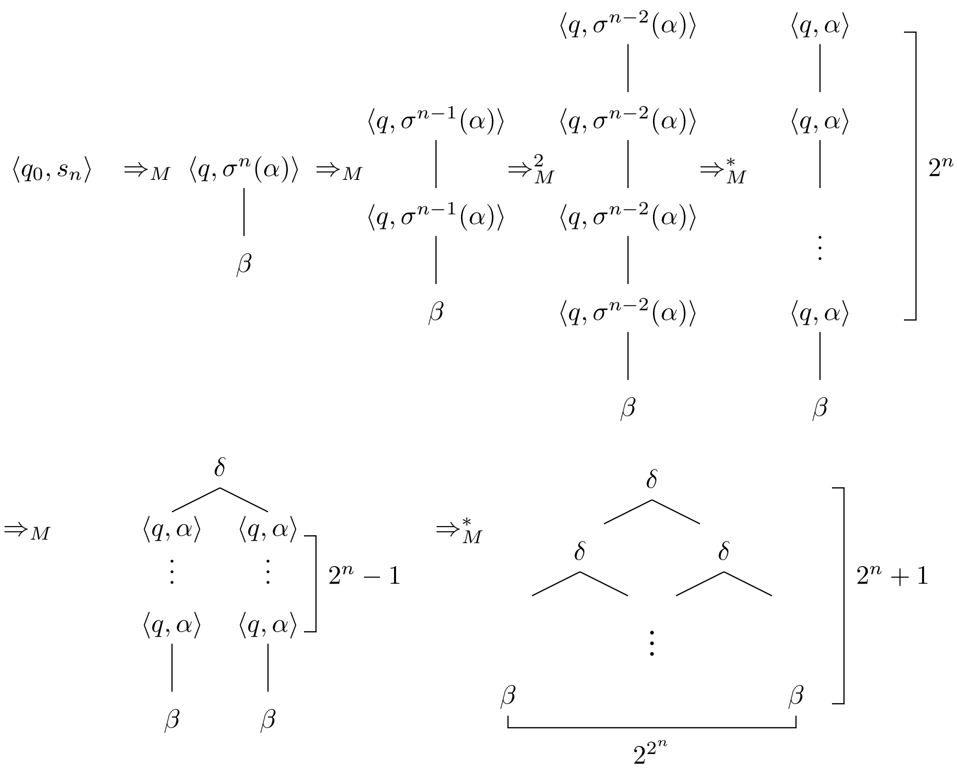

FIG. 3.1. *Derivation by* $\Rightarrow_M$.

be in $\overline{R}$. Clearly, $\tau_{\overline{M}}(\overline{r}(s)) = M_r(s)[y_j \leftarrow \overline{y}_j \mid j \in [n]]$ for every $s \in T_\Sigma$. Let $\overline{L} = \{t[y_j \leftarrow \overline{y}_j \mid j \in [m]] \mid t \in L\}$. By Lemma 7.4(1) of [24], $\tau_{\overline{M}}^{-1}(\overline{L})$ is (effectively) regular. Then also $\tau_{\overline{M}}^{-1}(\overline{L}) \cap \overline{q}(T_\Sigma) = \overline{q}(M_q^{-1}(L))$ is (effectively) regular (because regular tree languages are effectively closed under intersection; cf., e.g., Theorem II.4.2 of [31]). Since there is a linear top-down tree transducer that translates each tree $\overline{q}(t)$ into the tree t, and regular tree languages are (effectively) closed under linear top-down tree translations (see, e.g., Corollary IV.6.6 of [31]), we obtain that $M_q^{-1}(L)$ is (effectively) regular. $\square$

The next lemma follows from Theorem 4.5 of [12] and Theorem 7.3 of [24] (and from the obvious fact that every regular tree language is the range of a nondeterministic top-down tree transducer; cf., e.g., Proposition 20.1(ii) of [32]). Note that we have not defined nondeterministic MTT$^{\mathrm{R}}$s and that we need to apply Lemma 3.8 only once to a nondeterministic (top-down) tree transducer (in Lemma 5.7).

LEMMA 3.8 (Theorem 4.5 of [12]). *For a regular tree language L and a finite number of (possibly nondeterministic) MTT$^{\mathrm{R}}$s $M_1, \ldots, M_n$ it is decidable whether or not $\tau_{M_n}(\tau_{M_{n-1}}(\cdots \tau_{M_1}(L)\cdots))$ is finite. Moreover, if it is finite, it can be constructed.*

3.2. Subclasses defined by restrictions on the parameters. We now define two restrictions on the occurrences of parameters in the right-hand sides of the rules of an MTT$^{\mathrm{R}}$ M and then show that these restrictions carry over to the q-translations $M_q(s)$ of M.

DEFINITION 3.9 (nondeleting, nonerasing, productive). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^{R}. If for every $q \in Q^{(m)}$, $m \geq 1$, $\sigma \in \Sigma^{(k)}$, $k \geq 0$, $p_1, \ldots, p_k \in P$, and $j \in [m]$,*

- *y_j occurs at least once in* $\mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)$*, then M is* nondeleting*;*
- $\mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle) \notin Y$*, then M is* nonerasing.

If M is both nondeleting and nonerasing, then it is productive.

LEMMA 3.10 (Lemma 7.11 of [19]). *For every MTT^{R} M there is a productive MTT^{R} M' equivalent to M.*

The following lemma shows that the restrictions nondeleting and nonerasing carry over from the right-hand sides of an MTT$^{\mathrm{R}}$ to the q-translations of M. In Lemma 6.7 of [19] a similar result is proved: if in the right-hand side of every q-rule each parameter y_j of q occurs exactly once, then y_j occurs exactly once in $M_q(s)$.

LEMMA 3.11. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^{R}. For every $q \in Q^{(m)}$, $m \geq 0$, $j \in [m]$, and $s \in T_\Sigma$,*

(1) *if M is nondeleting, then $\#_{y_j}(M_q(s)) \geq 1$; and*

(2) *if M is nonerasing, then $M_q(s) \notin Y$.*

Proof. The proof is by induction on the structure of s. Let $s = \sigma(s_1, \ldots, s_k)$ with $k \geq 0$ and $s_1, \ldots, s_k \in T_\Sigma$. Denote by t the tree $\mathrm{rhs}_M(q, \sigma, \langle h(s_1), \ldots, h(s_k)\rangle)$. By Lemma 3.5, $M_q(s) = t\Phi$ with $\Phi = [\![\langle q', x_i\rangle \leftarrow M_{q'}(s_i) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!]$.

(1) By induction $\#_{y_\nu}(M_{q'}(s_i)) \geq 1$ for all $\langle q', x_i\rangle \in \langle Q, X_k\rangle^{(n)}$ and $\nu \in [n]$; i.e., the substitution Φ is nondeleting. Since M is nondeleting, $\#_{y_j}(t) \geq 1$ and thus, by Lemma 2.1, $\#_{y_j}(t\Phi) \geq 1$.

(2) By induction $M_{q'}(s_i) \notin Y$ for all $\langle q', x_i\rangle \in \langle Q, X_k\rangle$; i.e., the substitution Φ is nonerasing. Since M is nonerasing, $t \notin Y$ and thus, by Lemma 2.2, $t\Phi \notin Y$. ∎

4. Finite copying restrictions. In this section we define various restrictions on the copying that is performed by an MTT$^{\mathrm{R}}$. First, in section 4.1, copying restrictions for the input variables and for the parameters are defined. Both together form the "finite copying" restriction which was introduced in [19]; there it was shown (in Theorem 7.1) that the translations realized by finite copying MTT$^{\mathrm{R}}$s are precisely the MSO definable tree translations (cf. section 2.5). Since, by their definition, the MSO definable tree translations are lsi, this means that finite copying MTT$^{\mathrm{R}}$s are lsi. To keep this paper self-contained, we give, in section 4.3, a direct proof of this fact, which is based on the notion of "finite contribution." Intuitively, an MTT$^{\mathrm{R}}$ is of finite contribution if there is a bound on the number of output nodes contributed by a single node u of the input tree. In the terminology of [50], the node u is called the "origin" of the nodes of the output tree that it contributes; so, finite contribution means that there is a bound on the number of nodes that have the same origin. In [50] it is shown that for a primitive recursive scheme, which is an MTT, every node of an output tree has exactly one origin.

We also define, in section 4.2, a restriction on the copying that occurs on one path of the output tree, i.e., a restriction on the amount of nesting of states that occurs during the derivation of an MTT$^{\mathrm{R}}$. This notion will play an essential role in section 6, where it is proved that if the translation of an MTT$^{\mathrm{R}}$ is lsi, then it can also be realized by a finite copying MTT$^{\mathrm{R}}$ (and hence is MSO definable).

4.1. Finite copying in the input and in the parameters. Here we recall the definition of finite copying MTT$^{\mathrm{R}}$s from [19] and show that for an MTT$^{\mathrm{R}}$ it is decidable whether or not it is finite copying. The finite copying restriction was introduced in [1] for generalized syntax-directed translation schemes. For top-down tree transducers it was investigated in [22]. A top-down tree transducer is finite copying if every subtree of the input tree is translated by boundedly many occurrences of states, i.e., the length of the state sequence is bounded, where the state sequence at a subtree s/u consists of the states that translate s/u. For an MTT this restriction is called finite copying in the input (fci), and we additionally have a restriction for the parameters, called finite copying in the parameters (fcp). The fcp restriction requires that, for every state q and input tree s, the number of parameters that occur in the q-translation $M_q(s)$ of s is bounded.

In order to define the state sequence of a tree s at the node u of s, we first extend an MTT$^{\mathrm{R}}$ in such a way that the output tree t, for the input tree $s[u \leftarrow p]$, contains the states which process the subtree s/u (assuming that $p = h(s/u)$). More precisely, t contains $\langle\!\langle q, p\rangle\!\rangle$ if the state q translates s/u. Analogous to the definition of $\langle \Sigma, A\rangle$ let, for a ranked set Σ and a set A, $\langle\!\langle \Sigma, A\rangle\!\rangle$ be the ranked set of all symbols $\langle\!\langle \sigma, a\rangle\!\rangle$ of rank m for $\sigma \in \Sigma^{(m)}$ and $a \in A$.

DEFINITION 4.1 (Definition 3.5 of [19]: extension). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R. The* extension of M, *denoted by $\hat{M}$, is the MTT^R $(Q, P, \hat{\Sigma}, \hat{\Delta}, q_0, \hat{R}, \hat{h})$, where $\hat{\Sigma} = \Sigma \cup \{p^{(0)} \mid p \in P\}$, $\hat{\Delta} = \Delta \cup \langle\!\langle Q, P\rangle\!\rangle$, $\hat{R} = R \cup \{\langle q,p\rangle(y_1,\ldots,y_m) \rightarrow \langle\!\langle q,p\rangle\!\rangle(y_1,\ldots,y_m) \mid \langle q,p\rangle \in \langle Q,P\rangle^{(m)}\}$, $\hat{h}_p() = p$ for $p \in P$, and $\hat{h}_\sigma(p_1,\ldots,p_k) = h_\sigma(p_1,\ldots,p_k)$ for $\sigma \in \Sigma^{(k)}$, $k \geq 0$, and $p_1,\ldots,p_k \in P$.*

Note that if M is nondeleting or nonerasing, then so is $\hat{M}$. Before state sequences and the fci and fcp properties are defined, we present two useful lemmas about the q-translations of $\hat{M}$. The first lemma shows that the q-translation of an input tree s can be obtained by replacing in the q-translation of the "context" of a node u of s, $\hat{M}_q(s[u \leftarrow p])$, each occurrence of $\langle\!\langle q', p\rangle\!\rangle$ by the q'-translation $M_{q'}(s/u)$ of the subtree of s at u. In fact, the lemma is stated in the more general case that s/u may contain occurrences of symbols in P. The lemma can be seen as a generalization of Lemma 3.5 from the application of a rule at the root of s, to the translation of the context of an arbitrary node u.

LEMMA 4.2 (Lemma 3.6 of [19]). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R and $\hat{M} = (Q, P, \hat{\Sigma}, \hat{\Delta}, q_0, \hat{R}, \hat{h})$ its extension. Let $q \in Q$, $s \in T_{\hat{\Sigma}}$, $u \in V(s)$, and $p = \hat{h}(s/u)$ such that $s[u \leftarrow p]$ contains exactly one occurrence of an element of P. Then*

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

The next lemma is obtained by application of Lemma 3.5 to the $\hat{M}_{q'}(s/u)$ in the substitution of Lemma 4.2. It shows how to express the translation of the context of a child node in terms of the translation of the context of its parent and the translations of the subtrees of its siblings.

LEMMA 4.3. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R. Let $q \in Q$, $s \in T_\Sigma$, and $u \in V(s)$. If $s[u] = \sigma \in \Sigma^{(k)}$, $i \in [k]$, $p_i \in P$, $p_j = h(s/uj)$ for every $j \in [k] - \{i\}$, and $p = h_\sigma(p_1,\ldots,p_k)$, then*

$$\hat{M}_q(s[ui \leftarrow p_i]) = \hat{M}_q(s[u \leftarrow p])[\![rhs]\!][\![..]\!][\![i]\!],$$

where

$$\begin{array}{rcl}[\![\mathrm{rhs}]\!] & = & [\![\langle\!\langle q', p\rangle\!\rangle \leftarrow \mathrm{rhs}_M(q', \sigma, \langle p_1,\ldots,p_k\rangle) \mid q' \in Q]\!],\\ [\![..]\!] & = & [\![\langle r, x_j\rangle \leftarrow M_r(s/uj) \mid r \in Q, j \in [k] - \{i\}]\!], \text{ and}\\ [\![i]\!] & = & [\![\langle r, x_i\rangle \leftarrow \langle\!\langle r, p_i\rangle\!\rangle \mid r \in Q]\!].\end{array}$$

Proof. Let $s' = s[ui \leftarrow p_i]$. Since $p = \hat{h}(s'/u)$ and $s'[u \leftarrow p]$ contains exactly one occurrence of an element of P, we can apply Lemma 4.2 to get $\hat{M}_q(s') = \hat{M}_q(s[u \leftarrow$

$p])[\![\langle\!\langle q',p\rangle\!\rangle \leftarrow \hat{M}_{q'}(s'/u) \mid q' \in Q]\!]$. Now $s'/u = \sigma(s_1,\dots,s_k)$ with $s_i = p_i$ and $s_j = s/uj$ for every $j \in [k]-\{i\}$. By application of Lemma 3.5 to $\hat{M}_{q'}(s'/u)$ the above equals $\hat{M}_q(s[u \leftarrow p])[\![\langle\!\langle q',p\rangle\!\rangle \leftarrow \mathrm{rhs}_M(q',\sigma,\langle p_1,\dots,p_k\rangle)[\![\dots]\!] \mid q' \in Q]\!]$, where $[\![\dots]\!]$ denotes $[\![\langle r,x_j\rangle \leftarrow \hat{M}_r(s_j) \mid r \in Q, j \in [k]]\!]$. We now use the associativity of second-order tree substitution; cf. section 2.2. Since $\hat{M}_q(s[u \leftarrow p])$ does not contain elements of $\langle Q,X_k\rangle$ we can move $[\![\dots]\!]$ out of the substitution to get $\hat{M}_q(s[u \leftarrow p])[\![\mathrm{rhs}]\!][\![\dots]\!]$. For every $j \in [k]-\{i\}$, $\hat{M}_r(s_j) = M_r(s_j)$ does not contain elements of $\langle Q,\{x_i\}\rangle$; moreover, $\hat{M}_r(s_i) = \langle\!\langle r,p_i\rangle\!\rangle$. Thus we can write $[\![\dots]\!]$ as $[\![..]\!][\![i]\!]$. ◻

We now turn to the definition of state sequence and the finite copying properties. Recall that the preorder of the nodes of a tree is denoted by $<$.

DEFINITION 4.4 (Definition 3.7 of [19]: state sequence). *Let $M = (Q,P,\Sigma,\Delta,q_0,R,h)$ be an MTT^R, $s \in T_\Sigma$, and $u \in V(s)$. 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 $\{v \in V(\xi) \mid \xi[v] \in \langle\!\langle Q,\{p\}\rangle\!\rangle\} = \{v_1,\dots,v_n\}$ with $v_1 < \cdots < v_n$. The* state sequence *of s at u, denoted by $sts_M(s,u)$, is the sequence of states $q_1\cdots q_n$ such that $\xi[v_i] = \langle\!\langle q_i,p\rangle\!\rangle$ for every $i \in [n]$.*

Observe that $|\mathrm{sts}_M(s,u)| = \#_{\langle\!\langle Q,\{p\}\rangle\!\rangle}(\hat{M}_{q_0}(s[u \leftarrow p]))$, where $p = h(s/u)$.

DEFINITION 4.5 (Definition 6.1 of [19]: fci). *Let $M = (Q,P,\Sigma,\Delta,q_0,R,h)$ be an MTT^R. Then M is* fci *if there is an $N \in \mathbb{N}$ such that for every $s \in T_\Sigma$ and $u \in V(s)$: $|\mathrm{sts}_M(s,u)| \le N$. The number N is an* input copying bound *for M.*

DEFINITION 4.6 (Definition 6.2 of [19]: fcp). *Let $M = (Q,P,\Sigma,\Delta,q_0,R,h)$ be an MTT^R. Then M is* fcp *if there is an $N \in \mathbb{N}$ such that for every $q \in Q^{(m)}$, $s \in T_\Sigma$, and $j \in [m]$, $\#_{y_j}(M_q(s)) \le N$. The number N is a* parameter copying bound *for M.*

Note that the MTT M of Example 3.6 is neither fci nor fcp. There is exponential state copying: the state sequence $\mathrm{sts}_M(s_n,11^n)$ of $s_n = \sigma(\sigma^n(\beta))$ at 11^n equals q^{2^n}, and there is double exponential parameter copying: $\#_{y_1}(M_q(\sigma^n(\beta))) = 2^{2^n}$.

The following lemma shows that if M is fcp, i.e., if the number of occurrences of y_j in $M_q(s)$ is bounded by some N, for all states q and parameters y_j of q, then also for the q-translations of $\hat{M}$ of input trees $s[u \leftarrow p]$, the number of occurrences of y_j is bounded by N. However, we must assume that M is nondeleting.

LEMMA 4.7. *Let $M = (Q,P,\Sigma,\Delta,q_0,R,h)$ be a nondeleting fcp MTT^R and let N be a parameter copying bound for M. For every $q \in Q^{(m)}$, $j \in [m]$, $s \in T_\Sigma$, and $u \in V(s)$, $\#_{y_j}(\hat{M}_q(s[u \leftarrow h(s/u)])) \le N$.*

Proof. Let $p = h(s/u)$. By Lemma 4.2, $M_q(s) = \xi[\![\dots]\!]$ with $\xi = \hat{M}_q(s[u \leftarrow p])$ and $[\![\dots]\!] = [\![\langle\!\langle q',p\rangle\!\rangle \leftarrow M_{q'}(s/u) \mid q' \in Q]\!]$. By Lemma 2.6, $\#_{y_j}(\xi[\![\dots]\!]) = \sum_{v\in V_{y_j}(\xi)} \prod F^{[\![\dots]\!]}_{\xi,v}$. Let $V_{y_j}(\xi) = \{v_1,\dots,v_n\}$. Then the above sum equals

$$\prod F^{[\![\dots]\!]}_{\xi,v_1} + \cdots + \prod F^{[\![\dots]\!]}_{\xi,v_n} = \#_{y_j}(M_q(s)) \le N,$$

which implies that $n = \#_{y_j}(\xi) \le N$ because $\prod F^{[\![\dots]\!]}_{\xi,v_i} \ge 1$ for every $i \in [n]$, by the fact that M is nondeleting, and hence, by Lemma 3.11(1), $\#_{y_k}(M_{q'}(s/u)) \ge 1$ for every $q' \in Q^{(m')}$ and $k \in [m']$. ◻

Finally, the combination of fci and fcp yields the finite copying property.

DEFINITION 4.8 (finite copying). *An MTT^R is* finite copying *if it is both fci and fcp.*

We use the subscripts "fci," "fcp," or "fc" for classes of translations to denote that the corresponding $\mathrm{MTT^R}$s are fci, fcp, or finite copying, respectively. Thus $MTT^R_{\mathrm{fc}} = MTT^R_{\mathrm{fci,fcp}}$. The main result of [19] is that the translations of finite copying $\mathrm{MTT^R}$s are precisely the MSO definable tree translations (see section 2.5).

LEMMA 4.9 (Theorem 7.1 of [19]). *$MSOTT = MTT^R_{fc}$ (effectively).*

The main results of this paper are the following: (i) the translations of finite copying $\mathrm{MTT^R}$s are precisely the translations of $\mathrm{MTT^R}$s that are of linear size increase (i.e., $MTT^R \cap LSI = MTT^R_{\mathrm{fc}}$), and (ii) it is decidable for an $\mathrm{MTT^R}$ M whether or not there exists an equivalent finite copying $\mathrm{MTT^R}$ (i.e., whether $\tau_M \in MTT^R_{\mathrm{fc}}$). We now show that it is decidable for an $\mathrm{MTT^R}$ M whether or not M is finite copying. The proof is based on Lemma 3.8.

LEMMA 4.10. *It is decidable for an MTT^R M*

(i) *whether or not M is fci, and*

(ii) *whether or not M is fcp,*

and if so, a copying bound can be obtained effectively.

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$.

(i) To decide fci we define the MTT N that, when applied to $\tau_{\hat{M}}(s[u \leftarrow h(s/u)])$, generates the state sequence $\mathrm{sts}_M(s, u)$ as a monadic tree. Let $N = (Q', \Delta \cup \langle\!\langle Q, P\rangle\!\rangle, \Gamma, r_0, R')$ with $Q' = \{r_0^{(0)}, r^{(1)}\}$ and $\Gamma = \{q^{(1)} \mid q \in Q\} \cup \{e^{(0)}\}$. For every $k \geq 0$, $\langle\!\langle q, p\rangle\!\rangle \in \langle\!\langle Q, P\rangle\!\rangle^{(k)}$, and $\delta \in \Delta^{(k)}$ let the following rules be in R':

$$\begin{array}{lcl}
\langle r_0, \langle\!\langle q,p\rangle\!\rangle(x_1,\ldots,x_k)\rangle & \to & q(\langle r, x_1\rangle(\langle r, x_2\rangle(\cdots \langle r, x_k\rangle(e)\cdots))),\\
\langle r_0, \delta(x_1,\ldots,x_k)\rangle & \to & \langle r, x_1\rangle(\langle r, x_2\rangle(\cdots \langle r, x_k\rangle(e)\cdots)),\\
\langle r, \langle\!\langle q,p\rangle\!\rangle(x_1,\ldots,x_k)\rangle(y_1) & \to & q(\langle r, x_1\rangle(\langle r, x_2\rangle(\cdots \langle r, x_k\rangle(y_1)\cdots))),\\
\langle r, \delta(x_1,\ldots,x_k)\rangle(y_1) & \to & \langle r, x_1\rangle(\langle r, x_2\rangle(\cdots \langle r, x_k\rangle(y_1)\cdots)).
\end{array}$$

Then, for every $s \in T_\Sigma$ and $u \in V(s)$, $\mathrm{lpath}(\tau_N(\tau_{\hat{M}}(s[u \leftarrow h(s/u)])), v) = \mathrm{sts}_M(s, u)e$, where v is the unique leaf of $\tau_N(\tau_{\hat{M}}(s[u \leftarrow h(s/u)]))$.

Let L be the tree language $\{s[u \leftarrow h(s/u)] \mid s \in T_\Sigma, u \in V(s)\}$. Then M is fci if and only if $K = \tau_N(\tau_{\hat{M}}(L))$ is finite. Note that $L = \{s \in T_\Sigma(P') \mid \#_{P'}(s) = 1\}$, where $P' = \{p \in P \mid L_p \neq \varnothing\}$; hence L is (effectively) regular. Thus, finiteness of K can be decided by Lemma 3.8; in case of finiteness, K can be constructed and an input copying bound for M is $\max\{\mathrm{size}(t) \mid t \in K\} - 1$.

(ii) Let $\overline{M}$ be the $\mathrm{MTT^R}$ defined in the proof of Lemma 3.7 and let $\overline{\Delta} = \Delta \cup \{\overline{y}_j^{(0)} \mid j \in [\overline{m}]\}$ be its output alphabet, where $\overline{m}$ is the maximal rank of a state of M. Let $N = (\{r_0^{(0)}, r^{(1)}\}, \overline{\Delta}, \Gamma, r_0, R_N)$ be the MTT with $\Gamma = \{\overline{y}_j^{(1)} \mid j \in [\overline{m}]\} \cup \{e^{(0)}\}$. For $\delta \in \Delta^{(k)}$ with $k \geq 0$ the (r_0, δ)- and (r, δ)-rules are defined as for N in (i). For $j \in [\overline{m}]$ let the rules $\langle r_0, \overline{y}_j\rangle \to \overline{y}_j(e)$ and $\langle r, \overline{y}_j\rangle(y_1) \to \overline{y}_j(y_1)$ be in R_N.

Clearly, for every $q \in Q$ and $s \in T_\Sigma$, $\mathrm{size}(\tau_N(\tau_{\overline{M}}(\overline{q}(s)))) = 1 + \#_Y(M_q(s))$. Now, for the regular tree language $L = \{\overline{q}(s) \mid q \in Q, s \in T_\Sigma\}$, M is fcp if and only if $K = \tau_N(\tau_{\overline{M}}(L))$ is finite. As in (i), this can be decided by Lemma 3.8; in case of finiteness, K can be constructed and a parameter copying bound for M is $\max\{\mathrm{size}(t) \mid t \in K\} - 1$. ∎

In fact, the effectiveness of Lemma 4.9 was not completely proved in [19], but with Lemma 4.10 it can be shown as follows: Given an $\mathrm{MTT^R_{fc}}$ M we can use Lemma 4.10 to obtain a parameter copying bound N for M. Then, given M and N we can, by the proof of Lemma 6.3 of [19], construct an $\mathrm{MTT^R_{fci,surp}}$ M' equivalent to M (where "surp" means "single-use restricted in the parameters"). Now, again by Lemma 4.10 we can determine an input copying bound N for M'. Then, given M' and N we can, by the proof of Lemma 6.10 of [19], construct a single-use restricted $\mathrm{MTT^R}$ M'' equivalent to M'. Now by the proofs of Lemmas 5.9, 5.12, and 4.1 of [19], a single-use restricted attributed tree transducer with look-ahead (for short, $\mathrm{ATT^R}$) A equivalent to M'' can be constructed. Given A, the proof of Lemma 7 of [3] shows how to construct an equivalent MSO tree transducer. This proves the effectiveness going from $MTT^R_{\rm fc}$ to $MSOTT$. For the other direction, that is, starting with an MSO tree transducer M, we can proceed as follows: The proof of Theorem 14 of [3] gives a construction of an equivalent single-use restricted $\mathrm{ATT^R}$ A. The proofs of Lemmas 4.2 and 5.11 of [19] show how to construct an equivalent single-use restricted $\mathrm{MTT^R}$ M'. By the proof of Theorem 6.12 of [19], M' is finite copying.

4.2. Finite nested copying in the input. Consider the translation $\xi = \hat{M}_{q_0}(s[u \leftarrow p])$ of the context of a node u of the input tree s, where $p = h(s/u)$. The symbols of $\langle\langle Q, \{p\}\rangle\rangle$ can occur nested in ξ; i.e., they can occur on a common label path $\mathrm{lpath}(\xi, v)$ to some node v of ξ. Assuming that M is nondeleting, this means that a lot of copies of v will be generated; namely, $\prod F^{[\![\ldots]\!]}_{\xi,v}$ copies, where $[\![\ldots]\!]$ replaces $\langle\langle q,p\rangle\rangle$ by $M_q(s/u)$. Thus, a way to bound the copying carried out by M is to bound by some $B \in \mathbb{N}$ the number of elements of $\langle\langle Q, \{p\}\rangle\rangle$ that occur on a label path in ξ, i.e., to bound the nesting of states. This implies that the number of elements in the family $F^{[\![\ldots]\!]}_{\xi,v}$ is bounded by B. We call this property *finite nested copying in the input* (for short, fnest). Clearly, it is a much weaker restriction than the fci restriction. However, if an $\mathrm{MTT^R}$ is fnest and fcp, then $\prod F^{[\![\ldots]\!]}_{\xi,v}$ is bounded by N^B if N is a parameter copying bound for M.

DEFINITION 4.11 (fnest). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R. Then M is* fnest *if there is a $B \in \mathbb{N}$ such that for every $s \in T_\Sigma$, $u \in V(s)$, $p = h(s/u)$, and label path π in $\hat{M}_{q_0}(s[u \leftarrow p])$, $\#_{\langle\langle Q,\{p\}\rangle\rangle}(\pi) \leq B$. The number B is a* nesting bound *for M.*

We use the subscript "fnest" for classes of translations of $\mathrm{MTT^R}$s to denote that the corresponding transducers are fnest. The next lemma shows that the nesting bound B also holds for trees $\hat{M}_q(s[u \leftarrow p])$ with $s \in L_{p'}$, provided that $\langle\langle q,p'\rangle\rangle$ is reachable in the following sense.

DEFINITION 4.12 (reachable). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R, $q \in Q$, and $p \in P$. Then $\langle\langle q,p\rangle\rangle$ is* reachable *if there are $s \in T_\Sigma$ and $u \in V(s)$ such that $\langle\langle q,p\rangle\rangle$ occurs in $\hat{M}_{q_0}(s[u \leftarrow p])$.*

Note that reachability does not require that $h(s/u) = p$; however, for $L_p \neq \varnothing$ this can always be assumed (simply take $s' = s[u \leftarrow t]$ for some $t \in L_p$ if $h(s/u) \neq p$). Note that in that case, q occurs in the state sequence of s at u.

LEMMA 4.13. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be a nondeleting fnest MTT^R and let B be a nesting bound for M. If $\langle\langle q,p\rangle\rangle \in \langle\langle Q,P\rangle\rangle$ is reachable, then for every $s \in L_p$, $u \in V(s)$, $p_u = h(s/u)$, and label path π in $\hat{M}_q(s[u \leftarrow p_u])$, $\#_{\langle\langle Q,\{p_u\}\rangle\rangle}(\pi) \leq B$.*

Proof. Since $\langle\langle q,p\rangle\rangle$ is reachable, there are $t \in T_\Sigma$, $v \in V(t)$, and $\rho \in V_{\langle\langle q,p\rangle\rangle}(\hat{M}_{q_0}(t[v \leftarrow p]))$. We may assume that $t/v = s$ and hence $t/vu = s/u$. By Lemma 4.2, $\hat{M}_{q_0}(t[vu \leftarrow p_u]) = \hat{M}_{q_0}(t[v \leftarrow p])[\![\ldots]\!]$ with $[\![\ldots]\!] = [\![\langle\langle q',p\rangle\rangle \leftarrow \hat{M}_{q'}(s[u \leftarrow p_u]) \mid q' \in Q]\!]$. Clearly, $\mathrm{lpath}(\hat{M}_{q_0}(t[v \leftarrow p]), \rho) = w\langle\langle q,p\rangle\rangle$ for some $w \in (\langle\langle Q,\{p\}\rangle\rangle \cup \Delta)^*$. Since M is nondeleting (and hence so is $\hat{M}$), the substitution $[\![\ldots]\!]$ is nondeleting by Lemma 3.11(1), and thus, by Lemma 2.3(ii), there is a $w' \in (\langle\langle Q,\{p_u\}\rangle\rangle \cup \Delta)^*$ such that $w'\pi$ is a label path in $\hat{M}_{q_0}(t[v \leftarrow p])[\![\ldots]\!]$, i.e., in $\hat{M}_{q_0}(t[vu \leftarrow p_u])$. Now, $\#_{\langle\langle Q,\{p_u\}\rangle\rangle}(\pi) \leq \#_{\langle\langle Q,\{p_u\}\rangle\rangle}(w'\pi)$, which is $\leq B$, because B is a nesting bound for M. ∎

Consider a nondeleting $\mathrm{MTT^R}$ M and an input tree $s \in T_\Sigma$. In section 6 we will often be interested in the part of s that lies between two nodes u and v of s, where v is a descendant of u; this part can be represented by the tree $s/u[v' \leftarrow p_v]$, where $v = uv'$ and $p_v = h(s/v)$. The shaded region in Figure 4.1 shows such a part of s.

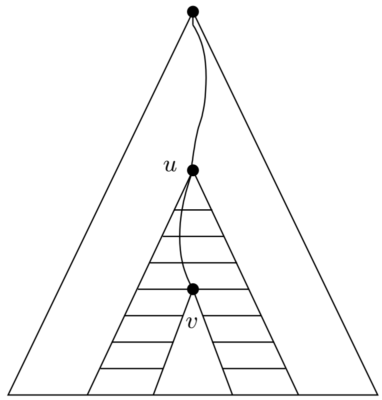

FIG. 4.1. *The tree s with shaded part $s/u[v' \leftarrow p_v]$.*

In particular, in section 6.2, we will need to know, if a state q of M processes this part, how many times the node v' is processed by a state q', i.e., how many times $\langle\!\langle q', p_v\rangle\!\rangle$ occurs in the tree $\hat{M}_q(s/u[v' \leftarrow p_v])$. If M is nondeleting and w is a node between u and v, i.e., a descendant of u and ancestor of v, then a lower bound for this number is given by summing for all states r the product of the number of occurrences of $\langle\!\langle r, p_w\rangle\!\rangle$ in $\hat{M}_q(s/u[w' \leftarrow p_w])$ and $\#_{\langle\!\langle q',p_v\rangle\!\rangle}(\hat{M}_r(s/w[v'' \leftarrow p_v]))$, where $v = wv''$. This is intuitively true because, due to nondeletion, for each occurrence of $\langle\!\langle r, p_w\rangle\!\rangle$ in $\hat{M}_q(s/u[w' \leftarrow p_w])$ there is in $\hat{M}_q(s/u[v' \leftarrow p_v])$ at least one occurrence of the tree $\hat{M}_r(s/w[v'' \leftarrow p_v])$ (without the parameters), and, due to parameter copying, there could be more than one such occurrence. This is stated in part (i) of the following lemma. Part (ii) of the lemma considers the case that M is fnest and fcp; then we can also give an upper bound for the number of occurrences of $\langle\!\langle q', p_v\rangle\!\rangle$ in $\hat{M}_q(s/u[v' \leftarrow p_v])$, because each occurrence of $\langle\!\langle r, p_w\rangle\!\rangle$ in $\hat{M}_q(s/u[w' \leftarrow p_w])$ can only be copied a bounded number of times.

LEMMA 4.14. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be a nondeleting MTT^R. Let $q, q' \in Q$, $s \in T_\Sigma$, and $u, w, v \in V(s)$ such that u is an ancestor of w and w is an ancestor of v, i.e., $w = uw'$ and $v = wv''$ for some $w', v'' \in \mathbb{N}^*$, and let $v' = w'v''$, $p_w = h(s/w)$, and $p_v = h(s/v)$. Finally, let*

$$S = \sum_{r \in Q} \#_{\langle\!\langle q',p_v\rangle\!\rangle}(\hat{M}_r(s/w[v'' \leftarrow p_v])) \cdot \#_{\langle\!\langle r,p_w\rangle\!\rangle}(\hat{M}_q(s/u[w' \leftarrow p_w])).$$

Then the following two statements hold:

(i) $\#_{\langle\!\langle q',p_v\rangle\!\rangle}(\hat{M}_q(s/u[v' \leftarrow p_v])) \geq S$.

(ii) *If M is fnest and fcp with nesting bound B and parameter copying bound N, and $\langle\!\langle q, h(s/u)\rangle\!\rangle$ is reachable, then $\#_{\langle\!\langle q',p_v\rangle\!\rangle}(\hat{M}_q(s/u[v' \leftarrow p_v])) \leq N^B \cdot S$.*

Proof. Note that for $s' = s/u[v' \leftarrow p_v]$, $\hat{h}(s'/w') = p_w$, $s'[w' \leftarrow p_w] = s/u[w' \leftarrow p_w]$, and $s'/w' = s/w[v'' \leftarrow p_v]$. Hence, by Lemma 4.2 applied to s' and w', $\hat{M}_q(s/u[v' \leftarrow p_v]) = \xi[\![\ldots]\!]$, where $\xi = \hat{M}_q(s/u[w' \leftarrow p_w])$ and $[\![\ldots]\!] = [\![\langle\!\langle r, p_w\rangle\!\rangle \leftarrow \hat{M}_r(s/w[v'' \leftarrow p_v]) \mid r \in Q]\!]$, and thus, by Lemma 2.6,

$$(\times)\quad \#_{\langle\!\langle q',p_v\rangle\!\rangle}(\hat{M}_q(s/u[v' \leftarrow p_v])) = \sum_{\tilde{u} \in V_{\langle\!\langle r,p_w\rangle\!\rangle}(\xi), r \in Q} \#_{\langle\!\langle q',p_v\rangle\!\rangle}(\hat{M}_r(s/w[v'' \leftarrow p_v])) \prod F_{\xi,\tilde{u}}^{[\![\ldots]\!]}.$$

Since M is nondeleting, by Lemma 3.11(1), $\#_{y_j}(\hat{M}_{r'}(s/w[v'' \leftarrow p_v])) \geq 1$ for every $r' \in Q^{(m)}$ and $j \in [m]$. This implies that $\prod F^{[\![\ldots]\!]}_{\xi,\tilde{u}} \geq 1$. Thus, the sum in $(\times)$ is $\geq S$, because $|V_{\langle\!\langle r,p_w\rangle\!\rangle}(\xi)|$ equals $\#_{\langle\!\langle r,p_w\rangle\!\rangle}(\hat{M}_q(s/u[w' \leftarrow p_w]))$. This proves part (i).

For (ii), $\prod F^{[\![\ldots]\!]}_{\xi,\tilde{u}} \leq N^B$, because the number of elements of $\langle\!\langle Q,\{p_w\}\rangle\!\rangle$ that occur in $\mathrm{lpath}(\xi,\tilde{u})$ is $\leq B$ by Lemma 4.13 (using the assumption that $\langle\!\langle q, h(s/u)\rangle\!\rangle$ is reachable) and because, by Lemma 4.7, $\#_{y_j}(\hat{M}_{r'}(s/w[v'' \leftarrow p_v])) \leq N$ for every $r' \in Q^{(m)}$ and $j \in [m]$. Thus, the sum in $(\times)$ is $\leq N^B \cdot \sum_{\tilde{u} \in V_{\langle\!\langle r,p_w\rangle\!\rangle}(\xi), r \in Q} \#_{\langle\!\langle q',p_v\rangle\!\rangle}(\hat{M}_r(s/w[v'' \leftarrow p_v])) = N^B \cdot S$. $\square$

Note that point (ii) of Lemma 4.14 can be strengthened by proving an upper bound of $N^{B-1} \cdot S$ for the number of occurrences of $\langle\!\langle q', p_v\rangle\!\rangle$ in $\hat{M}_q(s/u[v' \leftarrow p_v])$. This is true because in $F^{[\![\ldots]\!]}_{\xi,\tilde{u}}$, the node $\tilde{u}$ itself (which is labeled by $\langle\!\langle r, p_w\rangle\!\rangle$ for some state r) is not taken into account; i.e., only proper ancestors of $\tilde{u}$ that are labeled by elements of $\langle\!\langle Q,\{p_w\}\rangle\!\rangle$ are counted; thus there are at most $B-1$ of them. We decided to leave out the "-1," because in the application of the lemma in the proof of Lemma 6.5 this will make the numbers more readable.

4.3. Finite copying implies linear size increase. In this subsection it is proved that if an $\mathrm{MTT}^{\mathrm{R}}$ is finite copying, then it is lsi. Note that this result is not needed, because it follows from Lemma 4.9 (as discussed in the beginning of this section). The proof uses an intermediate, very natural notion, called *finite contribution*. Intuitively, an $\mathrm{MTT}^{\mathrm{R}}$ M is of finite contribution if there is a bound c on the number of output nodes that are contributed by a node of the input tree. Clearly, if M is of finite contribution, then it is lsi (with bound c). Thus, in order to prove that finite copying implies lsi, it suffices to prove that if M is finite copying, then it is of finite contribution (Lemma 4.18). In fact, since one of the main results of this paper is that $\mathrm{MTT}^{\mathrm{R}}$s of linear size increase realize the same class of translations as finite copying $\mathrm{MTT}^{\mathrm{R}}$s (Theorem 7.2 and Lemma 4.9), it means that this is also the class of translations realized by $\mathrm{MTT}^{\mathrm{R}}$s that are of finite contribution.

In order to compute the contribution by a node of the input tree s, we define an $\mathrm{MTT}^{\mathrm{R}}$ M^s, which keeps in the label of each output node v the corresponding input node u that generated v. More precisely, if Δ is the output alphabet of M, then M^s has output alphabet $\langle \Delta, V(s)\rangle$, and the contribution by the node u of s is the number of symbols in $\langle \Delta, \{u\}\rangle$ that appear in $M^s_{q_0}(s')$, where s' is the "decorated version" of s; i.e., s' is obtained from s by changing, for every node w, its label σ into $\langle \sigma, w\rangle$.

DEFINITION 4.15 (the $\mathrm{MTT}^{\mathrm{R}}$ M^s, decorated version, contribution). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R and $s \in T_\Sigma$. Then $M^s = (Q, P, \langle \Sigma, V(s)\rangle, \langle \Delta, V(s)\rangle, q_0, R^s, h^s)$ is the MTT^R such that for every $\langle \sigma, u\rangle \in \langle \Sigma, V(s)\rangle^{(k)}$, $k \geq 0$, and $p_1, \ldots, p_k \in P$,*

- $h^s_{\langle \sigma, u\rangle}(p_1, \ldots, p_k) = h_\sigma(p_1, \ldots, p_k)$, *and*
- $\mathrm{rhs}_{M^s}(q, \langle \sigma, u\rangle, \langle p_1, \ldots, p_k\rangle) = \mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)[\![\delta \leftarrow \langle \delta, u\rangle \mid \delta \in \Delta]\!]$.

The decorated version of s, *denoted by* $\mathrm{dec}(s)$, *is the unique tree in $T_{\langle \Sigma, V(s)\rangle}$ such that $V(\mathrm{dec}(s)) = V(s)$, and for every $u \in V(s)$, $\mathrm{dec}(s)[u] = \langle s[u], u\rangle$.*

For a node u of s, the set $V_{\langle \Delta,\{u\}\rangle}(M^s_{q_0}(\mathrm{dec}(s))) \subseteq V(M_{q_0}(s))$ is the set of output nodes contributed by u, *and the* contribution by u, *denoted by* $\mathrm{Contrib}_M(s,u)$, *is the cardinality $\#_{\langle \Delta,\{u\}\rangle}(M^s_{q_0}(\mathrm{dec}(s)))$ of this set.*

Note that every output node is contributed by a unique input node u (called its origin in [50]). Before we prove our first lemma about contribution, let us note some easy properties of the $\mathrm{MTT}^{\mathrm{R}}$ M^s. Let $u \in V(s)$ and $q \in Q$.

(P1) $h^s(\mathrm{dec}(s)/u) = h(s/u)$.

(P2) For $s' \in T_{\langle\Sigma,V(s)\rangle}$, $\pi_\Delta(M_q^s(s')) = M_q(\pi_\Delta(s'))$, where π_Δ changes each symbol $\langle\delta, u\rangle$ into δ; i.e., it is the canonical projection from $\langle\Delta, V(s)\rangle$ to Δ. For $\hat{M}^s$ and $\hat{M}$ a similar statement holds.
Additionally, note the following two obvious facts about the projection π_Δ. Let Ω be a ranked alphabet disjoint with $\langle\Delta, V(s)\rangle$, $\xi \in T_{\Omega\cup\langle\Delta,V(s)\rangle}(Y)$, and $\xi' \in T_{\Omega\cup\langle\Delta,\{u\}\rangle}(Y)$. We assume that π_Δ is the identity on elements of Ω.
(D1) For $\beta \in (\Omega \cup Y)$, $V_\beta(\pi_\Delta(\xi)) = V_\beta(\xi)$.
(D2) For $\delta \in \Delta$, $V_\delta(\pi_\Delta(\xi')) = V_{\langle\delta,u\rangle}(\xi')$.

(P3) Let $P_0 = \{p^{(0)} \mid p \in P\}$.
(a) For $\xi \in T_{\langle\Sigma,V(s)\rangle}$, if $\#_{\langle\Sigma,\{u\}\rangle}(\xi) = 0$, then $\#_{\langle\Delta,\{u\}\rangle}(M_q^s(\xi)) = 0$.
(b) For $\xi \in T_{\langle\Sigma,V(s)\rangle\cup P_0}$, if $\#_{\langle\Sigma,\{u\}\rangle}(\xi) = 0$, then $\#_{\langle\Delta,\{u\}\rangle}(\hat{M}_q^s(\xi)) = 0$.

Let us prove property (P3) by induction on the structure of ξ. Let $\xi = \langle\sigma, v\rangle(\xi_1, \ldots, \xi_k)$ with $\langle\sigma, v\rangle \in \langle\Sigma, V(s)\rangle^{(k)}$ and $k \geq 0$ such that $\#_{\langle\Sigma,\{u\}\rangle}(\xi) = 0$. By Lemma 3.5, $M_q^s(\xi) = \zeta[\![\langle q', x_i\rangle \leftarrow M_{q'}^s(\xi_i) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!]$ with $\zeta = \mathrm{rhs}_{M^s}(q, \langle\sigma, v\rangle, \langle h^s(\xi_1), \ldots, h^s(\xi_k)\rangle)$, and thus, by Lemma 2.6, $\#_{\langle\Delta,\{u\}\rangle}(M_q^s(\xi)) = S_1^{\langle\Delta,\{u\}\rangle} + S_2^{\langle\Delta,\{u\}\rangle}$, where $S_1^{\langle\Delta,\{u\}\rangle}$ and $S_2^{\langle\Delta,\{u\}\rangle}$ are the sums defined in that lemma (summed over all $\sigma \in \langle\Delta, \{u\}\rangle$). Now $S_1^{\langle\Delta,\{u\}\rangle} = 0$ because $V_{\langle\Delta,\{u\}\rangle}(\zeta) = \varnothing$ by the definition of the rules of M^s and by the fact that $v \neq u$ (because $\#_{\langle\Sigma,\{u\}\rangle}(\xi) = 0$). By induction, $\#_{\langle\Delta,\{u\}\rangle}(M_{q'}^s(\xi_i)) = 0$ and therefore also $S_2^{\langle\Delta,\{u\}\rangle} = 0$, which concludes the proof for the (a) case. For the (b) case the same proof holds, except that we have to consider the additional case $\xi = p \in P_0$: the right-hand side ζ of the p-rule of $\hat{M}^s$ is in $T_{\langle\!\langle Q,\{p\}\rangle\!\rangle}(Y)$ and thus $\#_{\langle\Delta,\{u\}\rangle}(\zeta) = 0$.

First, we want to present a lemma that computes, in the style of Lemma 2.6, the number $\mathrm{Contrib}_M(s, u)$ of output nodes contributed by u.

LEMMA 4.16. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R, $s \in T_\Sigma$, and $u \in V(s)$. Then*

$$\mathrm{Contrib}_M(s, u) = \sum_{\substack{v \,\in\, V_{\langle\!\langle q,p\rangle\!\rangle}(t) \\ q \,\in\, Q}} \;\sum_{w \in V_\Delta(\zeta_q)} \prod F^{[\![\_]\!]}_{\zeta_q,w} \prod F^{[\![\ldots]\!]}_{t,v}$$

with $p = h(s/u)$, $t = \hat{M}_{q_0}(s[u \leftarrow p])$, $\zeta_q = \mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)$ for all $q \in Q$, where $\sigma = s[u] \in \Sigma^{(k)}$, $k \geq 0$, and $p_i = h(s/ui)$ for all $i \in [k]$, $[\![\_]\!] = [\![\langle q', x_i\rangle \leftarrow M_{q'}(s/ui) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!]$, and $[\![\ldots]\!] = [\![\langle\!\langle q, p\rangle\!\rangle \leftarrow M_q(s/u) \mid q \in Q]\!]$.

Proof. By definition, $\mathrm{Contrib}_M(s, u) = \#_{\langle\Delta,\{u\}\rangle}(M_{q_0}^s(\mathrm{dec}(s)))$. Since, by the definition of dec, $\mathrm{dec}(s)[u] = \langle\sigma, u\rangle \in \langle\Sigma, V(s)\rangle^{(k)}$, we get by Lemmas 4.2 and 3.5 and property (P1), $M_{q_0}^s(\mathrm{dec}(s)) = t'[\![\mathrm{rhs}]\!][\![\_]\!]'$, where $t' = \hat{M}_{q_0}^s(\mathrm{dec}(s)[u \leftarrow p])$, $[\![\mathrm{rhs}]\!] = [\![\langle\!\langle q, p\rangle\!\rangle \leftarrow \zeta_q' \mid q \in Q]\!]$ with $\zeta_q' = \mathrm{rhs}_{M^s}(q, \langle\sigma, u\rangle, \langle p_1, \ldots, p_k\rangle)$ for $q \in Q$, and $[\![\_]\!]' = [\![\langle q, x_i\rangle \leftarrow M_q^s(\mathrm{dec}(s)/ui) \mid \langle q, x_i\rangle \in \langle Q, X_k\rangle]\!]$. The application of Lemma 2.6 to $\#_{\langle\Delta,\{u\}\rangle}(t''[\![\_]\!]')$ with $t'' = t'[\![\mathrm{rhs}]\!]$ gives $S_1 + S_2$, where $S_2 = 0$ because $\#_{\langle\Delta,\{u\}\rangle}(M_q^s(\mathrm{dec}(s)/ui)) = 0$ by property (P3)(a) and the fact that $\mathrm{dec}(s)/ui$ contains no symbol in $\langle\Sigma, \{u\}\rangle$ (by the definition of dec). Thus, $\mathrm{Contrib}_M(s, u) = S_1$, which equals

$$(*) \qquad \sum_{v \in V_{\langle\Delta,\{u\}\rangle}(t'[\![\mathrm{rhs}]\!])} \prod F^{[\![\_]\!]'}_{t'[\![\mathrm{rhs}]\!],v}.$$

By the claim below, for $\Phi = [\![\mathrm{rhs}]\!]$ and $\Psi = [\![\_]\!]'$, the sum in $(*)$ equals

$$\sum_{\gamma \in \langle\Delta,\{u\}\rangle} (S_1^\gamma + S_2^\gamma).$$

Now S_1^γ equals zero, because $V_{\langle\Delta,\{u\}\rangle}(t') = \varnothing$, which holds by property (P3)(b) and the fact that $\mathrm{dec}(s)[u \leftarrow p]$ contains no symbol in $\langle\Sigma,\{u\}\rangle$. Thus, the sum in $(*)$ equals $\sum_{\gamma\in\langle\Delta,\{u\}\rangle} S_2^\gamma =$

$$\sum_{\substack{v\,\in\, V_{\langle\!\langle q,p\rangle\!\rangle}(t')\\ q\,\in\, Q}} \;\sum_{w\in V_{\langle\Delta,\{u\}\rangle}(\zeta_q')} \prod F_{\zeta_q',w}^{[\![\_]\!]'} \prod F_{t',v}^{[\![\ldots]\!]'},$$

where $[\![\ldots]\!]'$ is the substitution $[\![\langle\!\langle q,p\rangle\!\rangle \leftarrow M_q^s(\mathrm{dec}(s)/u) \mid q \in Q]\!]$. Let us now show that this sum equals the one of the lemma. For every $q \in Q$ it follows from property (D1) (for $\Omega = \langle\!\langle Q,\{p\}\rangle\!\rangle$ and $\beta = \langle\!\langle q,p\rangle\!\rangle$) that $V_{\langle\!\langle q,p\rangle\!\rangle}(t') = V_{\langle\!\langle q,p\rangle\!\rangle}(\pi_\Delta(t'))$, which equals $V_{\langle\!\langle q,p\rangle\!\rangle}(t)$ by (the $\hat{M}$-version of) property (P2), where π_Δ is the projection defined in that property. Since $\zeta_q' \in T_{\langle Q,X_k\rangle\cup\langle\Delta,\{u\}\rangle}(Y)$ it follows from property (D2) that $V_{\langle\Delta,\{u\}\rangle}(\zeta_q') = V_\Delta(\pi_\Delta(\zeta_q'))$, which equals $V_\Delta(\zeta_q)$ because $\pi_\Delta(\zeta_q') = \zeta_q$ by the definition of the rules of M^s. Now for $w \in V(\zeta_q') = V(\zeta_q)$, $\prod F_{\zeta_q',w}^{[\![\_]\!]'} = \prod F_{\zeta_q,w}^{[\![\_]\!]}$ because for $q' \in Q$, by (D1), $V_{\langle q',x_i\rangle}(\zeta_q') = V_{\langle q',x_i\rangle}(\pi_\Delta(\zeta_q'))$, which equals $V_{\langle q',x_i\rangle}(\zeta_q)$, and for $y \in Y$, by (D1), $\#_y(M_q^s(\mathrm{dec}(s)/ui)) = \#_y(\pi_\Delta(M_q^s(\mathrm{dec}(s)/ui)))$, which equals $\#_y(M_q(s/ui))$ by (P2). Similarly, $\prod F_{t',v}^{[\![\ldots]\!]'} = \prod F_{t,v}^{[\![\ldots]\!]}$ for $v \in V(t') = V(t)$ because, as shown above, $V_{\langle\!\langle q,p\rangle\!\rangle}(t') = V_{\langle\!\langle q,p\rangle\!\rangle}(t)$ for $q \in Q$, and for $y \in Y$, by (D1), $\#_y(M_q^s(\mathrm{dec}(s)/u)) = \#_y(\pi_\Delta(M_q^s(\mathrm{dec}(s)/u)))$, which equals $\#_y(M_q(s))$ by (P2).

It remains to show the following claim, which is a generalization of Lemma 2.6 to two second-order tree substitutions Φ and Ψ. (More precisely, taking the substitution Ψ as the identity on $\Gamma - Y$ gives Lemma 2.6 for the case $\sigma = \gamma \notin \{\sigma_1,\ldots,\sigma_n\} \cup Y$.) Note that $\Phi\Psi$ denotes the composition of Ψ after Φ, i.e., $t(\Phi\Psi) = (t\Phi)\Psi$.

Claim. Let Γ be a ranked alphabet, $\Phi = [\![\sigma_i \leftarrow s_i \mid i \in [n]]\!]$ and $\Psi = [\![\tau_j \leftarrow \xi_j \mid j \in [m]]\!]$ second-order tree substitutions over Γ, and $t \in T_\Gamma$. For every $\gamma \in \Gamma - (\{\sigma_1,\ldots,\sigma_n,\tau_1,\ldots,\tau_m\} \cup Y)$,

$$(\times)\qquad \sum_{v\in V_\gamma(t\Phi)} \prod F_{t\Phi,v}^{\Psi} = S_1^\gamma + S_2^\gamma,$$

where

$$S_1^\gamma = \sum_{v\in V_\gamma(t)} \prod F_{t,v}^{\Phi\Psi} \qquad\text{and}\qquad S_2^\gamma = \sum_{\substack{v\,\in\, V_{\sigma_i}(t)\\ i\,\in\,[n]}} \;\sum_{w\in V_\gamma(s_i)} \prod F_{s_i,w}^{\Psi} \prod F_{t,v}^{\Phi\Psi}.$$

Proof of the claim. Note that the statement does not depend on the numbers $\#_\gamma(\xi_j)$. This is true because the substitution Ψ appears only in the Fs. In fact, for any node v of a tree ζ, $\prod F_{\zeta,v}^{\Psi} = \prod F_{\zeta,v}^{\Psi'}$ for every substitution $\Psi' = [\![\tau_j \leftarrow \xi_j' \mid j \in [m]]\!]$ with the property that $\#_y(\xi_j') = \#_y(\xi_j)$ for every $y \in Y$ and $j \in [m]$; we denote this property by $E(\Psi,\Psi')$. For S_1^γ and S_2^γ a similar statement holds. (Note that if $E(\Psi,\Psi')$, then $E(\Phi\Psi,\Phi\Psi')$; this is true because, by associativity of second-order substitution, $\Phi\Psi = [\![\sigma_i \leftarrow s_i\Psi, \tau_j \leftarrow \xi_j \mid G]\!]$ and $\Phi\Psi' = [\![\sigma_i \leftarrow s_i\Psi', \tau_j \leftarrow \xi_j' \mid G]\!]$, where G denotes the statement "$i \in [n], j \in [m]$ with $\tau_j \notin \{\sigma_1,\ldots,\sigma_n\}$"; by the above, $E(\Psi,\Psi')$ implies that $\prod F_{s_i,v}^{\Psi} = \prod F_{s_i,v}^{\Psi'}$ for any node v of s_i, and thus for every $y \in Y$, $\sum_{v\in V_y(s_i)} \prod F_{s_i,v}^{\Psi} = \sum_{v\in V_y(s_i)} \prod F_{s_i,v}^{\Psi'}$, which means, by Lemma 2.6, that $\#_y(s_i\Psi) = \#_y(s_i\Psi')$.)

The idea of the proof is as follows. We will apply Lemma 2.6 twice: first to $\#_\gamma(t'\Psi')$, where $t' = t\Phi$ and Ψ' is a substitution with $E(\Psi,\Psi')$, and second to $\#_\gamma(tB)$

with $B = \Phi\Psi'$. The first application will give the left-hand side of the equation $(\times)$, and the second one will give the right-hand side of that equation. Clearly, by definition of the composition of second-order tree substitutions, $\#_\gamma(t'\Psi') = \#_\gamma(tB)$.

Define $\Psi' = [\![\tau_j \leftarrow \xi'_j \mid j \in [m]]\!]$ with $E(\Psi, \Psi')$ and $\#_\gamma(\xi'_j) = 0$ for all $j \in [m]$. Then for $t' = t\Phi$, $\#_\gamma(t'\Psi')$ equals, by Lemma 2.6, $R_1^\gamma + R_2^\gamma$ with $R_1^\gamma = \sum_{v \in V_\gamma(t\Phi)} \prod F_{t\Phi,v}^{\Psi'}$ and $R_2^\gamma = 0$ because all the numbers $\#_\gamma(\xi'_j)$ are zero by the definition of Ψ'. Since $E(\Psi, \Psi')$, this means that $\#_\gamma(t'\Psi') = \sum_{v \in V_\gamma(t\Phi)} \prod F_{t\Phi,v}^{\Psi}$, which is the left-hand side of equation $(\times)$.

By the associativity of second-order tree substitution, $B = \Phi\Psi'$ equals

$$[\![\sigma_i \leftarrow s_i\Psi', \tau_j \leftarrow \xi'_j \mid i \in [n], j \in [m] \text{ with } \tau_j \notin \Sigma_n]\!],$$

where $\Sigma_n = \{\sigma_1, \dots, \sigma_n\}$. The application of Lemma 2.6 to $\#_\gamma(tB)$ gives $R_1^\gamma + R_2^\gamma$ with $R_1^\gamma = \sum_{v \in V_\gamma(t)} \prod F_{t,v}^{\Phi\Psi'}$ and

$$R_2^\gamma = \sum_{v \in V_{\sigma_i}(t), i \in [n]} \#_\gamma(s_i\Psi') \cdot \prod F_{t,v}^{\Phi\Psi'} + \sum_{v \in V_{\tau_j}(t), j \in [m], \tau_j \notin \Sigma_n} \#_\gamma(\xi'_j) \cdot \prod F_{t,v}^{\Phi\Psi'}.$$

Since $\#_\gamma(\xi'_j) = 0$, the second term of R_2^γ equals zero. In the first term of R_2^γ we apply Lemma 2.6 to $\#_\gamma(s_i\Psi')$, which gives $T_1^\gamma + T_2^\gamma$, where $T_2^\gamma = 0$ because $\#_\gamma(\xi'_j) = 0$, and $T_1^\gamma = \sum_{v \in V_{\sigma_i}(t), i \in [n]} \sum_{w \in V_\gamma(s_i)} \prod F_{s_i,w}^{\Psi'} \prod F_{t,v}^{\Phi\Psi'}$. Since $E(\Psi, \Psi')$, $R_1^\gamma = S_1^\gamma$ and $T_1^\gamma = S_2^\gamma$, which concludes the proof of the claim. ∎

Using Lemma 4.16 we can now prove that if an $\mathrm{MTT}^{\mathrm{R}}$ is finite copying, then it is of finite contribution, which is defined next.

DEFINITION 4.17 (finite contribution). *Let M be an MTT^R with input alphabet Σ. Then M is of* finite contribution *if there is a $c \in \mathbb{N}$ such that* $\mathrm{Contrib}_M(s, u) \leq c$ *for every $s \in T_\Sigma$ and $u \in V(s)$.*

Consider now a finite copying $\mathrm{MTT}^{\mathrm{R}}$ M. In the translations of M, every node of the input tree is translated at most $I \cdot N^{I-1}$ times (cf. the discussion on page 71 of [19]), where I and N are input and parameter copying bounds for M, respectively. This implies that the number $\mathrm{Contrib}_M(s, u)$ of output nodes contributed by the node u is bounded.

LEMMA 4.18. *Let M be an MTT^R. If M is finite copying, then it is of finite contribution.*

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$, $s \in T_\Sigma$, and $u \in V(s)$. Let I be an input copying bound for M and let N be a parameter copying bound for M. Furthermore, let m be the maximal size of the right-hand side of a rule of M. By the definition of fci it follows that for $t = \hat{M}_{q_0}(s[u \leftarrow p])$ and $p = h(s/u)$, $\#_{\langle\!\langle Q,\{p\}\rangle\!\rangle}(t) \leq I$. By the definition of fcp it follows that, for every $v \in V_{\langle\!\langle q,p\rangle\!\rangle}(t)$ and $q \in Q$, $\prod F_{t,v}^{[\![\dots]\!]} \leq N^{I-1}$, where $[\![\dots]\!] = [\![\langle\!\langle q,p\rangle\!\rangle \leftarrow M_q(s/u) \mid q \in Q]\!]$. By Lemma 4.16 this means that $\mathrm{Contrib}_M(s, u) \leq I \cdot N^{I-1} \cdot \max\{\sum_{w \in V_\Delta(\zeta_q)} \prod F_{\zeta_q,w}^{[\![\_]\!]} \mid q \in Q\}$, where $[\![\_]\!] = [\![\langle q', x_i\rangle \leftarrow M_{q'}(s/ui) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!]$. By the definition of m this is $\leq I \cdot N^{I-1} \cdot m \cdot \max\{\prod F_{\zeta_q,w}^{[\![\_]\!]} \mid q \in Q, w \in V_\Delta(\zeta_q)\} \leq I \cdot N^{I-1} \cdot m \cdot N^{m-1}$. Hence $\mathrm{Contrib}_M(s, u) \leq c$ for $c = I \cdot N^{I-1} \cdot m \cdot N^{m-1}$. ∎

As discussed in the beginning of this subsection, if an $\mathrm{MTT}^{\mathrm{R}}$ is of finite contribution, then it is lsi. This holds because, by (P2), $\mathrm{size}(M_{q_0}(s)) = \mathrm{size}(M_{q_0}^s(\mathrm{dec}(s)))$ $= \sum_{u \in V(s)} \mathrm{Contrib}_M(s, u) \leq c \cdot \mathrm{size}(s)$. Together with Lemma 4.18 this gives us the desired result: finite copying implies lsi.

THEOREM 4.19. *If an MTT^R is finite copying, then it is of linear size increase.*

5. Proper normal form. In section 4.3 we showed that if an $\mathrm{MTT}^{\mathrm{R}}$ is finite copying, then it is lsi. In sections 6 and 7 we want to prove that the converse also holds, i.e., that lsi implies finite copying. However, in general this does *not* hold: there are lsi $\mathrm{MTT}^{\mathrm{R}}$s that are not finite copying. Roughly speaking, the reason for this is that the part of the output tree of which unboundedly many copies are generated, by means of input variables or parameters, might be a fixed tree that does not change for different inputs. So, it can happen that an input tree s_n of size n generates a state sequence of length n, but the number of different output trees that are eventually generated by the states in the state sequence is finite. Then the $\mathrm{MTT}^{\mathrm{R}}$ is not finite copying in the input, but the translation it realizes can still be lsi (cf. the $\mathrm{MTT}^{\mathrm{R}}$ M at the beginning of section 5.1). Similarly, a tree $M_q(s_n)$ might contain n copies of a parameter y_j, but there are only finitely many different output trees that will be substituted for y_j in the actual output $M_{q_0}(s)$. Then M is not finite copying in the parameters, but the translation it realizes can be lsi (cf. the $\mathrm{MTT}^{\mathrm{R}}$ at the beginning of section 5.2).

Intuitively it should be clear that a state that generates, for any input, only a finite number of different output trees t is not needed; it can be eliminated by immediately substituting the correct tree t, which can be determined by regular look-ahead. This gives rise to a normal form, called *input proper*, which is treated in section 5.1. Similarly for a parameter y_j of a state q, if the number of actual output trees t that will be substituted for y_j is finite, then this parameter is not needed; it can be eliminated by immediately substituting the correct t, which can be computed in the states of the $\mathrm{MTT}^{\mathrm{R}}$. This gives rise to a normal form, called *parameter proper*; it is treated in section 5.2.

Altogether, an $\mathrm{MTT}^{\mathrm{R}}$ will be called *proper* if it is input proper, parameter proper, and productive. Again, this is a normal form; i.e., for every $\mathrm{MTT}^{\mathrm{R}}$ there is an equivalent one which is proper. Then in section 6 it can be proved that if a proper $\mathrm{MTT}^{\mathrm{R}}$ is lsi, then it is finite copying.

5.1. Input proper. Consider the following $\mathrm{MTT}^{\mathrm{R}}$ M, which is lsi but *not* fci. Let $M = (Q, \Sigma, \Delta, q_0, R)$ with $Q = \{q_0^{(0)}, q^{(0)}, q'^{(0)}\}$, $\Sigma = \{\gamma^{(1)}, a^{(0)}, b^{(0)}\}$, $\Delta = \{\sigma^{(2)}, a^{(0)}, b^{(0)}\}$, and R consisting of the following rules:

$$\begin{array}{lcl}
\langle q_0, \gamma(x_1)\rangle & \to & \sigma(\langle q, x_1\rangle, \langle q', x_1\rangle),\\
\langle q, \gamma(x_1)\rangle & \to & \langle q, x_1\rangle,\\
\langle q', \gamma(x_1)\rangle & \to & \sigma(\langle q, x_1\rangle, \langle q', x_1\rangle),\\
\langle r, \alpha\rangle & \to & \alpha \quad \text{(for every } r \in Q \text{ and } \alpha \in \{a, b\}).
\end{array}$$

Note that M is in fact a top-down tree transducer. Intuitively, M translates every monadic tree $s_n = \gamma(\dots\gamma(\alpha)\dots) = \gamma^n(\alpha)$ of height n (with $\alpha \in \{a, b\}$) into a comb $t_n = \sigma(\alpha, \sigma(\alpha, \dots \sigma(\alpha, \alpha)\dots))$ of height n. Thus, $\mathrm{size}(\tau_M(s)) \le 2 \cdot \mathrm{size}(s)$ for every $s \in T_\Sigma$ and so M is lsi. Clearly, M is not fci because $\mathrm{sts}_M(s_n, u) = q^n q'$ for $n \ge 1$ and $u = 1^n$ the unique leaf of s_n. The reason for this is that M generates n copies of q, but q generates only a finite number of different trees (viz. the trees a and b). How can we change M into an equivalent $\mathrm{MTT}^{\mathrm{R}}$ which is fci? The idea is to simply delete the state q and to determine by regular look-ahead the appropriate tree in $\{a, b\}$. In this example we just need $L_p = \{\gamma^n(a) \mid n \ge 0\}$ and $L_{p'} = \{\gamma^n(b) \mid n \ge 0\}$, and then the q_0-rule of M is replaced by two q_0-rules with right-hand sides $\sigma(a, \langle q', x_1\rangle)$ and $\sigma(b, \langle q', x_1\rangle)$ for look-ahead p and p', respectively, and similarly for the q'-rule.

We will say that an $\mathrm{MTT}^{\mathrm{R}}$ M is "input proper" if every state, except possibly the initial one, produces infinitely many output trees (in $T_\Delta(Y)$). More precisely, for every look-ahead state p of M and every state $q \neq q_0$, M should produce infinitely many output trees taking L_p (the trees for which the look-ahead automaton arrives in state p) as input; in fact, this is only required if $\langle\langle q,p\rangle\rangle$ is reachable, i.e., if $\langle\langle q,p\rangle\rangle$ occurs in $\hat{M}_{q_0}(s[u \leftarrow p])$ for some s and u (see Definition 4.12). Note that q_0 is the only state that *may* generate only finitely many output trees; this is to make sure that an MTT$^{\mathrm{R}}$ that generates only finitely many output trees can be transformed into one that is input proper.

The notion of input properness was defined in [1] for generalized syntax-directed translation schemes (which are a variant of top-down tree transducers) and was there called "reduced." We add two useful technical properties to it.

Definition 5.1 (input proper). *An MTT^R $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ is* input proper *(for short, i-proper) if*

(i) *for every $q \in Q$ and $p \in P$ such that $q \neq q_0$ and $\langle\langle q,p\rangle\rangle$ is reachable, the set* $\mathrm{Out}(q,p) = \{M_q(s) \mid s \in L_p\}$ *is infinite;*

(ii) *q_0 does not occur in the right-hand sides of the rules in R; and*

(iii) *$L_p \neq \varnothing$ for every $p \in P$.*

Note that $\mathrm{Out}(q,p) \subseteq T_\Delta(Y_m)$ for $q \in Q^{(m)}$. Before it is proved (in Lemma 5.4) that i-properness is a normal form for MTT$^{\mathrm{R}}$s, we need the following two straightforward lemmas about finiteness of $\mathrm{Out}(q,p)$.

Lemma 5.2. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R. For given $q \in Q^{(m)}$ and $p \in P$ it is decidable whether or not Out(q,p) is finite. Moreover,* $\mathrm{Out}(q,p)$ *can be constructed if it is finite.*

Proof. Let $\overline{M}$ be the MTT$^{\mathrm{R}}$ constructed in the proof of Lemma 3.7. Then, for every $s \in T_\Sigma$, $\tau_{\overline{M}}(\overline{q}(s)) = M_q(s)[y_j \leftarrow \overline{y}_j \mid j \in [m]]$ and hence $M_q(s) = \tau_{\overline{M}}(\overline{q}(s))\Pi$, where $\Pi = [\overline{y}_j \leftarrow y_j \mid j \in [m]]$. The substitution Π can be realized by a (very simple) top-down tree transducer. Thus, for the regular tree language $L = \{\overline{q}(s) \mid s \in L_p\}$, $\mathrm{Out}(q,p) = \{M_q(s) \mid s \in L_p\} = \{\tau_{\overline{M}}(s)\Pi \mid s \in L\} = \tau_N(\tau_{\overline{M}}(L))$. By Lemma 3.8 the finiteness of $\tau_N(\tau_{\overline{M}}(L))$ is decidable, and in case of finiteness $\tau_N(\tau_{\overline{M}}(L))$ can be constructed. ◻

Lemma 5.3. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be a nondeleting MTT^R. Let $q \in Q$, $\sigma \in \Sigma^{(k)}$, $k \geq 1$, and $p, p_1, \ldots, p_k \in P$ such that $p = h_\sigma(p_1, \ldots, p_k)$ and $L_{p_j} \neq \varnothing$ for every $j \in [k]$. If $\langle r, x_i\rangle \in \langle Q, X_k\rangle$ occurs in* $\mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)$ *and* $\mathrm{Out}(q,p)$ *is finite, then* $\mathrm{Out}(r, p_i)$ *is finite.*

Proof. For $j \in [k] - \{i\}$ fix trees $s_j \in T_\Sigma$ with $h(s_j) = p_j$. Let $\xi = \zeta[\![\ldots]\!]$ with $\zeta = \mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)$ and $[\![\ldots]\!] = [\![\langle q', x_j\rangle \leftarrow M_{q'}(s_j) \mid q' \in Q, j \in [k] - \{i\}]\!]$. By the definition of $\mathrm{Out}(q,p)$, Lemma 3.5, and associativity of second-order tree substitution, $O = \{M_q(\sigma(s_1, \ldots, s_k)) \mid s_i \in L_{p_i}\} = \{\xi[\![s_i]\!] \mid s_i \in L_{p_i}\}$, where $[\![s_i]\!]$ denotes the substitution $[\![\langle q', x_i\rangle \leftarrow M_{q'}(s_i) \mid q' \in Q]\!]$ is a subset of $\mathrm{Out}(q,p)$ and hence finite. Since M is nondeleting, both $[\![\ldots]\!]$ and $[\![s_i]\!]$ are nondeleting by Lemma 3.11(1). Hence, by Lemma 2.1, ξ has a subtree $\langle r, x_i\rangle(\xi_1, \ldots, \xi_m)$, where $m = \mathrm{rank}_Q(r)$. Again by Lemma 2.1, $\xi[\![s_i]\!]$ has a subtree $\langle r, x_i\rangle(\xi_1, \ldots, \xi_m)[\![s_i]\!] = M_r(s_i)[y_j \leftarrow \xi_j[\![s_i]\!] \mid j \in [m]]$. Thus, for every $t \in \mathrm{Out}(r, p_i)$ (i.e., $t = M_r(s_i)$ for some $s_i \in L_{p_i}$) the tree $t[y_j \leftarrow \xi_j[\![s_i]\!] \mid j \in [m]]$ is a subtree of $\xi[\![s_i]\!]$, i.e., it is a subtree of a tree in the finite set O. This implies finiteness of $\mathrm{Out}(r, p_i)$. ◻

We are now ready to prove that i-properness is a normal form. The construction involved is similar to the one of Lemma 5.5 of [1] except that we apply it repeatedly to obtain an i-proper MTT$^{\mathrm{R}}$ as opposed to their single application, which is insufficient (also in their formalism, which means that their proof of the lemma is incorrect).

Lemma 5.4. *For every MTT^R M there is (effectively) an i-proper and productive*

MTT^R M' equivalent to M. If M is a T^R, then so is M'.

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an $\mathrm{MTT}^{\mathrm{R}}$. By Lemma 3.10 we may assume that M is productive. Moreover, we may assume that q_0 does not occur in the right-hand side of any rule of M (if it does, replace it in all rules by a new state q_0' which has the same rules as q_0).

Before we construct the $\mathrm{MTT}^{\mathrm{R}}$ M' which is i-proper and realizes the same translation as M, let us define an auxiliary notion. For each $p \in P$, let F_p denote the set $\{q \in Q \mid \mathrm{Out}(q,p) \text{ is finite}\}$ of states which produce finitely many output trees in $T_\Delta(Y)$ on input trees in L_p. Note that F_p can be constructed effectively, because, by Lemma 5.2, it is decidable whether or not $\mathrm{Out}(q,p)$ is finite. Moreover, $\mathrm{Out}(q,p)$ can be constructed for every $q \in F_p$.

The $\mathrm{MTT}^{\mathrm{R}}$ M' is constructed in such a way that if $\langle r, x_i\rangle$ occurs in $\mathrm{rhs}_{M'}(q, \sigma, \langle p_1, \ldots, p_k\rangle)$, then $r \notin F_{p_i}$. This implies point (i) of i-properness of M' as follows: If $\langle\langle r, p\rangle\rangle \in \langle\langle Q, P\rangle\rangle$ is reachable (with $r \neq q_0$), then there are $s \in T_\Sigma$ and $u \in V(s)$ such that $\langle\langle r, p\rangle\rangle$ occurs in $\hat{M}'_{q_0}(s[u \leftarrow p])$. Since $r \neq q_0$, $u = vi$ for some $i \geq 1$ and $v \in \mathbb{N}^*$. By Lemma 4.3 this implies that $\langle r, x_i\rangle$ occurs in the right-hand side of a rule of M' with $p_i = p$. This means that $r \notin F_p$, i.e., $\mathrm{Out}(r,p)$ is infinite.

Note that an example for the construction in this proof can be found in Example 5.5.

We first construct the $\mathrm{MTT}^{\mathrm{R}}$ $\pi(M)$ by simply deleting occurrences of $\langle r, x_i\rangle$ with $r \in F_{p_i}$ and replacing them by the correct tree in $\mathrm{Out}(r, p_i)$, which is determined by regular look-ahead. Due to the change of look-ahead automaton, an occurrence of $\langle r, x_i\rangle$ in the $(q, \sigma, \langle p_1, \ldots, p_k\rangle)$-rule of M with $r \notin F_{p_i}$ might produce only finitely many trees for the new look-ahead states (p_i, φ_i). For this reason we have to iterate the application of π until the sets F_p do not change anymore. This results in the desired $\mathrm{MTT}^{\mathrm{R}}$ M'.

For each $p \in P$ let Φ_p be the (finite) set of all mappings $\varphi : F_p \to T_\Delta(Y)$ such that there is an $s \in L_p$ with $\varphi(q) = M_q(s)$ for every $q \in F_p$. Note that Φ_p is finite because $\varphi(q) \in \mathrm{Out}(q,p)$, which is finite for $q \in F_p$. This also implies that Φ_p can be obtained effectively by checking, for the (finitely many) mappings $\varphi : F_p \to \bigcup_{q \in F_p} \mathrm{Out}(q,p)$, whether or not φ is in Φ_p. This is decidable because $\varphi \in \Phi_p$ if and only if $K_{p,\varphi} = L_p \cap \bigcap_{q \in F_p} M_q^{-1}(\{\varphi(q)\})$ is nonempty; $K_{p,\varphi}$ is regular by Lemma 3.7 (and the closure of the regular tree languages under intersection) and hence has a decidable emptiness problem (cf., e.g., Theorem II.10.2 of [31]). The mappings in Φ_p partition L_p into the sets $K_{p,\varphi}$, which can be determined by regular look-ahead.

We now construct the $\mathrm{MTT}^{\mathrm{R}}$ $\pi(M) = (Q, P', \Sigma, \Delta, q_0, R', h')$ as follows. Let $P' = \{(p, \varphi) \mid p \in P, \varphi \in \Phi_p\}$. For $\sigma \in \Sigma^{(k)}$ and $(p_1, \varphi_1), \ldots, (p_k, \varphi_k) \in P'$ let, for every $q \in Q^{(m)}$, the rule

$$\langle q, \sigma(x_1, \ldots, x_k)\rangle(y_1, \ldots, y_m) \to \zeta_q \Theta \quad \langle (p_1, \varphi_1), \ldots, (p_k, \varphi_k)\rangle$$

be in R', where $\zeta_q = \mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)$ and $\Theta = [\![\langle r, x_i\rangle \leftarrow \varphi_i(r) \mid r \in F_{p_i}, i \in [k]]\!]$, and let $h'_\sigma((p_1, \varphi_1), \ldots, (p_k, \varphi_k)) = (p, \varphi)$, where $p = h_\sigma(p_1, \ldots, p_k)$ and $\varphi = \{(q, \zeta_q\Theta) \mid q \in F_p, \zeta_q = \mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)\}$.

Before we prove that the look-ahead automaton of $\pi(M)$ is as desired, let us show that it is well defined, i.e., that $\varphi \in \Phi_p$. We must show that there is an $s \in L_p$ such that, for every $q \in F_p$, $\varphi(q) = M_q(s)$. Since $\varphi_i \in \Phi_{p_i}$ for $i \in [k]$, there are $s_i \in L_{p_i}$ such that $\varphi_i(r) = M_r(s_i)$ for all $i \in [k]$ and $r \in F_{p_i}$. Hence, for $q \in F_p$, $\varphi(q) = \zeta_q\Theta$ with $\zeta_q = \mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)$ and $\Theta = [\![\langle r, x_i\rangle \leftarrow M_r(s_i) \mid \langle r, x_i\rangle \in \langle F_{p_i}, X_k\rangle]\!]$. By

Lemma 5.3 and the definition of F_p, only $\langle r, x_i\rangle$ with $r \in F_{p_i}$ occur in ζ_q. Therefore we can extend Θ to all elements of $\langle Q, X_k\rangle$. By Lemma 3.5 we get $\varphi(q) = M_q(s)$ for $s = \sigma(s_1, \ldots, s_k)$. Since $p = h_\sigma(p_1, \ldots, p_k)$, $s \in L_p$.

Claim 1. Let $s \in T_\Sigma$. If $h'(s) = (p, \varphi)$, then $p = h(s)$ and $\varphi(q) = M_q(s)$ for every $q \in F_p$.

The proof is by induction on the structure of s. Let $s = \sigma(s_1, \ldots, s_k)$ with $s_1, \ldots, s_k \in T_\Sigma$ and $h'(s_i) = (p_i, \varphi_i) \in P'$ for $i \in [k]$. By definition, p equals $h_\sigma(p_1, \ldots, p_k) = h(s)$. For $q \in F_p$, $\varphi(q) = \mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)\Theta$. By induction, $\varphi_i(r) = M_r(s_i)$ for all $i \in [k]$ and $r \in F_{p_i}$. For the same reason as above we can extend Θ to all elements of $\langle Q, X_k\rangle$ to get $M_q(s)$.

This claim implies that $\pi(M)$ satisfies point (iii) of i-properness. In fact, if $(p, \varphi) \in P'$, then $\varphi \in \Phi_p$, and so there exists $s \in L_p$ such that $\varphi(q) = M_q(s)$ for every $q \in F_p$. Thus, by Claim 1, $h'(s) = (p, \varphi)$. Hence, $L_{(p,\varphi)} \neq \varnothing$.

The MTT$^{\mathrm{R}}$ $\pi(M)$ realizes the same translation as M. This follows from Claim 2 for $q = q_0$.

Claim 2. For $q \in Q$ and $s \in T_\Sigma$, $\pi(M)_q(s) = M_q(s)$.

Again we prove this by induction on s. Let $s = \sigma(s_1, \ldots, s_k)$ with $s_1, \ldots, s_k \in T_\Sigma$ and $h'(s_i) = (p_i, \varphi_i) \in P'$ for $i \in [k]$. By the definition of the rules of $\pi(M)$ and by Lemma 3.5, $\pi(M)_q(s)$ equals $\mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)\Theta[\![\_]\!]$, where $[\![\_]\!] = [\![\langle q', x_i\rangle \leftarrow \pi(M)_{q'}(s_i) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!]$. By Claim 1, Θ equals $[\![\langle r, x_i\rangle \leftarrow M_r(s_i) \mid r \in F_{p_i}, i \in [k]]\!]$, and by induction $[\![\_]\!] = [\![\langle q', x_i\rangle \leftarrow M_{q'}(s_i) \mid \langle q', x_i\rangle \in \langle Q, X_k\rangle]\!]$. Thus $\Theta[\![\_]\!] = [\![\_]\!]$ and we get $\mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)[\![\_]\!]$, which, by Lemma 3.5, equals $M_q(s)$.

The MTT$^{\mathrm{R}}$ $\pi(M)$ is productive because M is productive and the application of Θ does not delete nodes. Formally, consider a right-hand side $\zeta_q\Theta$ of $\pi(M)$ with $\zeta_q = \mathrm{rhs}_M(q, \sigma, \langle p_1, \ldots, p_k\rangle)$, $q \in Q^{(m)}$, and $m \geq 0$. For every $r \in F_{p_i}$, $\varphi_i(r) = M_r(s)$ for some $s \in T_\Sigma$. Thus, by Lemma 3.11(1), $\#_{y_\nu}(\varphi_i(r)) \geq 1$ for every $\nu \in [\mathrm{rank}_Q(r)]$; i.e., the substitution Θ is nondeleting. Since, for $j \in [m]$, $\#_{y_j}(\zeta_q) \geq 1$ this implies, by Lemma 2.1, that $\#_{y_j}(\zeta_q\Theta) \geq 1$; i.e., $\pi(M)$ is nondeleting. Analogously, by Lemma 3.11(2), $\#_{y_\nu}(\varphi_i(r)) \notin Y$ for $r \in F_{p_i}$ and $\nu \in [\mathrm{rank}_Q(r)]$; i.e., the substitution Θ is nonerasing. Since, for $j \in [m]$, $\zeta_q \notin Y$ this implies, by Lemma 2.2, that $\zeta_q\Theta \notin Y$; i.e., $\pi(M)$ is nonerasing.

Since $\pi(M)$ has the same states as M, $\pi(M)$ is a T$^{\mathrm{R}}$ if M is.

We now discuss the reason for iterating π. Consider an occurrence of $\langle r, x_i\rangle$ in the right-hand side of a rule of $\pi(M)$. We know that $r \notin F_{p_i}$, because each such occurrence is removed by the substitution Θ in the definition of the rules of $\pi(M)$. Thus, $\mathrm{Out}(r, p_i)$ is infinite. However, through the new look-ahead, the set L_{p_i} is partitioned into sets $L_{(p_i,\varphi_i)}$, $\varphi_i \in \Phi_{p_i}$ (to see this, consider an $s \in L_{p_i}$; then, by Claim 1, $s \in L_{(p_i,\varphi_i)}$, where φ_i is defined as $\varphi_i(q) = M_q(s)$ for every $q \in F_{p_i}$). Thus, we merely know, by Claim 2, that the union of $\mathrm{Out}(r, (p_i, \varphi_i))$ for all $\varphi_i \in \Phi_{p_i}$ is infinite, but for a particular $\varphi_i \in \Phi_{p_i}$, $\mathrm{Out}(r, (p_i, \varphi_i))$ might be finite, which means that $\pi(M)$ is not i-proper (see Example 5.5).

Let us now show that the iterative application of π yields an i-proper MTT$^{\mathrm{R}}$. In particular, we iterate the application of π until

$$(*) \qquad F_{(p,\varphi)} = F_p \quad \text{for every } (p, \varphi) \in P'.$$

It follows from $(*)$ that if $\langle r, x_i\rangle$ occurs in the right-hand side of a rule of $\pi(M)$, then by the definition of Θ, $r \notin F_{p_i}$, and hence by $(*)$, $r \notin F_{(p_i,\varphi_i)}$. Thus $(*)$ implies (point (i) of) i-properness of $\pi(M)$, as argued in the beginning of this proof.

It remains to show that after a finite number of applications of π, $(*)$ holds. Clearly, $F_p \subseteq F_{(p,\varphi)} \subseteq Q$, because $\mathrm{Out}(q, (p, \varphi)) \subseteq \mathrm{Out}(q, p)$, as argued above. Let us first show that, for every $(p,\varphi) \in P'$, $F_{(p,\varphi)} = F_p$ implies that (after constructing $\pi(\pi(M))$) $F_{((p,\varphi),\varphi')} = F_{(p,\varphi)}$ for every $\varphi' \in \Phi_{(p,\varphi)}$. Let $\varphi' \in \Phi_{(p,\varphi)}$; i.e., there is an $s \in L_{(p,\varphi)}$ such that $\varphi'(q) = \pi(M)_q(s)$ for every $q \in F_{(p,\varphi)} = F_p$. Since, by Claims 1 and 2, $\pi(M)_q(s) = M_q(s) = \varphi(q)$ for every $q \in F_p$, it follows that $\varphi' = \varphi$. This means that $L_{((p,\varphi),\varphi')} = \{s \in L_{(p,\varphi)} \mid \pi(M)_q(s) = \varphi'(q) \text{ for all } q \in F_{(p,\varphi)}\}$ equals $\{s \in L_{(p,\varphi)} \mid M_q(s) = \varphi(q) \text{ for all } q \in F_p\} = L_{(p,\varphi)}$. This implies that $\mathrm{Out}(q,((p,\varphi),\varphi')) = \mathrm{Out}(q,(p,\varphi))$ and thus $F_{((p,\varphi),\varphi')} = \{q \in Q \mid \mathrm{Out}(q,((p,\varphi),\varphi'))$ is finite$\} = \{q \in Q \mid \mathrm{Out}(q,(p,\varphi))$ is finite$\} = F_{(p,\varphi)}$.

Now, after at most $k = |Q|$ iterations of π, $(*)$ holds. Let $(\cdots((p,\varphi_1),\varphi_2)\ldots,\varphi_k)$ be denoted by $(p,\varphi_1,\ldots,\varphi_k)$. Then, for every look-ahead state $(p,\varphi_1,\ldots,\varphi_k)$ of $\pi^k(M)$, $F_{(p,\varphi_1,\ldots,\varphi_{k-1})} = F_{(p,\varphi_1,\ldots,\varphi_k)}$. This is true because $F_p = \varnothing$ implies $F_{(p,\varphi_1)} = \varnothing$ (since $\Phi_p = \{\varphi_1\}$), and $F_{(p,\varphi_1,\ldots,\varphi_i)} = F_{(p,\varphi_1,\ldots,\varphi_{i+1})}$ implies that $F_{(p,\varphi_1,\ldots,\varphi_j)} = F_{(p,\varphi_1,\ldots,\varphi_i)}$ for all $j \geq i$ (by the above). Since a sequence of nonempty subsets of Q in which each set is a proper subset of the next one has length at most $|Q| = k$, $F_{(p,\varphi_1,\ldots,\varphi_{k-1})} = F_{(p,\varphi_1,\ldots,\varphi_k)}$. Thus, $M' = \pi^k(M)$ is i-proper. ∎

The next example illustrates the construction of an i-proper $\mathrm{MTT}^{\mathrm{R}}$ following the proof of Lemma 5.4.

Example 5.5. For simplicity let us consider an $\mathrm{MTT}^{\mathrm{R}}$ without parameters, i.e., a T^{R}. Let $M = (Q,P,\Sigma,\Delta,q_0,R,h)$ be a T^{R} with $Q = \{q_0,q,q',i\}$, $P = \{p\}$, $\Sigma = \{\alpha^{(0)},\gamma^{(1)},\sigma^{(1)}\}$, $\Delta = \{\alpha^{(0)},\beta^{(0)},\gamma^{(1)},\sigma^{(1)},\delta^{(2)}\}$, and let R consist of the following rules:

$$
\begin{array}{lcll}
\langle q_0,\gamma(x_1)\rangle & \to & \delta(\langle q,x_1\rangle,\langle i,x_1\rangle) & \langle p\rangle,\\
\langle q_0,\sigma(x_1)\rangle & \to & \langle q',x_1\rangle & \langle p\rangle,\\
\langle q,\gamma(x_1)\rangle & \to & \alpha & \langle p\rangle,\\
\langle q,\sigma(x_1)\rangle & \to & \beta & \langle p\rangle,\\
\langle q',\gamma(x_1)\rangle & \to & \alpha & \langle p\rangle,\\
\langle q',\sigma(x_1)\rangle & \to & \sigma(\langle i,x_1\rangle) & \langle p\rangle,\\
\langle i,\gamma(x_1)\rangle & \to & \gamma(\langle i,x_1\rangle) & \langle p\rangle,\\
\langle i,\sigma(x_1)\rangle & \to & \sigma(\langle i,x_1\rangle) & \langle p\rangle,\\
\langle r,\alpha\rangle & \to & \alpha & \text{for each } r \in Q.
\end{array}
$$

Let us now define $M_1 = \pi(M) = (Q,P',\Sigma,\Delta,q_0,R',h')$. We obtain $F_p = \{q\}$ and $\Phi_p = \{\varphi_\alpha,\varphi_\beta\}$ with $\varphi_\alpha = \{(q,\alpha)\}$ and $\varphi_\beta = \{(q,\beta)\}$, and thus $P' = \{(p,\varphi_\alpha),(p,\varphi_\beta)\}$. As can easily be verified, the rules of the look-ahead automaton of M_1 are the following: $h'_\alpha = (p,\varphi_\alpha)$, $h'_\gamma((p,\varphi_\alpha)) = h'_\gamma((p,\varphi_\beta)) = (p,\varphi_\alpha)$, $h'_\sigma((p,\varphi_\alpha)) = h'_\sigma((p,\varphi_\beta)) = (p,\varphi_\beta)$.

The q-, q'-, and i-rules in R' are identical to the ones in R for both new look-ahead states. The q_0-rules in R' are the following:

$$
\begin{array}{lcll}
\langle q_0,\gamma(x_1)\rangle & \to & \delta(\alpha,\langle i,x_1\rangle) & \langle (p,\varphi_\alpha)\rangle,\\
\langle q_0,\gamma(x_1)\rangle & \to & \delta(\beta,\langle i,x_1\rangle) & \langle (p,\varphi_\beta)\rangle,\\
\langle q_0,\sigma(x_1)\rangle & \to & \langle q',x_1\rangle & \langle (p,\varphi_\alpha)\rangle,\\
\langle q_0,\sigma(x_1)\rangle & \to & \langle q',x_1\rangle & \langle (p,\varphi_\beta)\rangle.
\end{array}
$$

Note that $L_{(p,\varphi_\alpha)} = \{\alpha\} \cup \{\gamma(s) \mid s \in T_\Sigma\}$ and $L_{(p,\varphi_\beta)} = \{\sigma(s) \mid s \in T_\Sigma\}$. Hence $\mathrm{Out}(q',(p,\varphi_\alpha)) = \{\alpha\}$, and so the T^{R} M_1 is not i-proper yet, because $F_{(p,\varphi_\alpha)} = \{q,q'\} \neq F_p$. Thus we have to apply π again. Let $M' = \pi(M_1) = (Q,P'',\Sigma,\Delta,q_0,R'',h'')$. We get $\Phi_{(p,\varphi_\alpha)} = \{\varphi\}$, with $\varphi = \{(q,\alpha),(q',\alpha)\}$ and $\Phi_{(p,\varphi_\beta)} = \{\varphi_\beta\}$. Thus $P'' = \{((p,\varphi_\alpha),\varphi),((p,\varphi_\beta),\varphi_\beta)\}$. The look-ahead automaton of M' stays the same as for M_1 except for a renaming of states: (p,φ_α) becomes $((p,\varphi_\alpha),\varphi)$ and (p,φ_β)

becomes $((p,\varphi_\beta),\varphi_\beta)$. The q-, q'-, and i-rules in R'' are identical to the ones in R' (and R) for all look-ahead states. The q_0-rules in R'' are the following:

$$\begin{array}{lcll}
\langle q_0,\gamma(x_1)\rangle & \to & \delta(\alpha,\langle i,x_1\rangle) & \langle((p,\varphi_\alpha),\varphi)\rangle,\\
\langle q_0,\gamma(x_1)\rangle & \to & \delta(\beta,\langle i,x_1\rangle) & \langle((p,\varphi_\beta),\varphi_\beta)\rangle,\\
\langle q_0,\sigma(x_1)\rangle & \to & \alpha & \langle((p,\varphi_\alpha),\varphi)\rangle,\\
\langle q_0,\sigma(x_1)\rangle & \to & \langle q',x_1\rangle & \langle((p,\varphi_\beta),\varphi_\beta)\rangle.
\end{array}$$

The $\mathrm{T^R}$ M' is i-proper because $F_{((p,\varphi_\alpha),\varphi)}=\{q,q'\}=F_{(p,\varphi_\alpha)}$ and $F_{((p,\varphi_\beta),\varphi_\beta)}=\{q\}=F_{(p,\varphi_\beta)}$. We finally note that it is easy to transform M into a generalized syntax-directed translation scheme that forms a counterexample to the proof of Lemma 5.5 of [1].

5.2. Parameter proper. Consider the following MTT M which is lsi but *not* fcp. Let $M=(Q,\Sigma,\Delta,q_0,R)$ with $Q=\{q_0^{(0)},q^{(1)}\}$, $\Sigma=\{\sigma^{(2)},\gamma^{(2)},\alpha^{(0)},\beta^{(0)}\}$, and $\Delta=\{\sigma^{(2)},\gamma^{(2)},\alpha^{(1)},\beta^{(1)},\bar\sigma^{(0)},\bar\gamma^{(0)}\}$. For all $\delta\in\{\sigma,\gamma\}$ and $a\in\{\alpha,\beta\}$, let the following rules be in R:

$$\begin{array}{lcl}
\langle q_0,\delta(x_1,x_2)\rangle & \to & \delta(\langle q,x_1\rangle(\bar\delta),\langle q,x_2\rangle(\bar\delta)),\\
\langle q,\delta(x_1,x_2)\rangle(y_1) & \to & \delta(\langle q,x_1\rangle(y_1),\langle q,x_2\rangle(y_1)),\\
\langle q_0,a\rangle & \to & a(\bar a),\\
\langle q,a\rangle(y_1) & \to & a(y_1).
\end{array}$$

Intuitively, M moves the root symbol of the input tree to each of its leaves; e.g., for $s=\sigma(\gamma(\alpha,\beta),\alpha)$ we get $\tau_M(s)=\sigma(\gamma(\alpha(\bar\sigma),\beta(\bar\sigma),\alpha(\bar\sigma)))$. Thus, M is lsi (because $\mathrm{size}(\tau_M(s))\le 2\cdot\mathrm{size}(s)$). Clearly, M is not fcp, because $\#_{y_1}(M_q(s))$ equals the number of leaves of s. This time, the reason is that M generates a lot of parameter occurrences which have only finitely many "argument trees" (viz., $\bar\sigma$ and $\bar\gamma$). A jth argument tree for q and p is a tree ξ_j such that $\langle\!\langle q,p\rangle\!\rangle(\xi_1,\dots,\xi_m)$ is a subtree of some $\hat M_{q_0}(s[u\leftarrow p])$.

The idea of the next normal form is to eliminate parameters y_j of q for which there are only finitely many jth argument trees (for look-ahead p). This can be done by keeping the information on these argument trees in the states of the new $\mathrm{MTT^R}$ and by appropriately replacing y_j by the correct argument tree in each right-hand side. For the example MTT M above, we have to add states $q_{\bar\delta}$, $\delta\in\{\sigma,\gamma\}$ of rank zero, and take as rules

$$\begin{array}{lcll}
\langle q_0,\delta(x_1,x_2)\rangle & \to & \delta(\langle q_{\bar\delta},x_1\rangle,\langle q_{\bar\delta},x_2\rangle), & \\
\langle q_{\bar\delta},\rho(x_1,x_2)\rangle & \to & \rho(\langle q_{\bar\delta},x_1\rangle,\langle q_{\bar\delta},x_2\rangle) & \text{for } \rho\in\{\sigma,\gamma\},\\
\langle q_0,a\rangle & \to & a(\bar a) & \text{for } a\in\{\alpha,\beta\},\\
\langle q_{\bar\delta},a\rangle & \to & a(\bar\delta) & \text{for } a\in\{\alpha,\beta\}.
\end{array}$$

This shows that the translation τ_M can actually be realized by a top-down tree transducer (which is in general, of course, not the case).

DEFINITION 5.6 (parameter proper, proper). *An MTT^R $M=(Q,P,\Sigma,\Delta,q_0,R,h)$ is* parameter proper *(for short, p-proper) if for every $q\in Q^{(m)}$, $m\ge 1$, $j\in[m]$, and $p\in P$*

(i) *if $\langle\!\langle q,p\rangle\!\rangle$ is reachable, then the set* $\mathrm{Arg}(q,j,p)$ *defined as*

$$\{t/vj\mid \exists s\in T_\Sigma,u\in V(s): t=\hat M_{q_0}(s[u\leftarrow p]),v\in V(t),t[v]=\langle\!\langle q,p\rangle\!\rangle\}$$

is infinite; and

(ii) *if* $\langle\!\langle q,p\rangle\!\rangle$ *is not reachable, then* $\#_{y_j}(M_q(s)) \le 1$ *for all* $s \in L_p$.

The MTT^R *M is* proper *if it is productive and both i-proper and p-proper.*

Note that $\mathrm{Arg}(q,j,p) \subseteq T_{\langle\!\langle Q,\{p\}\rangle\!\rangle\cup\Delta}$. Note also that $\langle\!\langle q,p\rangle\!\rangle$ is reachable if and only if $\mathrm{Arg}(q,j,p) \neq \varnothing$.

Point (ii) in Definition 5.6 says that if a parameter appears more than once in $M_q(s)$, then $\langle\!\langle q,h(s)\rangle\!\rangle$ is reachable. This (mild) additional requirement is needed to force an lsi MTT$^{\mathrm{R}}$ to be fcp, because Definition 4.6 of the fcp property requires $\#_{y_j}(M_q(s)) \le N$ for *all* states q; i.e., $\langle\!\langle q,h(s)\rangle\!\rangle$ might *not* be reachable.

Similar to the case of i-properness, we present two lemmas concerning the finiteness of $\mathrm{Arg}(q,j,p)$. First, let us show that it is decidable whether $\mathrm{Arg}(q,j,p)$ is infinite.

LEMMA 5.7. *Let* $M = (Q,P,\Sigma,\Delta,q_0,R,h)$ *be an MTT*^{R}. *For given* $q \in Q^{(m)}$, $m \ge 1$, $j \in [m]$, *and* $p \in P$, *it is decidable whether or not* $\mathrm{Arg}(q,j,p)$ *is finite. Moreover,* $\mathrm{Arg}(q,j,p)$ *can be constructed if it is finite.*

Proof. Let K_p be the regular tree language $\{s \in T_{\hat\Sigma} \mid p$ occurs exactly once in $s\}$ with $\hat\Sigma = \Sigma \cup \{p^{(0)}\}$. Then $\tau_{\hat M}(K_p) \subseteq T_{\langle\!\langle Q,\{p\}\rangle\!\rangle\cup\Delta}$. We now construct a partial nondeterministic top-down tree transducer N which takes a tree in $T_{\langle\!\langle Q,\{p\}\rangle\!\rangle\cup\Delta}$ as input and generates as output the jth subtree of an occurrence of $\langle\!\langle q,p\rangle\!\rangle$. (A partial nondeterministic top-down tree transducer is defined as in Definitions 3.1 and 3.2, but for q and σ there may be no or several rules of the form $\langle q,\sigma(x_1,\dots,x_k)\rangle \to \zeta$.) Let $N = (\{r^{(0)},\mathrm{id}^{(0)}\},\Gamma,\Gamma,r,R')$, where $\Gamma = \langle\!\langle Q,\{p\}\rangle\!\rangle \cup \Delta$ and R' consists of the following rules:

$$\begin{array}{lcll}
\langle r,\gamma(x_1,\dots,x_k)\rangle & \to & \langle r,x_i\rangle & \text{for all } \gamma \in \Gamma^{(k)}, k \ge 1, i \in [k],\\
\langle r,\langle\!\langle q,p\rangle\!\rangle(x_1,\dots,x_m)\rangle & \to & \langle \mathrm{id},x_j\rangle, & \\
\langle \mathrm{id},\gamma(x_1,\dots,x_k)\rangle & \to & \gamma(\langle \mathrm{id},x_1\rangle,\dots,\langle \mathrm{id},x_k\rangle) & \text{for all } \gamma \in \Gamma^{(k)}, k \ge 0.
\end{array}$$

Clearly, $\tau_N(\tau_{\hat M}(K_p)) = \mathrm{Arg}(q,j,p)$, because every tree t in $\tau_{\hat M}(K_p)$ equals $\hat M_{q_0}(s[u \leftarrow p])$ for some s and u, and for every subtree $\langle\!\langle q,p\rangle\!\rangle(\xi_1,\dots,\xi_m)$ of t, $(t,\xi_j) \in \tau_N$. The finiteness of $L = \tau_N(\tau_{\hat M}(K_p))$ can be decided by Lemma 3.8, and in case of finiteness L can be constructed. ∎

LEMMA 5.8. *Let* $M = (Q,P,\Sigma,\Delta,q_0,R,h)$ *be an i-proper and productive MTT*^{R}. *Let* $q \in Q^{(n)}$, $\sigma \in \Sigma^{(k)}$, $n,k \ge 0$, *and* $p,p_1,\dots,p_k \in P$ *such that* $p = h_\sigma(p_1,\dots,p_k)$ *and* $\langle\!\langle q,p\rangle\!\rangle$ *is reachable. Let* $\langle r,x_i\rangle(t_1,\dots,t_m)$ *be a subtree of* $\mathrm{rhs}_M(q,\sigma,\langle p_1,\dots,p_k\rangle)$ *with* $r \in Q^{(m)}$, $m \ge 0$, $i \in [k]$, *and* $t_1,\dots,t_m \in T_{\langle Q,X_k\rangle\cup\Delta}(Y_n)$.

For $j \in [m]$, *the set* $\mathrm{Arg}(r,j,p_i)$ *is infinite if in* t_j *there is*

(i) *an occurrence of* $y_\mu \in Y_n$, *where* $\mathrm{Arg}(q,\mu,p)$ *is infinite, or*

(ii) *an occurrence of an element of* $\langle Q, X_k - \{x_i\}\rangle$, *or*

(iii) *an occurrence of* $y_\mu \in Y_n$ *such that there is a* $\xi \in \mathrm{Arg}(q,\mu,p)$ *for which* $\xi[\![\mathrm{rhs}]\!]$ *contains an occurrence of an element of* $\langle Q, X_k - \{x_i\}\rangle$, *where* $[\![\mathrm{rhs}]\!]$ *denotes the substitution* $[\![\langle\!\langle q',p\rangle\!\rangle \leftarrow \mathrm{rhs}_M(q',\sigma,\langle p_1,\dots,p_k\rangle) \mid q' \in Q]\!]$.

Proof. Informally, and roughly speaking, the idea of this lemma is the following. Consider the given rule

$$\langle q,\sigma(x_1,\dots,x_k)\rangle(y_1,\dots,y_n) \to \dots \langle r,x_i\rangle(t_1,\dots,t_j,\dots,t_m) \dots \qquad \langle p_1,\dots,p_k\rangle.$$

In case (i), if $\xi_\mu \in \mathrm{Arg}(q,\mu,p)$, i.e., ξ_μ is a μth argument tree of q and p, then, due to the application of the above rule, the tree $t_j[y_\mu \leftarrow \xi_\mu[\![\mathrm{rhs}]\!]]$ determines a jth argument tree of r and p_i, i.e., an element of $\mathrm{Arg}(r,j,p_i)$. By "determines" we mean that the remaining parameters and the input variables $\neq x_i$ have to be substituted by appropriate trees, and then all $\langle q', s_\nu\rangle$ (where s_ν is substituted for $x_\nu \neq x_i$) have to be substituted by $M_{q'}(s_\nu)$, and $\langle q', x_i\rangle$ by $\langle\!\langle q', p_i\rangle\!\rangle$. Now, since it is given that there are infinitely many such ξ_μ, there are also infinitely many $t_j[y_\mu \leftarrow \xi_\mu[\![\mathrm{rhs}]\!]]$, and these determine infinitely many elements of $\mathrm{Arg}(r, j, p_i)$ because all involved substitutions are productive (see Lemma 2.7). In case (ii), if t_j contains an occurrence of $\langle q', x_\nu\rangle$ with $\nu \neq i$, and $t \in \mathrm{Out}(q', p_\nu)$, then the tree $t_j[\![\langle q', x_\nu\rangle \leftarrow t]\!]$ determines an element of $\mathrm{Arg}(r, j, p_i)$. Since M is i-proper, $\mathrm{Out}(q', p_\nu)$ is infinite and hence so is $\mathrm{Arg}(r, j, p_i)$. Finally, in case (iii), if $\xi[\![\mathrm{rhs}]\!]$ contains an occurrence of $\langle q', x_\nu\rangle$ with $\nu \neq i$, and $t \in \mathrm{Out}(q', p_\nu)$, then $t_j[y_\mu \leftarrow \xi[\![\mathrm{rhs}]\!][\![\langle q', x_\nu\rangle \leftarrow t]\!]]$ determines an element of $\mathrm{Arg}(r, j, p_i)$. We now turn to the formal proof.

Consider $s \in T_\Sigma$, $u \in V(s)$, and $\xi_1, \dots, \xi_n \in T_{\langle\!\langle Q,\{p\}\rangle\!\rangle\cup\Delta}$ such that $\langle\!\langle q, p\rangle\!\rangle(\xi_1, \dots, \xi_n)$ is a subtree of $\hat{M}_{q_0}(s[u \leftarrow p])$. Consider also $s_\nu \in L_{p_\nu}$ for $\nu \in [k]$. Note that such trees exist because $\langle\!\langle q, p\rangle\!\rangle$ is reachable and because M satisfies point (iii) of i-properness.

Let $s' = s[u \leftarrow \sigma(s_1, \dots, s_k)]$. Note that $s'/u = \sigma(s_1, \dots, s_k)$ is in L_p and that $s'[u \leftarrow p] = s[u \leftarrow p]$. By Lemma 4.3, $\hat{M}_{q_0}(s'[ui \leftarrow p_i]) = \hat{M}_{q_0}(s[u \leftarrow p])[\![\mathrm{rhs}]\!]\Psi_{s_1,\dots,s_k}[\![i]\!]$, with $[\![\mathrm{rhs}]\!]$ as in (iii), $\Psi_{s_1,\dots,s_k} = [\![\langle q', x_\nu\rangle \leftarrow M_{q'}(s_\nu) \mid q' \in Q, \nu \in [k] - \{i\}]\!]$, and $[\![i]\!] = [\![\langle q', x_i\rangle \leftarrow \langle\!\langle q', p_i\rangle\!\rangle \mid q' \in Q]\!]$.

Since M is nondeleting, so is $[\![\mathrm{rhs}]\!]$ and, by Lemma 3.11(1), so is $\Psi_{s_1,\dots,s_k}$. Then, by Lemma 2.1, the tree $\hat{M}_{q_0}(s'[ui \leftarrow p_i])$ has a subtree $\langle\!\langle q, p\rangle\!\rangle(\xi_1, \dots, \xi_n)[\![\mathrm{rhs}]\!]\Psi_{s_1,\dots,s_k}[\![i]\!] = \zeta\Pi_{\xi_1,\dots,\xi_n}\Psi_{s_1,\dots,s_k}[\![i]\!]$ with $\zeta = \mathrm{rhs}_M(q, \sigma, \langle p_1, \dots, p_k\rangle)$ and $\Pi_{\xi_1,\dots,\xi_n} = [y_\eta \leftarrow \xi_\eta[\![\mathrm{rhs}]\!] \mid \eta \in [n]]$. Again by Lemma 2.1 it has a subtree $\langle\!\langle r, p_i\rangle\!\rangle(t'_1, \dots, t'_m)$, where, for $j \in [m]$,

$$(*) \qquad t'_j = t_j\Pi_{\xi_1,\dots,\xi_n}\Psi_{s_1,\dots,s_k}[\![i]\!] \in \mathrm{Arg}(r, j, p_i).$$

(i) Let $j \in [m]$ such that y_μ is a subtree of t_j. By Lemma 2.1, $y_\mu\Pi_{\xi_1,\dots,\xi_n}\Psi_{s_1,\dots,s_k}[\![i]\!] = \xi_\mu[\![\mathrm{rhs}]\!]\Psi_{s_1,\dots,s_k}[\![i]\!]$ is a subtree of t'_j. Thus $\mathrm{size}(t'_j) \geq \mathrm{size}(\xi_\mu[\![\mathrm{rhs}]\!]\Psi_{s_1,\dots,s_k}[\![i]\!])$, which is $\geq \mathrm{size}(\xi_\mu)$ by Lemma 2.7 and the fact that $[\![\mathrm{rhs}]\!]$ and $\Psi_{s_1,\dots,s_k}$ are productive by Lemma 3.11. We now let $\xi_1, \dots, \xi_n$ vary in $(*)$: For every ξ_μ in the infinite set $\mathrm{Arg}(q, \mu, p)$ there are $s \in T_\Sigma$, $u \in V(s)$, and ξ_η, $\eta \in [n] - \{\mu\}$ such that $\langle\!\langle q, p\rangle\!\rangle(\xi_1, \dots, \xi_n)$ is a subtree of $\hat{M}_{q_0}(s[u \leftarrow p])$; then the size of $t_j\Pi_{\xi_1,\dots,\xi_n}\Psi_{s_1,\dots,s_k}[\![i]\!] \in \mathrm{Arg}(r, j, p_i)$ is $\geq \mathrm{size}(\xi_\mu)$. Thus, $\mathrm{Arg}(r, j, p_i)$ is infinite.

(ii) Let $j \in [m]$, $q' \in Q^{(l)}$, $l \geq 0$, and $\nu \in [k] - \{i\}$ such that t_j has a subtree $\langle q', x_\nu\rangle(\bar{t}_1, \dots, \bar{t}_l)$ for some trees $\bar{t}_1, \dots, \bar{t}_l$. Then $\langle\!\langle q', p_\nu\rangle\!\rangle$ is reachable by the same argument as given above equation $(*)$ (where we showed that $\langle\!\langle r, p_i\rangle\!\rangle$ is reachable). By Lemma 2.1, t'_j has the subtree $M_{q'}(s_\nu)[y_\eta \leftarrow \bar{t}_\eta\Pi_{\xi_1,\dots,\xi_n}\Psi_{s_1,\dots,s_k}[\![i]\!] \mid \eta \in [l]]$, the size of which is $\geq \mathrm{size}(M_{q'}(s_\nu))$. Since M satisfies points (i) and (ii) of i-properness, the set $\mathrm{Out}(q', p_\nu) = \{M_{q'}(s_\nu) \mid s_\nu \in L_{p_\nu}\}$ is infinite. We now let s_ν vary in $(*)$: For every $s_\nu \in L_{p_\nu}$ the size of $t_j\Pi_{\xi_1,\dots,\xi_n}\Psi_{s_1,\dots,s_k}[\![i]\!] \in \mathrm{Arg}(r, j, p_i)$ is $\geq \mathrm{size}(M_{q'}(s_\nu))$. Thus, $\mathrm{Arg}(r, j, p_i)$ is infinite.

(iii) Let $s \in T_\Sigma$ and $u \in V(s)$ such that $\hat{M}_{q_0}(s[u \leftarrow p])$ has a subtree $\langle\!\langle q, p\rangle\!\rangle(\xi_1, \dots, \xi_n)$ for trees $\xi_1, \dots, \xi_n$ and $\xi_\mu[\![\mathrm{rhs}]\!]$ has a subtree $\langle q', x_\nu\rangle(\bar{t}_1, \dots, \bar{t}_l)$ for some $q' \in Q^{(l)}$, $l \geq 0$, $\nu \in [k] - \{i\}$, and trees $\bar{t}_1, \dots, \bar{t}_l$. It follows from Lemma 2.6 ($S_1^{\langle q', x_\nu\rangle} = 0$) that ξ_μ contains some $\langle\!\langle q'', p\rangle\!\rangle$, $q'' \in Q$, such that $\mathrm{rhs}_M(q'', \sigma, \langle p_1, \dots, p_k\rangle)$ contains $\langle q', x_\nu\rangle$. Since $\langle\!\langle q'', p\rangle\!\rangle$ is reachable (because ξ_μ is a subtree of $\hat{M}_{q_0}(s[u \leftarrow p])$), $\langle\!\langle q', p_\nu\rangle\!\rangle$ is reachable by the same argument as used above $(*)$. Thus, $\mathrm{Out}(q', p_\nu)$ is infinite. Let $j \in [m]$ such that y_μ occurs in t_j. Then, by Lemma 2.1, t'_j has a subtree $M_{q'}(s_\nu)[y_\eta \leftarrow \bar{t}_\eta\Psi_{s_1,\dots,s_k}[\![i]\!] \mid \eta \in [l]]$, the size of which is $\geq \mathrm{size}(M_{q'}(s_\nu))$. Letting s_ν range over L_{p_ν} in $(*)$ this implies, analogous to case (ii), that $\mathrm{Arg}(r, j, p_i)$ is infinite. ∎

We are now ready to prove that properness (i.e., i-properness, p-properness, and productivity) is a normal form for $\mathrm{MTT}^{\mathrm{R}}$s.

THEOREM 5.9. *For every MTT^R M there is (effectively) a proper MTT^R* $\mathrm{prop}(M)$ *equivalent to M. If M is a T^R, then so is* $\mathrm{prop}(M)$.

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$. By Lemma 5.4 we may assume that M is productive and i-proper. Let $q \in Q^{(n)}$ and $p \in P$. The idea of constructing $\mathrm{prop}(M)$ is to delete all parameters y_j of q for which $\mathrm{Arg}(q, j, p)$ is finite and to keep the parameters $y_{j_1}, \ldots, y_{j_m}$ of q for which $\mathrm{Arg}(q, j_\nu, p)$ is infinite. The information on the actual parameter tree which has to be substituted for y_j is stored in the states of $\mathrm{prop}(M)$. More precisely, a state of $\mathrm{prop}(M)$ will be of the form (q, φ), where φ is a mapping which associates with j_ν the new parameter y_ν, and with j a tree ξ_j in the finite set $\mathrm{Arg}(q, j, p)$.

Let us first define an auxiliary notion. For every $q \in Q^{(n)}$, $n \geq 0$, and $p \in P$, let $\Phi_{q,p}$ be the (finite) set of all mappings φ from $[n]$ to $T_{\langle\langle Q,\{p\}\rangle\rangle \cup \Delta} \cup Y$ such that there are $s \in T_\Sigma$, $u \in V(s)$, and $\xi_1, \ldots, \xi_n \in T_{\langle\langle Q,\{p\}\rangle\rangle \cup \Delta}$: $\hat{M}_{q_0}(s[u \leftarrow p])$ has a subtree $\langle\langle q, p\rangle\rangle(\xi_1, \ldots, \xi_n)$ and $F_{q,p}(\varphi, \xi_1, \ldots, \xi_n)$. The predicate $F_{q,p}(\varphi, \xi_1, \ldots, \xi_n)$ holds if for all $j \in [n]$ the following holds: if $j = j_\eta$ for an $\eta \in [m]$, then $\varphi(j) = y_\eta$, and otherwise $\varphi(j) = \xi_j$, where $\{j_1, \ldots, j_m\} = \{j \in [n] \mid \mathrm{Arg}(q, j, p) \text{ is infinite}\}$ and $j_1 < \cdots < j_m$.

By the definition of Arg, $\varphi(j) \notin Y$ implies $\varphi(j) \in \mathrm{Arg}(q, j, p)$. Note that $\Phi_{q,p}$ is finite because $\varphi(j) \in Y_m \cup K_j$ with $K_j = \mathrm{Arg}(q, j, p)$ for finite $\mathrm{Arg}(q, j, p)$, and $K_j = \varnothing$ otherwise. Therefore, $\Phi_{q,p}$ can be obtained effectively by checking, for the (finitely many) mappings $\varphi : [n] \to K$, whether or not $\varphi \in \Phi_{q,p}$ (where $K = Y_m \cup \bigcup_{j \in [n]} K_j$ can be constructed by Lemma 5.7). This is decidable because, apart from the requirement that $\varphi(j_\eta) = y_\eta$ for all $\eta \in [m]$ (which is decidable by Lemma 5.7), φ is in $\Phi_{q,p}$ if and only if $\tau_{\hat{M}}^{-1}(L) \cap S$ is nonempty, where $S = \{s[u \leftarrow p] \mid s \in T_\Sigma, u \in V(s)\}$ and L consists of all trees in $T_{\langle\langle Q,\{p\}\rangle\rangle \cup \Delta}$ which have a subtree $\langle\langle q, p\rangle\rangle(\xi_1, \ldots, \xi_n)$ with $\xi_j = \varphi(j)$ for all $j \notin \varphi^{-1}(Y)$. Clearly, L is regular and hence, by Lemma 3.7, $\tau_{\hat{M}}^{-1}(L)$ is regular. Since S is regular, so is $\tau_{\hat{M}}^{-1}(L) \cap S$, which implies that its emptiness is decidable.

We first construct the $\mathrm{MTT}^{\mathrm{R}}$ $\pi(M)$ by deleting, in the right-hand side of a rule (with look-ahead $\langle p_1, \ldots, p_k\rangle$), all parameters y_j of $\langle r, x_i\rangle$ for which $\mathrm{Arg}(r, j, p_i)$ is finite and replacing them by the appropriate tree in $\mathrm{Arg}(r, j, p_i)$. This tree is coded in the states of $\pi(M)$. Due to the new states of $\pi(M)$, a parameter y_{j_ν} of r with $\mathrm{Arg}(r, j_\nu, p_i)$ infinite might correspond in $\pi(M)$ to the parameter y_ν of a state (r, φ) with finite $\mathrm{Arg}((r, \varphi), \nu, p_i)$. For this reason we have to iterate the application of π (as in the construction in the proof of Lemma 5.4) until the ranks of the states do not change anymore. This results in the desired $\mathrm{MTT}^{\mathrm{R}}$ $\mathrm{prop}(M)$.

Define $\pi(M) = (Q', P, \Sigma, \Delta, (q_0, \varnothing), R', h)$ with $Q' = \{(q, \varphi)^{(m)} \mid q \in Q, \exists p \in P : \varphi \in \Phi_{q,p}, m = |\varphi^{-1}(Y)|\}$. For every $(q, \varphi) \in Q'^{(m)}$, $\sigma \in \Sigma^{(k)}$, $q \in Q^{(n)}$, $m, n, k \geq 0$, and $p, p_1, \ldots, p_k \in P$ with $p = h_\sigma(p_1, \ldots, p_k)$, let the rule

$$\langle (q, \varphi), \sigma(x_1, \ldots, x_k)\rangle(y_1, \ldots, y_m) \to \zeta \quad \langle p_1, \ldots, p_k\rangle$$

be in R' such that if $\varphi \notin \Phi_{q,p}$, then ζ is an arbitrary ("dummy") tree in $T_\Delta(Y_m) - Y$ with $\#_{y_j}(\zeta) = 1$ for every $j \in [m]$, and if $\varphi \in \Phi_{q,p}$, then $\zeta = \mathrm{repl}(\mathrm{rhs}(\rho)\Pi)$, where ρ is the $(q, \sigma, \langle p_1, \ldots, p_k\rangle)$-rule of M, Π denotes the substitution

$$[y_j \leftarrow \varphi(j)[\![\mathrm{rhs}]\!] \mid j \in [n]] \;\text{ with }\; [\![\mathrm{rhs}]\!] = [\![\langle\langle r, p\rangle\rangle \leftarrow \mathrm{rhs}_M(r, \sigma, \langle p_1, \ldots, p_k\rangle) \mid r \in Q]\!],$$

and for every subtree $t \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$ of $\mathrm{rhs}(\rho)\Pi$ the tree $\mathrm{repl}(t)$ is recursively defined as follows:

- for $t \in Y_m$, $\mathrm{repl}(t) = t$,
- for $t = \delta(t_1, \ldots, t_l)$ with $\delta \in \Delta^{(l)}$, $l \geq 0$, and $t_1, \ldots, t_l \in T_{\langle Q,X_k\rangle \cup \Delta}(Y_m)$, $\mathrm{repl}(t) = \delta(\mathrm{repl}(t_1), \ldots, \mathrm{repl}(t_l))$, and
- for $t = \langle q', x_i\rangle(t_1, \ldots, t_l)$, $\langle q', x_i\rangle \in \langle Q, X_k\rangle^{(l)}$, $l \geq 0$, and $t_1, \ldots, t_l \in T_{\langle Q,X_k\rangle \cup \Delta}(Y_m)$,
$$\mathrm{repl}(t) = \langle (q', \varphi'), x_i\rangle(\mathrm{repl}(t_{j_1}), \ldots, \mathrm{repl}(t_{j_\mu})),$$
where $\{j_1, \ldots, j_\mu\} = \{j \in [l] \mid \mathrm{Arg}(q', j, p_i) \text{ is infinite}\}$, $j_1 < \cdots < j_\mu$, and for $j \in [l]$,
$$\varphi'(j) = \begin{cases} y_\eta & \text{if } j = j_\eta \text{ for an } \eta \in [\mu], \\ t_j[\![i]\!] & \text{otherwise} \end{cases}$$
with $[\![i]\!] = [\![\langle r, x_i\rangle \leftarrow \langle\!\langle r, p_i\rangle\!\rangle \mid r \in Q]\!]$.

This ends the construction of $\pi(M)$.

Well-definedness of $\pi(M)$. To prove that $\pi(M)$ is well defined, we have to show that $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)$ is in $T_{\langle Q',X_k\rangle \cup \Delta}(Y_m)$. Since $\mathrm{rhs}(\rho) \in T_{\langle Q,X_k\rangle \cup \Delta}(Y_n)$ and $\varphi(Y_n) \subseteq Y_m \cup T_{\langle\!\langle Q,\{p\}\rangle\!\rangle \cup \Delta}$ (because $\varphi \in \Phi_{q,p}$), it follows that $\mathrm{rhs}(\rho)\Pi \in T_{\langle Q,X_k\rangle \cup \Delta}(Y_m)$. To prove that $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi) \in T_{\langle Q',X_k\rangle \cup \Delta}(Y_m)$ we must show that, in the definition of repl, if $\langle q', x_i\rangle(t_1, \ldots, t_l)$ is a subtree of $\mathrm{rhs}(\rho)\Pi$, then $(q', \varphi') \in Q'$; i.e., there is a p' such that $\varphi' \in \Phi_{q',p'}$.

We will show that $\varphi' \in \Phi_{q',p_i}$, i.e., that there are $s' \in T_\Sigma$, $u' \in V(s')$, and $\xi'_1, \ldots, \xi'_l \in T_{\langle\!\langle Q,\{p_i\}\rangle\!\rangle \cup \Delta}$ such that $\langle\!\langle q', p_i\rangle\!\rangle(\xi'_1, \ldots, \xi'_l)$ is a subtree of $\hat{M}_{q_0}(s'[u' \leftarrow p_i])$ and $F_{q',p_i}(\varphi', \xi'_1, \ldots, \xi'_l)$. Since $\varphi \in \Phi_{q,p}$, there are $s \in T_\Sigma$, $u \in V(s)$, and $\xi_1, \ldots, \xi_n \in T_{\langle\!\langle Q,\{p\}\rangle\!\rangle \cup \Delta}$ such that $\langle\!\langle q, p\rangle\!\rangle(\xi_1, \ldots, \xi_n)$ is a subtree of $\hat{M}_{q_0}(s[u \leftarrow p])$ and $F_{q,p}(\varphi, \xi_1, \ldots, \xi_n)$. Note in particular that $\langle\!\langle q, p\rangle\!\rangle$ is reachable. Take $s' = s[u \leftarrow \sigma(s_1, \ldots, s_k)]$ with $s_\nu \in L_{p_\nu}$ for all $\nu \in [k]$, and take $u' = ui$. The s_ν exist, because M is i-proper (point (iii)). By Lemma 4.3, $\hat{M}_{q_0}(s'[u' \leftarrow p_i])$ equals $\hat{M}_{q_0}(s[u \leftarrow p])[\![\mathrm{rhs}]\!][\![..]\!][\![i]\!]$, where $[\![..]\!]$ denotes $[\![\langle r, x_\nu\rangle \leftarrow M_r(s_\nu) \mid \langle r, x_\nu\rangle \in \langle Q, X_k - \{x_i\}\rangle]\!]$, and $[\![\mathrm{rhs}]\!]$ and $[\![i]\!]$ are as in the definition of $\pi(M)$. Since $\langle\!\langle q, p\rangle\!\rangle(\xi_1, \ldots, \xi_n)$ is a subtree of $\hat{M}_{q_0}(s[u \leftarrow p])$ it follows, by Lemma 2.1 and the fact that $[\![..]\!]$ is nondeleting by Lemma 3.11(1), that $\hat{M}_{q_0}(s'[u' \leftarrow p_i])$ has a subtree $\mathrm{rhs}(\rho)\Pi'[\![..]\!][\![i]\!]$, where $\Pi' = [y_\mu \leftarrow \xi_\mu[\![\mathrm{rhs}]\!] \mid \mu \in [n]]$.

Consider the following two cases: (i) there are $t'_1, \ldots, t'_l \in T_{\langle Q,X_k\rangle \cup \Delta}(Y_n)$ such that $\langle q', x_i\rangle(t'_1, \ldots, t'_l)$ is a subtree of $\mathrm{rhs}(\rho)$ and $t'_j\Pi = t_j$ for all $j \in [l]$, and (ii) $\langle q', x_i\rangle(t_1, \ldots, t_l)$ is a subtree of $\varphi(\mu)[\![\mathrm{rhs}]\!]$ for some $\mu \in [n]$.

(i) Since $\mathrm{rhs}(\rho)$ has a subtree $\langle q', x_i\rangle(t'_1, \ldots, t'_l)$, it follows, by application of $\Pi'[\![..]\!][\![i]\!]$ (and Lemma 2.1), that $\hat{M}_{q_0}(s'[u' \leftarrow p_i])$ has a subtree $\langle\!\langle q', p_i\rangle\!\rangle(\xi'_1, \ldots, \xi'_l)$ with $\xi'_j = t'_j\Pi'[\![..]\!][\![i]\!]$ for every $j \in [l]$. Let $j \in [l]$ such that $\mathrm{Arg}(q', j, p_i)$ is finite. Then by Lemma 5.8(ii) and (iii), both t'_j and all $\xi_\mu[\![\mathrm{rhs}]\!]$ such that y_μ occurs in t'_j do not contain elements of $\langle Q, X_k - \{x_i\}\rangle$. Thus $\xi'_j = t'_j\Pi'[\![..]\!][\![i]\!]$ equals $t'_j\Pi'[\![i]\!]$. By Lemma 5.8(i), t'_j does not contain any $y_\mu \in Y_n$ such that $\mathrm{Arg}(q, \mu, p)$ is infinite. Thus, since $F_{q,p}(\varphi, \xi_1, \ldots, \xi_n)$, $t'_j\Pi'[\![i]\!] = t'_j\Pi[\![i]\!] = t_j[\![i]\!]$. By the definition of φ' this shows that $F_{q',p_i}(\varphi', \xi'_1, \ldots, \xi'_l)$.

(ii) There is an occurrence of y_μ in $\mathrm{rhs}(\rho)$, because M is nondeleting. Since $\varphi(\mu) = \xi_\mu$, by the fact that $F_{q,p}(\varphi, \xi_1, \ldots, \xi_n)$ holds, this means that in $\mathrm{rhs}(\rho)\Pi'[\![..]\!][\![i]\!]$ there is a subtree $\langle\!\langle q', p_i\rangle\!\rangle(\xi'_1, \ldots, \xi'_l)$ with $\xi'_j = t_j[\![..]\!][\![i]\!]$ for $j \in [l]$. Since $\langle q', x_i\rangle(t_1, \ldots, t_l)$ is a subtree of $\xi_\mu[\![\mathrm{rhs}]\!]$, it follows from the definition of second-order tree substitution that ξ_μ has a subtree $\langle\!\langle q'', p\rangle\!\rangle(\zeta_1, \ldots, \zeta_\lambda)$ and the right-hand side of the $(q'', \sigma, \langle p_1, \ldots, p_k\rangle)$-rule ρ'' has a subtree $\langle q', x_i\rangle(t'_1, \ldots, t'_l)$ such that $t_j = t'_j[y_\nu \leftarrow \zeta_\nu[\![\mathrm{rhs}]\!] \mid \nu \in [\lambda]]$ for every $j \in [l]$. Note that $\langle\!\langle q'', p \rangle\!\rangle$ is reachable because it occurs in ξ_μ. Now let $j \in [l]$ such that $\mathrm{Arg}(q', j, p_i)$ is finite. Then, as in case (i), by Lemma 5.8(ii) and (iii) applied to ρ'', both t'_j and all $\zeta_\nu[\![\mathrm{rhs}]\!]$ such that y_ν occurs in t'_j do not contain elements of $\langle Q, X_k - \{x_i\}\rangle$. Hence t_j does not contain elements of $\langle Q, X_k - \{x_i\}\rangle$ and thus $\xi'_j = t_j[\![..]\!][\![i]\!] = t_j[\![i]\!]$. By the definition of φ' this shows that $F_{q',p_i}(\varphi', \xi'_1, \ldots, \xi'_l)$.

Equivalence of $\pi(M)$ *and* M. We now prove that $\pi(M)$ realizes the same translation as M. This follows from Claim 1 for $(q, \varphi) = (q_0, \varnothing)$.

Claim 1. Let $s \in T_\Sigma$, $q \in Q^{(n)}$, $n \geq 0$, and $p = h(s)$. For every $\varphi \in \Phi_{q,p}$, $\pi(M)_{(q,\varphi)}(s) = M_q(s)\Pi'$, where $\Pi' = [y_j \leftarrow \varphi(j)[\![\langle\!\langle r, p\rangle\!\rangle \leftarrow M_r(s) \mid r \in Q]\!] \mid j \in [n]]$.

This claim is proved by induction on the structure of s. Let the induction hypothesis be denoted by IH1. Let $s = \sigma(s_1, \ldots, s_k)$ with $\sigma \in \Sigma^{(k)}$, $k \geq 0$, and $s_1, \ldots, s_k \in T_\Sigma$. For $i \in [k]$ let $p_i = h(s_i)$ and let $m = \mathrm{rank}_{Q'}((q, \varphi))$.

By Lemma 3.5, $\pi(M)_{(q,\varphi)}(\sigma(s_1, \ldots, s_k)) = \mathrm{rhs}_{\pi(M)}((q, \varphi), \sigma, \langle p_1, \ldots, p_k\rangle)[\![\_]\!]$, where $[\![\_]\!] = [\![\langle r, x_i\rangle \leftarrow \pi(M)_r(s_i) \mid \langle r, x_i\rangle \in \langle Q', X_k\rangle]\!]$. By the definition of the right-hand sides of the rules of $\pi(M)$ we get $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)[\![\_]\!]$, where repl, ρ, and Π are as in the definition of the rules of $\pi(M)$.

For $t = \mathrm{rhs}(\rho)\Pi$ it follows from Claim 2 that $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)[\![\_]\!] = \mathrm{rhs}(\rho)\Pi[\![\ldots]\!]$, where $[\![\ldots]\!] = [\![\langle r, x_i\rangle \leftarrow M_r(s_i) \mid \langle r, x_i\rangle \in \langle Q, X_k\rangle]\!]$. If we apply $[\![\ldots]\!]$ to $\mathrm{rhs}(\rho)\Pi$ and use Lemma 3.5 for M, then we get $M_q(s)\Pi'$, which proves Claim 1.

Claim 2. Let $t \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$ be a subtree of $\mathrm{rhs}(\rho)\Pi$. Then $\mathrm{repl}(t)[\![\_]\!] = t[\![\ldots]\!]$.

This claim is proved by induction on the structure of t. The induction hypothesis is denoted by IH2.

If $t \in Y_m$, then $\mathrm{repl}(t)[\![\_]\!] = t[\![\_]\!] = t = t[\![\ldots]\!]$. If $t = \delta(t_1, \ldots, t_l)$ with $\delta \in \Delta^{(l)}$, $l \geq 0$, and $t_1, \ldots, t_l \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$, then $\mathrm{repl}(\delta(t_1, \ldots, t_l))[\![\_]\!]$ equals $\delta(\mathrm{repl}(t_1)[\![\_]\!], \ldots, \mathrm{repl}(t_l)[\![\_]\!])$. By IH2 this equals $\delta(t_1[\![\ldots]\!], \ldots, t_l[\![\ldots]\!]) = t[\![\ldots]\!]$.

If $t = \langle q', x_i\rangle(t_1, \ldots, t_l)$ with $\langle q', x_i\rangle \in \langle Q, X_k\rangle^{(l)}$, $l \geq 0$, and $t_1, \ldots, t_l \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$, then $\mathrm{repl}(t)[\![\_]\!]$ equals $\langle (q', \varphi'), x_i\rangle(\mathrm{repl}(t_{j_1}), \ldots, \mathrm{repl}(t_{j_\mu}))[\![\_]\!]$ with $\{j_1, \ldots, j_\mu\} = \{j \in [l] \mid \mathrm{Arg}(q', j, p_i) \text{ is infinite}\}$ and φ' as in the definition of repl. Applying the substitution $[\![\_]\!]$ we get

$$\pi(M)_{(q',\varphi')}(s_i)[y_\eta \leftarrow \mathrm{repl}(t_{j_\eta})[\![\_]\!] \mid \eta \in [\mu]].$$

Since $\varphi' \in \Phi_{q',p_i}$ (as shown for the well-definedness of $\pi(M)$), we can apply IH1 to $\pi(M)_{(q',\varphi')}(s_i)$ and IH2 to $\mathrm{repl}(t_{j_\eta}[\![\_]\!])$ to get

$$M_{q'}(s_i)\Pi''[y_\eta \leftarrow t_{j_\eta}[\![\ldots]\!] \mid \eta \in [\mu]]$$

with $\Pi'' = [y_j \leftarrow \varphi'(j)[\![.]\!] \mid j \in [l]]$ and $[\![.]\!] = [\![\langle\!\langle r, p_i\rangle\!\rangle \leftarrow M_r(s_i) \mid r \in Q]\!]$.

By the definition of φ' we can write this as

$$M_{q'}(s_i)[y_j \leftarrow t_j[\![i]\!][\![.]\!] \mid j \in [l], j \neq j_\eta \text{ for } \eta \in [\mu]] \\ [y_{j_\eta} \leftarrow y_\eta \mid \eta \in [\mu]]\,[y_\eta \leftarrow t_{j_\eta}[\![\ldots]\!] \mid \eta \in [\mu]].$$

Since $\varphi' \in \Phi_{q',p_i}$, t_j is in $T_{\langle Q, \{x_i\}\rangle \cup \Delta}$ for $j \neq j_\eta$. Therefore, in $t_j[\![i]\!][\![.]\!] = t_j[\![\langle r, x_i\rangle \leftarrow M_r(s_i) \mid r \in Q]\!]$ we can extend the substitution to all elements of $\langle Q, X_k\rangle$ to get $t_j[\![\ldots]\!]$. Altogether we get

$$M_{q'}(s_i)[y_j \leftarrow t_j[\![\ldots]\!] \mid j \in [l], j \neq j_\eta \text{ for } \eta \in [\mu]][y_{j_\eta} \leftarrow t_{j_\eta}[\![\ldots]\!] \mid \eta \in [\mu]],$$

which equals $M_{q'}(s_i)[y_j \leftarrow t_j[\![\ldots]\!] \mid j \in [l]] = \langle q', x_i\rangle(t_1, \ldots, t_l)[\![\ldots]\!]$. This ends the proof of Claim 2.

Nondeleting of $\pi(M)$. Consider the $((q,\varphi),\sigma,\langle p_1,\dots,p_k\rangle)$-rule r of $\pi(M)$ and let $\varphi^{-1}(Y_m)=\{j_1,\dots,j_m\}$ with $j_1<\cdots<j_m$. Let $\nu\in[m]$. If r is a dummy rule, then $\#_{y_\nu}(\mathrm{rhs}(r))=1$. Otherwise $\mathrm{rhs}(r)=\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)$, where ρ is the $(q,\sigma,\langle p_1,\dots,p_k\rangle)$-rule of M. Since M is nondeleting, y_{j_ν} occurs in $\mathrm{rhs}(\rho)$. Since $\varphi\in\Phi_{q,p}$, $\varphi(j_\nu)=y_\nu$; this means that the substitution Π replaces y_{j_ν} by y_ν, and hence y_ν occurs in $\mathrm{rhs}(\rho)\Pi$. To show that y_ν occurs in $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)$, we prove that for $t\in T_{\langle Q,X_k\rangle\cup\Delta}(Y_m)$, if y_ν occurs in t, then it also occurs in $\mathrm{repl}(t)$. The proof is by induction on the structure of t. It is obvious for $t\in Y_m$ and $t=\delta(t_1,\dots,t_l)$. For $t=\langle q',x_i\rangle(t_1,\dots,t_l)$, let $j\in[l]$ such that y_ν occurs in t_j, and let φ' be as in the definition of repl. By induction, y_ν occurs in $\mathrm{repl}(t_j)$. Then y_ν occurs also in $t_j[\![i]\!]$, where $[\![i]\!]$ is as in the definition of repl. This means that $t_j[\![i]\!]\notin T_{\langle\!\langle Q,\{p_i\}\rangle\!\rangle\cup\Delta}$ and since $\varphi'\in\Phi_{q',p_i}$, this implies that $\varphi'(j)=y_\eta$ for some $\eta\in[\mu]$ with $j=j'_\eta$, where $\{j'_1,\dots,j'_\mu\}={\varphi'}^{-1}(Y_\mu)$ and $j'_1<\cdots<j'_\mu$. By the definition of repl, $\mathrm{repl}(t_{j'_\eta})=\mathrm{repl}(t_j)$ is a subtree of $\mathrm{repl}(t)$ and therefore y_ν occurs in $\mathrm{repl}(t)$.

Nonerasing of $\pi(M)$. Clearly, from the definition of repl, if $\mathrm{repl}(t)\in Y$, then $t\in Y$. Hence $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)\in Y$ implies $\mathrm{rhs}(\rho)\Pi\in Y$ and so, obviously, $\mathrm{rhs}(\rho)\in Y$. Thus, since M is nonerasing, so is $\pi(M)$.

I-properness of $\pi(M)$. Since $\pi(M)$ has the same look-ahead automaton as M, point (iii) of i-properness is preserved. It follows from the definition of Π and repl and from i-properness of M that no (q_0,φ) appears in the right-hand side of a rule of $\pi(M)$. Using Lemma 4.3 (and the fact that, in the definition of $\mathrm{repl}(t)$, $\varphi'\in\Phi_{q',p_i}$) it is not difficult to see that if $\langle\!\langle (q,\varphi),p\rangle\!\rangle$ is reachable, then $\varphi\in\Phi_{q,p}$ and hence, by the definition of $\Phi_{q,p}$, $\langle\!\langle q,p\rangle\!\rangle$ is reachable. Also by Lemma 4.3, if $(q,\varphi)\neq(q_0,\varnothing)$, then (q,φ) appears in the right-hand side of a rule of $\pi(M)$, and so $q\neq q_0$. By Claim 1, $\pi(M)_{(q,\varphi)}(s)=M_q(s)\Pi'$ with $\Pi'=[y_j\leftarrow\varphi(j)[\![\langle\!\langle r,p\rangle\!\rangle\leftarrow M_r(s)\mid r\in Q]\!]\mid j\in[n]]$. Since $\mathrm{size}(M_q(s)\Pi')\geq\mathrm{size}(M_q(s))$, $\mathrm{Out}((q,\varphi),p)=\{M_q(s)\Pi'\mid s\in L_p\}$ is infinite if $\{M_q(s)\mid s\in L_p\}=\mathrm{Out}(q,p)$ is infinite, which holds by i-properness of M.

P-properness. By constructing $\pi(M)$ we have kept only those parameter positions j of q for which $\mathrm{Arg}(q,j,p)$ is infinite. But even if $\mathrm{Arg}(q,j,p)$ is infinite, there might be a $\varphi\in\Phi_{q,p}$ for which $\mathrm{Arg}((q,\varphi),j,p)$ is finite. This means that $\pi(M)$ need not be p-proper yet (see Example 5.10), and, as in the case of i-properness, we have to iterate the application of π. For the termination condition of this iteration we only need to consider particular states which are actually used in the derivations of $\pi^k(M)$. Denote the state $(\cdots((q,\varphi_1),\varphi_2)\dots,\varphi_k)$ of $\pi^k(M)$ by $(q,\varphi_1,\dots,\varphi_k)$. The state $(q,\varphi_1,\dots,\varphi_k)$ is *p-uniform* if for each $0\leq i\leq k-1$, $\varphi_{i+1}\in\Phi_{(q,\varphi_1,\dots,\varphi_i),p}$. We iterate the application of π until we obtain the $\mathrm{MTT}^{\mathrm{R}}$ N (with set of states Q_N) such that

$$(*)\quad \text{for every } p\in P \text{ and } p\text{-uniform state } (q,\varphi) \text{ of } M'=\pi(N),\qquad \mathrm{rank}_{Q'}((q,\varphi))=\mathrm{rank}_{Q_N}(q),$$

where Q' is the set of states of M'.

Let us now show that, indeed, after a finite number of applications of π, $(*)$ holds. For $q\in Q$ and $p\in P$, define the tree $T_{q,p}$ as follows. For $k\geq 0$, the state $(q,\varphi_1,\dots,\varphi_k)$ of $\pi^k(M)$ is a node of $T_{q,p}$ if it is p-uniform and there is a p-uniform state $(q,\varphi_1,\dots,\varphi_k,\dots,\varphi_l)$ of $\pi^l(M)$ with $l>k$ which is of smaller rank than $(q,\varphi_1,\dots,\varphi_k)$. There is an edge in $T_{q,p}$ from every node $(q,\varphi_1,\dots,\varphi_k)$ to every node $(q,\varphi_1,\dots,\varphi_k,\varphi_{k+1})$. Clearly, if $T_{q,p}$ is finite for every $q\in Q$ and $p\in P$, then the iteration of π terminates: Let l be maximal such that $(q,\varphi_1,\dots,\varphi_l)$ is a leaf of $T_{q,p}$ for some $q\in Q$ and $p\in P$. Then the statement in $(*)$ holds for $N=\pi^{l+1}(M)$, because no p-uniform state $(q, \varphi_1, \ldots, \varphi_l, \varphi_{l+1})$ is a node of $T_{q,p}$ and hence, by the definition of the nodes of $T_{q,p}$, every p-uniform state $(q, \varphi_1, \ldots, \varphi_{l+1}, \varphi_{l+2})$ has the same rank as $(q, \varphi_1, \ldots, \varphi_{l+1})$. To show the finiteness of $T_{q,p}$ it suffices, by König's lemma, to show that every path ρ of $T_{q,p}$ is finite. Assume to the contrary that ρ is infinite. Let $u = (q, \varphi_1, \ldots, \varphi_k)$ be a node of ρ. Then there is a descendant of u on the path ρ, that has lower rank than u. This can be seen as follows: By the definition of the node u, there is a p-uniform state $(q, \varphi_1, \ldots, \varphi_k, \ldots, \varphi_l)$ of $\pi^l(M)$, $l > k$, which has lower rank than u. Now, for each $i \in \{k+1, \ldots, l\}$ such that $v = (q, \varphi_1, \ldots, \varphi_k, \ldots, \varphi_{i-1})$ is on the path ρ, either $v' = (v, \varphi_i) = (q, \varphi_1, \ldots, \varphi_k, \ldots, \varphi_i)$ has the same rank as v and then v' is on the path ρ because $\Phi_{v,p} = \{v'\}$ by the definition of $\Phi_{v,p}$ or v' has a lower rank n than v, and then, by the definition of $\Phi_{v,p}$, each state (v, φ) has rank n, in particular the child of v that is on the path ρ. Since each node u of ρ has a descendant on ρ that has a lower rank than u, there is an infinite sequence of nodes on ρ with strictly decreasing ranks. This contradicts the finiteness of the rank of q.

Before we show that M' is p-proper, we prove a claim about p-uniformity.

Claim 3. Let $k \geq 0$, let q be a state of $\pi^k(M)$, and let $p \in P$.

(i) If $\langle q, x_i \rangle$ appears in the right-hand side of a $(q', \sigma, \langle p_1, \ldots, p_{k'} \rangle)$-rule of $\pi^k(M)$ for some state q' of $\pi^k(M)$, $k' \geq 0$, $i \in [k']$, and $p_1, \ldots, p_{k'} \in P$, then q is p_i-uniform.

(ii) If $\langle\!\langle q, p \rangle\!\rangle$ is reachable (by $\pi^k(M)$), then q is p-uniform.

The proof of part (i) of Claim 3 is by induction on k. For $k = 0$, every state is p-uniform for all $p \in P$, and thus the statement holds. Now assume the statement holds for $\pi^k(M)$. If $\langle (q, \varphi), x_i \rangle$ appears in the right-hand side ζ of the $((q', \varphi'), \sigma, \langle p_1, \ldots, p_{k'} \rangle)$-rule of $\pi(\pi^k(M))$, then, by the definition of the rules of $\pi(\pi^k(M))$, ζ is of the form $\text{repl}(\text{rhs}(\rho)\Pi)$, where ρ is the $(q', \sigma, \langle p_1, \ldots, p_{k'} \rangle)$-rule of $\pi^k(M)$. Thus, by the definition of repl and Π, $\langle q, x_i \rangle$ occurs in $\text{rhs}(\rho)$, which means, by induction, that q is p_i-uniform. In the proof of well-definedness of $\pi(M)$ it is shown that $\varphi \in \Phi_{q,p_i}$, and hence also (q, φ) is p_i-uniform. This proves part (i) of the claim. To prove part (ii), we may assume that $q \neq r_0$, the initial state of $\pi^k(M)$; in fact, $r_0 = (q_0, \varnothing, \ldots, \varnothing)$ is p-uniform for every p. If $\langle\!\langle q, p \rangle\!\rangle$ is reachable (by $\pi^k(M)$), then, by definition, it appears in $\widehat{\pi^k(M)}_{r_0}(s[u \leftarrow p])$ for some tree s and node u of s, where $\widehat{\pi^k(M)}$ denotes the extension of $\pi^k(M)$. Since $q \neq r_0$, u must be of the form $u'j$ with $u' \in \mathbb{N}^*$ and $j \geq 1$. Hence, by Lemma 4.3, $\langle q, x_j \rangle$ must occur in the right-hand side of some rule of $\pi^k(M)$ with look-ahead $\langle p_1, \ldots, p_l \rangle$, $l \geq 1$, and $p_j = p$. By part (i) of the claim this implies that q is p-uniform. This concludes the proof of Claim 3.

Let us now prove (i) of p-properness for N. Let $\langle\!\langle q, p \rangle\!\rangle$ be reachable (by N). By Claim 3(ii), q is p-uniform. Since $\langle\!\langle q, p \rangle\!\rangle$ is reachable, the set $\Phi_{q,p}$ must, by definition, contain some element φ. Then (q, φ) is p-uniform, and it follows from $(*)$ that $n = |\varphi^{-1}(Y)|$ and thus $\{j \in [n] \mid \text{Arg}(q, j, p) \text{ is infinite}\} = \{1, \ldots, n\}$. Thus (i) of p-properness holds for N. Now consider M'. Note that, by the previous argument, if (q, φ) is a p-uniform state of M', then $\varphi = \varphi_n$, where $q \in Q_N^{(n)}$ and $\varphi_n(j) = y_j$ for every $j \in [n]$. Clearly, (i) of p-properness also holds for M'. Formally this can be shown by proving that $\text{Arg}((q, \varphi_n), j, p) = \text{Arg}(q, j, p)[\![\text{rel}]\!]$, where $[\![\text{rel}]\!]$ denotes the relabeling $[\![\langle\!\langle q', p \rangle\!\rangle \leftarrow \langle\!\langle (q', \varphi_{n'}), p \rangle\!\rangle \mid q' \in Q_N^{(n')}, n' \geq 0]\!]$. This follows from Claim 4 (for q equal to the initial state of N and φ equal to $\varnothing$).

Claim 4. Let $s \in T_\Sigma$, $u \in V(s)$, and $p \in P$, and let (q, φ) be an $h(s[u \leftarrow p])$-uniform state of M'. Then

$$\hat{M}'_{(q,\varphi)}(s[u \leftarrow p]) = \hat{N}_q(s[u \leftarrow p])[\![\text{rel}]\!].$$

The proof is by induction on the structure of s. Let $s = \sigma(s_1, \ldots, s_k)$ with $\sigma \in \Sigma^{(k)}$, $k \geq 0$, and $s_1, \ldots, s_k \in T_\Sigma$. For $u = \varepsilon$ we get $\hat{M}'_{(q,\varphi)}(s[u \leftarrow p]) = \langle\!\langle (q,\varphi), p\rangle\!\rangle$. Since $\varphi = \varphi_n$, where n is the rank of q, $\langle\!\langle (q,\varphi), p\rangle\!\rangle = \langle\!\langle q, p\rangle\!\rangle[\![\mathrm{rel}]\!] = \hat{N}_q(s[u \leftarrow p])[\![\mathrm{rel}]\!]$. For $u = ju'$ with $j \geq 1$ and $u' \in \mathbb{N}^*$, $s[u \leftarrow p] = \sigma(\tilde{s}_1, \ldots, \tilde{s}_k)$ with $\tilde{s}_j = s_j[u' \leftarrow p]$ and $\tilde{s}_i = s_i$ for $i \in [k] - \{j\}$. By Lemma 3.5 and the definition of the right-hand sides of M', $\hat{M}'_{(q,\varphi)}(s[u \leftarrow p]) = \mathrm{repl}(\mathrm{rhs}(\rho)\Pi)[\![\_]\!]$, where ρ is the $(q, \sigma, \langle h(\tilde{s}_1), \ldots, h(\tilde{s}_k)\rangle)$-rule of N and $[\![\_]\!] = [\![\langle (q',\varphi'), x_i\rangle \leftarrow \hat{M}'_{(q',\varphi')}(\tilde{s}_i) \mid \langle (q',\varphi'), x_i\rangle \in \langle Q', X_k\rangle]\!]$. By Claim 3(i), if $\langle (q',\varphi'), x_i\rangle$ occurs in $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)$, then (q',φ') is $h(\tilde{s}_i)$-uniform and, by the argument given above Claim 4, $\varphi' = \varphi_{n'}$, where $q' \in Q_N^{(n')}$. Clearly, $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)$ equals $\mathrm{rhs}(\rho)[\![\ ]\!]$ with $[\![\ ]\!] = [\![\langle q', x_i\rangle \leftarrow \langle (q', \varphi_{n'}), x_i\rangle \mid \langle q', x_i\rangle \in \langle Q_N, X_k\rangle^{(n')}, n' \geq 0]\!]$. Furthermore, we can restrict the substitution $[\![\_]\!]$ to those $\langle (q',\varphi'), x_i\rangle$ which occur in $\mathrm{repl}(\mathrm{rhs}(\rho)\Pi)$ and then apply the induction hypothesis to $\tilde{s}_j = s_j[u \leftarrow p]$. If we combine the resulting substitution with $[\![\ ]\!]$ and apply Claim 1 to $\tilde{s}_i = s_i$ for $i \in [k]-\{j\}$ (where Π' is the identity), then we get $\mathrm{rhs}(\rho)[\![\langle q', x_i\rangle \leftarrow \hat{N}_{q'}(\tilde{s}_i)[\![\mathrm{rel}]\!] \mid \langle q', x_i\rangle \in \langle Q_N, X_k\rangle$ occurs in $\mathrm{rhs}(\rho)]\!] = \mathrm{rhs}(\rho)[\![\langle q', x_i\rangle \leftarrow \hat{N}_{q'}(\tilde{s}_i)[\![\mathrm{rel}]\!] \mid \langle q', x_i\rangle \in \langle Q_N, X_k\rangle]\!]$, which equals $\hat{N}_q(s[u \leftarrow p])[\![\mathrm{rel}]\!]$. This proves Claim 4.

To show (ii) of p-properness of M', note that if $\varphi \in \Phi_{q,p}$, then $\langle\!\langle q, p\rangle\!\rangle$ is reachable (by N), and hence, by Claim 3(ii), q is p-uniform; then also (q,φ) is p-uniform, $\varphi = \varphi_n$, and, by Claim 4, $\langle\!\langle (q,\varphi), p\rangle\!\rangle$ is reachable (by M'). Thus, if $\langle\!\langle (q,\varphi), p\rangle\!\rangle$ is not reachable, then $\varphi \notin \Phi_{q,p}$. This implies a dummy right-hand side for all $((q,\varphi), \sigma, \langle p_1, \ldots, p_k\rangle)$-rules with $h_\sigma(p_1, \ldots, p_k) = p$ and therefore $\#_{y_j}(M'_{(q,\varphi)}(s)) = 1$ for all $s \in L_p$. This proves (ii) of p-properness and concludes the proof of properness of M'. Hence, the lemma holds for $\mathrm{prop}(M) = M'$. $\square$

The following example illustrates the construction of a proper $\mathrm{MTT}^{\mathrm{R}}$ as given in the proof of Theorem 5.9.

Example 5.10. Let $M = (Q, \{p\}, \Sigma, \Delta, q_0, R, h)$ be the MTT with $Q = \{q_0^{(0)}, q^{(2)}\}$, $\Sigma = \{a^{(1)}, b^{(1)}, e^{(0)}\}$, $\Delta = \{\sigma^{(3)}, \gamma^{(1)}, a^{(0)}, b^{(0)}, e^{(0)}\}$, and R consisting of the following rules (where the only look-ahead $\langle p\rangle$ is omitted, as usual):

$$
\begin{array}{lll@{\qquad}lll}
\langle q_0, a(x_1)\rangle & \to & \langle q, x_1\rangle(a,a), & \langle q, a(x_1)\rangle(y_1,y_2) & \to & \sigma(y_1, y_2, \langle q, x_1\rangle(a,a)),\\
\langle q_0, b(x_1)\rangle & \to & \langle q, x_1\rangle(b,b), & \langle q, b(x_1)\rangle(y_1,y_2) & \to & \sigma(y_1, y_2, \langle q, x_1\rangle(b,\gamma(y_2))),\\
\langle q_0, e\rangle & \to & e, & \langle q, e\rangle(y_1,y_2) & \to & \sigma(y_1, y_2, e).
\end{array}
$$

Note that M is productive and i-proper. Let us now construct the MTT $\pi(M)$ as defined in the proof of Theorem 5.9. Clearly, $\mathrm{Arg}(q,1,p) = \{a,b\}$ and $\mathrm{Arg}(q,2,p) = \{\gamma^n(c) \mid n \geq 0, c \in \{a,b\}\}$. Thus, $\Phi_{q,p}$ consists of the two mappings φ_a and φ_b with $\varphi_a(1) = a$, $\varphi_a(2) = y_1$, $\varphi_b(1) = b$, and $\varphi_b(2) = y_1$. Therefore the states of $M_1 = \pi(M)$ are $(q_0, \varnothing)^{(0)}, (q,\varphi_a)^{(1)}, (q,\varphi_b)^{(1)}$, abbreviated by q_0, q_a, q_b, respectively. For every $c \in \{a,b\}$, M_1 has the following rules.

$$
\begin{array}{lll@{\qquad}lll}
\langle q_0, a(x_1)\rangle & \to & \langle q_a, x_1\rangle(a), & \langle q_c, a(x_1)\rangle(y_1) & \to & \sigma(c, y_1, \langle q_a, x_1\rangle(a)),\\
\langle q_0, b(x_1)\rangle & \to & \langle q_b, x_1\rangle(b), & \langle q_c, b(x_1)\rangle(y_1) & \to & \sigma(c, y_1, \langle q_b, x_1\rangle(\gamma(y_1))),\\
\langle q_0, e\rangle & \to & e, & \langle q_c, e\rangle(y_1) & \to & \sigma(c, y_1, e).
\end{array}
$$

Now for M_1, $\mathrm{Arg}(q_a,1,p) = \{a\}$ and $\mathrm{Arg}(q_b,1,p) = \{\gamma^n(c) \mid n \geq 0, c \in \{a,b\}\}$. Since $\langle\!\langle q_a, p\rangle\!\rangle$ is reachable this means that M_1 is not p-proper.

Following the proof of Theorem 5.9, we have to construct the MTT $N = \pi(M_1)$, because $\mathrm{rank}_{Q'}((q,\varphi_a)) < \mathrm{rank}_Q(q)$. Clearly, $\Phi_{q_a,p} = \{\varphi'_a\}$ with $\varphi'_a(1) = a$, and $\Phi_{q_b,p} = \{\varphi_1\}$ with $\varphi_1(1) = y_1$. Thus, the states of N are $(q_0,\varnothing)^{(0)}, (q_a,\varphi'_a)^{(0)}$, and

$(q_b, \varphi_1)^{(1)}$, abbreviated by q_0, q_a, and q_b, respectively. The rules of N are as follows:

$$\begin{array}{lcl}
\langle q_0, a(x_1)\rangle & \to & \langle q_a, x_1\rangle, \\
\langle q_0, b(x_1)\rangle & \to & \langle q_b, x_1\rangle(b), \\
\langle q_0, e\rangle & \to & e, \\
\langle q_a, a(x_1)\rangle & \to & \sigma(a, a, \langle q_a, x_1\rangle), \\
\langle q_b, a(x_1)\rangle(y_1) & \to & \sigma(b, y_1, \langle q_a, x_1\rangle), \\
\langle q_a, b(x_1)\rangle & \to & \sigma(a, a, \langle q_b, x_1\rangle(\gamma(a))), \\
\langle q_b, b(x_1)\rangle(y_1) & \to & \sigma(b, y_1, \langle q_b, x_1\rangle(\gamma(y_1))), \\
\langle q_a, e\rangle & \to & \sigma(a, a, e), \\
\langle q_b, e\rangle(y_1) & \to & \sigma(b, y_1, e).
\end{array}$$

The MTT N is p-proper because $\mathrm{Arg}(q_b, 1, p) = \{\gamma^n(c) \mid n \geq 0, c \in \{a, b\}\}$ (and all elements of $\langle\!\langle Q_N, \{p\}\rangle\!\rangle$ are reachable). It is easy to see that N is equivalent to M.

6. From linear size increase to finite copying. In this section we prove that if a proper $\mathrm{MTT}^{\mathrm{R}}$ M is lsi, then it is finite copying (i.e., both fci and fcp; see section 4.1). The proof is split up into the following three stages, using fnest (see section 4.2) as an intermediate notion:

(I) If M is lsi, then it is fnest.

(II) If M is lsi and fnest, then it is fcp.

(III) If M is lsi, fnest, and fcp, then it is fci.

We first prove (II), then (III), and finally (I). The reason for this order is that the proof of (I) will use results that are proved in (III). The idea in each stage is roughly as follows: First, it is proved that if M's copying is *not* bounded (i.e., M is not fcp, not fci, and not fnest for (II), (III), and (I), respectively), then we can find an input tree in which some part s can be pumped, i.e., repeated; each repetition of s will produce a copy of a certain parameter (for (II)) or of a certain state (for (III) and (I)). Second, it is shown that this repetition gives a size increase that is not linearly bounded (by any c); in this part the properness of M is used: it is shown that for any c we can pick a sufficiently large output tree t, a copy of which is generated with each repetition of s, and a sufficiently large i such that after i repetitions of s the size of the corresponding output tree is larger than c times the size of the input tree.

6.1. From lsi and fnest to fcp (II). We now present (in Lemma 6.2) a pumping lemma for non-fcp $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{fnest}}$s, which allows us to prove (in Theorem 6.3) that if a proper $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{fnest}}$ is lsi, then it is fcp.

First, for an $\mathrm{MTT}^{\mathrm{R}}$ M, consider the number k of occurrences of y_ν in $\hat{M}_r(t[u \leftarrow p])$ with $p = h(t/u)$. Clearly, if $\hat{M}_r(t[u \leftarrow p])$ has a subtree $\langle\!\langle r_1, p\rangle\!\rangle(\xi_1, \ldots, \xi_{m_1})$ such that y_ν occurs in ξ_{ν_1} for some $\nu_1 \in [m_1]$, then, assuming that M is nondeleting, the number of y_ν's in $M_r(t)$ must be at least $k - 1$ plus the number of y_{ν_1}'s in $M_{r_1}(t/u)$. This is proved in the next lemma, in such a way that the idea can be iterated.

LEMMA 6.1. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be a nondeleting MTT^R. For $r_0 \in Q^{(m_0)}$, $r_1 \in Q^{(m_1)}$, $\nu_0 \in [m_0]$, $\nu_1 \in [m_1]$, $t_0 \in T_\Sigma$, $u_1 \in V(t_0)$, and $k \in \mathbb{N}$, let $\mathcal{P}(r_0, \nu_0, t_0, r_1, \nu_1, u_1, k)$ be the following statement, with p_1 denoting $h(t_0/u_1)$:*

$\#_{y_{\nu_0}}(\hat{M}_{r_0}(t_0[u_1 \leftarrow p_1])) \geq k$ and $\hat{M}_{r_0}(t_0[u_1 \leftarrow p_1])$ has a subtree $\langle\!\langle r_1, p_1\rangle\!\rangle(\xi_1, \ldots, \xi_{m_1})$ for certain $\xi_1, \ldots, \xi_{m_1}$ such that $\#_{y_{\nu_0}}(\xi_{\nu_1}) \geq 1$.

Let $r_2 \in Q^{(m_2)}$, $\nu_2 \in [m_2]$, $u_2 \in V(t_0/u_1)$, and $l \in \mathbb{N}$. If $\mathcal{P}(r_0, \nu_0, t_0, r_1, \nu_1, u_1, k)$ and $\mathcal{P}(r_1, \nu_1, t_0/u_1, r_2, \nu_2, u_2, l)$, then $\mathcal{P}(r_0, \nu_0, t_0, r_2, \nu_2, u_1u_2, k + l - 1)$.

Proof. Note that $t_0/u_1u_2 = (t_0/u_1)/u_2$. Let $t_1 = t_0/u_1$, $p_1 = h(t_1)$, and $p_2 = h(t_0/u_1u_2) = h(t_1/u_2)$. By Lemma 4.2, $\hat{M}_{r_0}(t_0[u_1u_2 \leftarrow p_2])$ equals $t[\![\ldots]\!]$ with $t = \hat{M}_{r_0}(t_0[u_1 \leftarrow p_1])$ and $[\![\ldots]\!] = [\![\langle\!\langle q', p_1\rangle\!\rangle \leftarrow \hat{M}_{q'}(t_1[u_2 \leftarrow p_2]) \mid q' \in Q]\!]$. We use Lemma 2.6 to compute the number of occurrences of y_{ν_0}'s in this tree. By the first assumption, t has at least k leaves $u \in V_{y_{\nu_0}}(t)$, and it has a subtree $\langle\!\langle r_1, p_1\rangle\!\rangle(\xi_1, \ldots, \xi_{m_1})$ with $\#_{y_{\nu_0}}(\xi_{\nu_1}) \geq 1$. Thus, t has a leaf $u \in V_{y_{\nu_0}}(t)$ such that $\prod F_{t,u}^{[\![\ldots]\!]} \geq \#_{y_{\nu_1}}(\hat{M}_{r_1}(t_1[u_2 \leftarrow p_2]))$, which is $\geq l$ by the second assumption. Hence, $S_1^{y_{\nu_0}} + S_2^{y_{\nu_0}}$ of Lemma 2.6 equals $S_1^{y_{\nu_0}} \geq k - 1 + l$. We have used the fact that $\#_{y_\nu}(\hat{M}_{q'}(t_1[u_2 \leftarrow p_2])) \geq 1$ for all ν and q', which follows from Lemma 3.11(1) because M is nondeleting (and hence so is $\hat{M}$).

The substitution $[\![\ldots]\!]$ is nondeleting, because $\hat{M}$ is nondeleting. Thus, since t has a subtree $\langle\!\langle r_1, p_1\rangle\!\rangle(\xi_1, \ldots, \xi_{m_1})$, it follows from Lemma 2.1 that $\hat{M}_{r_0}(t_0[u_1u_2 \leftarrow p_2]) = t[\![\ldots]\!]$ has a subtree $\langle\!\langle r_1, p_1\rangle\!\rangle(\xi_1, \ldots, \xi_{m_1})[\![\ldots]\!] = \hat{M}_{r_1}(t_1[u_2 \leftarrow p_2])[\ldots]$, where $[\ldots]$ denotes $[y_j \leftarrow \xi_j[\![\ldots]\!] \mid j \in [m_1]]$.

By the second assumption, $\hat{M}_{r_1}(t_1[u_2 \leftarrow p_2])$ has a subtree $\langle\!\langle r_2, p_2\rangle\!\rangle(\zeta_1, \ldots, \zeta_{m_2})$ with $\#_{y_{\nu_1}}(\zeta_{\nu_2}) \geq 1$. Thus we obtain a subtree $\langle\!\langle r_2, p_2\rangle\!\rangle(\zeta_1[\ldots], \ldots, \zeta_{m_2}[\ldots])$ and $\zeta_{\nu_2}[\ldots]$ has a subtree $\xi_{\nu_1}[\![\ldots]\!]$ which contains y_{ν_0} (because $\#_{y_{\nu_0}}(\xi_{\nu_1}) \geq 1$ and M is nondeleting). ∎

LEMMA 6.2. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be a nondeleting MTT^R_{fnest} with the property: if $\langle\!\langle q, p\rangle\!\rangle \in \langle\!\langle Q, P\rangle\!\rangle^{(m)}$ is not reachable, then $\#_{y_j}(M_q(s)) \leq 1$ for all $j \in [m]$ and $s \in L_p$ (property* (ii) *of Definition* 5.6 *of p-properness).*

If M is not fcp, then there are $m \geq 1$, $q \in Q^{(m)}$, $j \in [m]$, $s \in T_\Sigma$, $u \in V(s)$, and $p \in P$ such that

(1) $\#_{y_j}(\hat{M}_q(s[u \leftarrow p])) \geq 2$,

(2) $\hat{M}_q(s[u \leftarrow p])$ *has a subtree* $\langle\!\langle q, p\rangle\!\rangle(\xi_1, \ldots, \xi_m)$ *with* $\#_{y_j}(\xi_j) \geq 1$, *and*

(3) $p = h(s) = h(s/u)$.

Proof. We first define an auxiliary notion. For $t \in T_\Sigma$, u an ancestor of $v \in V(t)$, $q \in Q^{(m)}$, $\mu \in [m]$, $q' \in Q^{(m')}$, $\mu' \in [m']$, define $(q, \mu) \rightarrow_{u,v} (q', \mu')$ if, for $\xi_{q,u,v} = \hat{M}_q(t/u[v' \leftarrow p_v])$ with $v = uv'$ and $p_v = h(t/v)$, $\#_{y_\mu}(\xi_{q,u,v}) \geq 2$ and $\xi_{q,u,v}$ has a subtree $\langle\!\langle q', p_v\rangle\!\rangle(\xi_1, \ldots, \xi_{m'})$ such that $\#_{y_\mu}(\xi_{\mu'}) \geq 1$. Note that $(q, \mu) \rightarrow_{u,v} (q', \mu')$ if and only if $\mathcal{P}(q, \mu, t/u, q', \mu', v', 2)$, where $\mathcal{P}$ is the statement of Lemma 6.1. The relation $\rightarrow$ is transitive; i.e., for a descendant w of v,

$$\text{if } (q, \mu) \rightarrow_{u,v} (q', \mu') \text{ and } (q', \mu') \rightarrow_{v,w} (q'', \mu''), \text{ then } (q, \mu) \rightarrow_{u,w} (q'', \mu'').$$

This follows from Lemma 6.1, because $(q, \mu) \rightarrow_{u,v} (q', \mu')$ and $(q', \mu') \rightarrow_{v,w} (q'', \mu'')$ imply that $\mathcal{P}(q, \mu, t/u, q'', \mu'', v'w', 3)$ with $w' \in \mathbb{N}^*$ such that $w = vw'$, and thus $(q, \mu) \rightarrow_{u,w} (q'', \mu'')$.

Assume that M is not fcp. Then, in terms of the $\rightarrow$ notation, the lemma says that there are $m \geq 1$, $q \in Q^{(m)}$, $j \in [m]$, $s \in T_\Sigma$, $u \in V(s)$, and $p \in P$ such that

- $(q, j) \rightarrow_{\varepsilon,u} (q, j)$ (points (1) and (2) of the lemma),
- $p = h(s) = h(s/u)$ (point (3) of the lemma).

Since M is not fcp, for every $n \in \mathbb{N}$, there are $q \in Q^{(m)}$, $j \in [m]$, and $t \in T_\Sigma$ such that $\#_{y_j}(M_q(t)) > n$. The following claim shows that if $\#_{y_j}(M_q(t/u))$ is "large" for a node u of t, then there must be a descendant v of u, a state r, and a parameter y_ν of r such that $(q, j) \rightarrow_{u,v} (r, \nu)$ and $\#_{y_\nu}(M_r(t/v))$ is still "large." The application of this claim can be iterated to show the existence of a sequence of descendants v and a sequence of steps $\rightarrow$, which will eventually lead to a repetition of a state-parameter pair that allows us to define s and u such that (1)–(3) hold.

Let B be a nesting bound for M. Let η be the maximal height of the right-hand side of a rule of M, i.e., $\eta = \max\{\text{height}(\text{rhs}(\rho)) \mid \rho \in R\}$, and let $\kappa \geq 1$ be an upper bound for the number of occurrences of one particular parameter in the right-hand side of a rule of M, i.e., $\#_y(\text{rhs}(\rho)) \leq \kappa$ for every $y \in Y$ and $\rho \in R$.

Claim. For every $c \geq 1$, $t \in T_\Sigma$, $u \in V(t)$, $q \in Q^{(m)}$, and $\mu \in [m]$, if $\#_{y_\mu}(M_q(t/u))$ is greater than $c^{B\eta} \cdot \kappa^B$, then there exist a descendant v of u, a state $r \in Q^{(m')}$, and a $\nu \in [m']$ such that $(q, \mu) \to_{u,v} (r, \nu)$ and $\#_{y_\nu}(M_r(t/v)) > c$.

Proof of the claim. Let w be a longest descendant of u such that $\#_{y_\mu}(\xi_{q,u,w}) = 1$. Clearly, such a w exists, because $\#_{y_\mu}(\xi_{q,u,u}) = 1$. Then there must be a child v of w that satisfies the requirements of the claim. Assume to the contrary, that if v is a child of w, then it does *not* satisfy the requirements, i.e., for every $r \in Q^{(m')}$ and $\nu \in [m']$ with $(q, \mu) \to_{u,v} (r, \nu)$, $\#_{y_\nu}(M_r(t/v)) \leq c$. This will lead to a contradiction.

By Lemmas 4.2 (applied to t/u and w) and 3.5,

$$M_q(t/u) = \xi_{q,u,w}[\![\text{rhs}]\!][\![\ldots]\!],$$

where $[\![\text{rhs}]\!] = [\![\langle\langle r, p_w\rangle\rangle \leftarrow \text{rhs}_M(r, \sigma, \langle p_1, \ldots, p_k\rangle) \mid r \in Q]\!]$ with $\sigma = t[w] \in \Sigma^{(k)}$, $k \geq 0$, $p_w = h(t/w)$, $p_i = h(t/wi)$ for $i \in [k]$, and $[\![\ldots]\!] = [\![\langle r, x_i\rangle \leftarrow M_r(t/wi) \mid \langle r, x_i\rangle \in \langle Q, X_k\rangle]\!]$. Now, $\#_{y_\mu}(\xi_{q,u,w}[\![\text{rhs}]\!]) \leq \kappa^B$. This is true because by Lemma 2.6, $\#_{y_\mu}(\xi_{q,u,w}[\![\text{rhs}]\!]) = S_1^{y_\mu} = \sum_{z \in V_{y_\mu}(\xi_{q,u,w})} \prod F^{[\![\text{rhs}]\!]}_{\xi_{q,u,w},z}$, which equals $\prod F^{[\![\text{rhs}]\!]}_{\xi_{q,u,w},z}$ for the unique z with $V_{y_\mu}(\xi_{q,u,w}) = \{z\}$. Since $\#_{y_\mu}(M_q(t/u)) > 1$, $\langle\langle q, h(t/u)\rangle\rangle$ is reachable by the assumption of the lemma. Thus, by Lemma 4.13, there are at most B occurrences of elements of $\langle\langle Q, \{p_w\}\rangle\rangle$ on the label path $\text{lpath}(\xi_{q,u,w}, z)$. Hence, $\prod F^{[\![\text{rhs}]\!]}_{\xi_{q,u,w},z}$ is the product of at most B numbers $\#_{y_\nu}(\text{rhs}_M(r, \sigma, \langle p_1, \ldots, p_k\rangle)) \leq \kappa$ for $r \in Q$ and $\nu \in [\text{rank}_Q(r)]$, and therefore $\prod F^{[\![\text{rhs}]\!]}_{\xi_{q,u,w},z} \leq \kappa^B$.

Since every label path of $\xi_{q,u,w}$ is of the form $w_0\langle\langle q_1, p_w\rangle\rangle w_1 \cdots \langle\langle q_l, p_w\rangle\rangle w_l$ with $l \leq B$, $q_1, \ldots, q_l \in Q$, and $w_0, \ldots, w_l \in \Delta^*$, it follows from Lemma 2.3(i) that every label path π in $\xi_{q,u,w}[\![\text{rhs}]\!]$ is of the form $w_0 v_1 w_1 \cdots v_l w_l$, where each v_i is a label path in $\text{rhs}_M(q_i, \sigma, \langle p_1, \ldots, p_k\rangle)$. By the definition of η, the length of v_i is $\leq \eta$. Thus, $\#_{\langle Q, X_k\rangle}(\pi) = \sum_{i \in [l]} \#_{\langle Q, X_k\rangle}(v_i) \leq B\eta$.

Let $\zeta = \xi_{q,u,w}[\![\text{rhs}]\!]$. By Lemma 2.6, $\#_{y_\mu}(\zeta[\![\ldots]\!]) = \sum_{z \in V_{y_\mu}(\zeta)} \prod F^{[\![\ldots]\!]}_{\zeta,z}$. This is $\leq \kappa^B \cdot \prod F^{[\![\ldots]\!]}_{\zeta,z}$, where $z \in V_{y_\mu}(\zeta)$ such that $\prod F^{[\![\ldots]\!]}_{\zeta,z}$ is maximal, because $\#_{y_\mu}(\zeta) \leq \kappa^B$. Since $\#_{\langle Q, X_k\rangle}(\pi) \leq B\eta$ for $\pi = \text{lpath}(\zeta, z)$, $\prod F^{[\![\ldots]\!]}_{\zeta,z}$ is the product of at most $B\eta$ numbers $\#_{y_\nu}(M_r(t/wi))$. Let us now show that each such number is $\leq c$. We need to show that $(q, \mu) \to_{u,wi} (r, \nu)$. By the definition of w, $\#_{y_\mu}(\xi_{q,u,wi}) \neq 1$. Since M is nondeleting it follows from Lemma 3.11(1) that $\#_{y_\mu}(\xi_{q,u,wi}) \geq 1$, and thus $\#_{y_\mu}(\xi_{q,u,wi}) \geq 2$. Since $\langle r, x_i\rangle$ occurs in ζ at some node z' with $z = z'\nu z''$, ζ has a subtree $\langle r, x_i\rangle(\zeta_1, \ldots, \zeta_{m'})$ for some $\zeta_1, \ldots, \zeta_{m'} \in T_{\langle Q, X_k\rangle \cup \Delta}(Y_m)$, and y_μ occurs in ζ_ν. By Lemma 4.3, $\xi_{q,u,wi} = \zeta[\![..]\!][\![i]\!]$, with $[\![..]\!]$ and $[\![i]\!]$ as in that lemma. It follows from Lemma 3.11(1) that $[\![..]\!][\![i]\!]$ is nondeleting. Thus, by Lemma 2.1, $\xi_{q,u,wi}$ has a subtree $\langle\langle r, p_i\rangle\rangle(\zeta_1[\![..]\!][\![i]\!], \ldots, \zeta_{m'}[\![..]\!][\![i]\!])$ and y_μ occurs in $\zeta_\nu[\![..]\!][\![i]\!]$. This proves that $(q, \mu) \to_{u,wi} (r, \nu)$ and thus, by assumption, $\#_{y_\nu}(M_r(t/wi)) \leq c$. We get $\#_{y_\mu}(M_q(t/u)) \leq c^{B\eta} \cdot \kappa^B$, which is a contradiction and ends the proof of the claim.

Now, let $c_0 = 1$ and $c_i = c_{i-1}^{B\eta}\kappa^B$ for $i \geq 1$. Since M is not fcp, for every $n \geq 1$ there exist $r_0 \in Q^{(m_0)}$, $\nu_0 \in [m_0]$, and $t \in T_\Sigma$ such that $\#_{y_{\nu_0}}(M_{r_0}(t)) > c_n$. Let $v_0 = \varepsilon$. We apply the claim for $i = 0, 1, \ldots, n-1$ to $u = v_i$, $q = r_i$, and $\mu = \nu_i$ to obtain that there exist a descendant v_{i+1} of v_i, a state $r_{i+1} \in Q^{(m_{i+1})}$, and $\nu_{i+1} \in [m_{i+1}]$ such that $(r_i, \nu_i) \to_{v_i, v_{i+1}} (r_{i+1}, \nu_{i+1})$ and $\#_{y_{\nu_{i+1}}}(M_{r_{i+1}}(t/v_{i+1})) > c_{n-(i+1)}$.

Take $n = |Q| \cdot \overline{m} \cdot |P|$, where $\overline{m}$ is the maximal rank of a state of M. Then there are indices $0 \leq i < i' \leq n$ such that $q = r_i = r_{i'}$, $j = \nu_i = \nu_{i'}$, and $p = h(t/v_i) = h(t/v_{i'})$. Then $(q, j) \rightarrow_{v_i, v_{i'}} (q, j)$ by the transitivity of $\rightarrow$. Let $s = t/v_i$ and $v_i u = v_{i'}$. Clearly (3) holds. Moreover, in s, $(q, j) \rightarrow_{\varepsilon, u} (q, j)$, which means that (1) and (2) hold. ∎

We now prove that if a proper $\mathrm{MTT}^{\mathrm{R}}_{\mathrm{fnest}}$ M is lsi, then it is fcp; i.e., we prove step (II). The idea is to assume that M is not fcp, and then to "pump" the tree $s[u \leftarrow p]$ of Lemma 6.2 in order to show that this implies that M is not lsi. We use the following notation to pump a tree. For $s \in T_\Sigma$, $u \in V(s)$, $p \in P$, and $s' \in T_\Sigma(P)$, let $s[u \leftarrow p] \bullet s'$ denote $s[u \leftarrow s']$. Let $(s[u \leftarrow p])^0 = p$, and for $n \in \mathbb{N}$ let $(s[u \leftarrow p])^{n+1} = (s[u \leftarrow p]) \bullet (s[u \leftarrow p])^n$. Thus, e.g.,

$$\begin{aligned}
&(s[u \leftarrow p])^1 = s[u \leftarrow p] \bullet p = s[u \leftarrow p],\\
&(s[u \leftarrow p])^2 = (s[u \leftarrow p]) \bullet (s[u \leftarrow p]) = s[u \leftarrow s[u \leftarrow p]], \quad \text{and}\\
&(s[u \leftarrow p])^3 = (s[u \leftarrow p]) \bullet s[u \leftarrow s[u \leftarrow p]] = s[u \leftarrow s[u \leftarrow s[u \leftarrow p]]].
\end{aligned}$$

We will only pump the tree $s[u \leftarrow p]$, for a given $\mathrm{MTT}^{\mathrm{R}}$, if $\hat{h}(s[u \leftarrow p]) = p$. Note that this condition is satisfied in Lemma 6.2 by point (3).

THEOREM 6.3. *Let M be a proper MTT^{R}_{fnest}. If M is lsi, then it is fcp.*

Proof. Let $M = (Q, \Sigma, \Delta, q_0, R, P, h)$ be lsi; i.e., there is a $c \in \mathbb{N}$ such that for every input tree t,

$$(*) \qquad \mathrm{size}(\tau_M(t)) \leq c \cdot \mathrm{size}(t).$$

Assume now that M is not fcp. We will derive a contradiction by constructing an input tree t such that $\mathrm{size}(\tau_M(t)) > c \cdot \mathrm{size}(t)$. Let $q \in Q^{(m)}$, $m \geq 1$, $j \in [m]$, $s \in T_\Sigma$, $p = h(s)$, and $u \in V(s)$ be such that (1)–(3) of Lemma 6.2 hold. Note that since M is proper it satisfies the conditions of Lemma 6.2.

The idea of constructing a t such that $(*)$ does not hold is as follows: Let $s_0 \in T_\Sigma$ and $u_0 \in V(s_0)$ such that

$$(\dagger) \qquad \hat{M}_{q_0}(s_0[u_0 \leftarrow p]) \text{ has a subtree } \langle\!\langle q, p \rangle\!\rangle(\xi_1, \ldots, \xi_m)$$

for some trees $\xi_1, \ldots, \xi_m$. Consider input trees t_i obtained by i times pumping the tree $s[u \leftarrow p]$ in the tree $s_0[u_0 \leftarrow s]$. Then the size of the trees t_i grows at most linearly with constant $\mathrm{size}(s[u \leftarrow p])$. In the output tree $\tau_M(t_i)$ there are at least i occurrences of the subtree $\xi_j[\![\ldots]\!]$ for some second-order tree substitution $[\![\ldots]\!]$. Hence, the size of the trees $\tau_M(t_i)$ grows at least linearly with constant $\mathrm{size}(\xi_j)$. Thus, if we choose s_0 and u_0 in such a way that $\mathrm{size}(\xi_j)$ is larger than the product of c and $\mathrm{size}(s[u \leftarrow p])$, then $\mathrm{size}(\tau_M(t_i))$ grows faster than $c \cdot \mathrm{size}(t_i)$, which implies that we can find an i such that $(*)$ does not hold for $t = t_i$.

Recall Definition 5.6 of p-properness. In order to choose s_0 and u_0 appropriately we need the set $\mathrm{Arg}(q, j, p)$ to be infinite, i.e., to contain arbitrarily large trees. This is guaranteed by point (i) of Definition 5.6 if $\langle\!\langle q, p \rangle\!\rangle$ is reachable. The latter holds for the following reason: Since M is nondeleting, by Lemma 3.11(1), $\#_{y_\nu}(M_r(s/u)) \geq 1$ for every $r \in Q^{(m')}$ and $\nu \in [m']$. By Lemmas 4.2 and 2.6 and the fact that $\#_{y_j}(\hat{M}_q(s[u \leftarrow p])) \geq 2$ by (1), this implies that $\#_{y_j}(M_q(s)) \geq 2$. Thus, $\langle\!\langle q, p \rangle\!\rangle$ is reachable by point (ii) of Definition 5.6.

We now show the effect of pumping the tree $s[u \leftarrow p]$ in the input tree $s = s[u \leftarrow p] \bullet s/u$. For $i \geq 0$ let $t'_i = (s[u \leftarrow p])^i \bullet s/u$. Then $\#_{y_j}(M_q(t'_i)) > i$. Using the fact that M is nondeleting, this follows (as above, by Lemmas 4.2 and 2.6) from

$\#_{y_j}(\hat{M}_q(t'_i[u^i \leftarrow p])) > i$, which is a consequence of the next claim and the definition of $\mathcal{P}$ (cf. Lemma 6.1).

Claim. For $i \geq 0$, $\mathcal{P}(q, j, t'_i, q, j, u^i, i+1)$.

The proof of this claim is by induction on i. For $i = 0$, $\mathcal{P}(q, j, t'_i, q, j, u^i_1, i+1)$ because $\xi = \hat{M}_q(s/u[\varepsilon \leftarrow p]) = \langle\langle q, p\rangle\rangle(y_1, \ldots, y_m)$ and thus $\#_{y_j}(\xi) \geq 1$ and ξ has a subtree $\langle\langle q, p\rangle\rangle(\xi_1, \ldots, \xi_m)$ with $\#_{y_j}(\xi_j) = \#_{y_j}(y_j) = 1$. For $i+1 > 0$, by induction, $\mathcal{P}(q, j, t'_i, q, j, u^i, i+1)$. Clearly, by (3), $h(t'_{i+1}/u^i) = h(s) = p = h(s/u) = h(t'_i/u^i)$, and $t'_{i+1}[u^i \leftarrow p] = t'_i[u^i \leftarrow p]$. Thus, $\mathcal{P}(q, j, t'_{i+1}, q, j, u^i, i+1)$. By (1) and (2), $\mathcal{P}(q, j, s, q, j, u, 2)$, which is equivalent to $\mathcal{P}(q, j, t'_{i+1}/u^i, q, j, u, 2)$ because $t'_{i+1}/u^i = s$. By Lemma 6.1 this means that $\mathcal{P}(q, j, t'_{i+1}, q, j, u^i u, i+2)$, which concludes the proof of the claim.

Now let $t_i = s_0[u_0 \leftarrow t'_i]$, where $s_0 \in T_\Sigma$ and $u_0 \in V(s_0)$ satisfy (†). Thus, t_i is the result of pumping the tree $s[u \leftarrow p]$ in the input tree $s_0[u_0 \leftarrow s]$. Since $\#_{y_j}(\hat{M}_q(t'_i) > i$, we obtain $\text{size}(\tau_M(t_i)) > i \cdot \text{size}(\xi_j)$ as follows: By Lemma 4.2, $\tau_M(t_i) = \hat{M}_{q_0}(s_0[u_0 \leftarrow p])[\![\ldots]\!]$, where $[\![\ldots]\!] = [\![\langle\langle r, p\rangle\rangle \leftarrow M_r(t'_i) \mid r \in Q]\!]$. By Lemma 2.1, $\hat{M}_{q_0}(s_0[u_0 \leftarrow p])[\![\ldots]\!]$ has a subtree $\xi = \langle\langle q, p\rangle\rangle(\xi_1, \ldots, \xi_m)[\![\ldots]\!] = M_q(t'_i)[y_\nu \leftarrow \xi_\nu[\![\ldots]\!] \mid \nu \in [m]]$. By Lemma 2.4 (summing for all $\delta \in \Delta$), $\text{size}(\xi) = \#_\Delta(\xi) = \#_\Delta(M_q(t'_i)) + \sum_{\nu \in [m]} \#_{y_\nu}(M_q(t'_i)) \cdot \#_\Delta(\xi_\nu[\![\ldots]\!]) \geq \sum_{\nu=j} \#_{y_\nu}(M_q(t'_i)) \cdot \#_\Delta(\xi_\nu[\![\ldots]\!]) = \#_{y_j}(M_q(t'_i)) \cdot \text{size}(\xi_j[\![\ldots]\!])$. Since M is productive, Lemmas 2.7 and 3.11 imply that $\text{size}(\xi_j[\![\ldots]\!]) \geq \text{size}(\xi_j)$. Since $\#_{y_j}(M_q(t'_i)) > i$, this implies that $\text{size}(\tau_M(t_i)) > i \cdot \text{size}(\xi_j)$.

Since $\text{Arg}(q, j, p)$ is infinite, we can choose s_0 and u_0 such that (†) and

$$\text{size}(\xi_j) > c \cdot c_1,$$

where $c_1 = \text{size}(s[u \leftarrow p]) - 1$. Let $i = c(c_0 + c_2)$ for $c_0 = \text{size}(s_0[u_0 \leftarrow p]) - 1$ and $c_2 = \text{size}(s/u)$. Since $\text{size}(t_i) = c_0 + ic_1 + c_2$ this means that $\text{size}(\tau_M(t_i)) > c \cdot \text{size}(t_i)$ because $\text{size}(\tau_M(t_i)) > i \cdot \text{size}(\xi_j) \geq i \cdot (c \cdot c_1 + 1) = icc_1 + c(c_0 + c_2) = c(c_0 + ic_1 + c_2) = c \cdot \text{size}(t_i)$. This contradicts (∗) and concludes the proof. ∎

6.2. From lsi, fnest, and fcp to fci (III). Here we present a pumping lemma for $\text{MTT}^{\text{R}}_{\text{fnest,fcp}}$s that are not fci (Lemma 6.5) and apply it in Lemma 6.6 to show that if an $\text{MTT}^{\text{R}}_{\text{fnest,fcp}}$ is lsi, then it is fci. We first define, in general, what is required of an MTT^{R} in order to get a repetition of states by pumping a part of an input tree; this is called *input pumpable.* It means that there is a state q_1 that is reachable, i.e., appears in $\hat{M}_{q_0}(s_0[u_0 \leftarrow p])$ for some input tree s_0 and node u_0 of s_0 (with $p = h(s_0/u_0)$), and going from node u_0 to node u_0u_1 in s_0, q_1 will generate a copy of itself and of a state q_2; furthermore, the state q_2 generates a copy of itself when going from u_0 to u_0u_1.

DEFINITION 6.4 (input pumpable). *An MTT^R $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ is* input pumpable *if there are $q_1, q_2 \in Q$, $s_0 \in T_\Sigma$, $u_0 \in V(s_0)$, $u_1 \in V(s_0/u_0)$, and $p \in P$ such that the following four conditions hold:*

(1) *$\langle\langle q_1, p\rangle\rangle$ occurs in $\hat{M}_{q_0}(s_0[u_0 \leftarrow p])$,*

(2) *$\langle\langle q_1, p\rangle\rangle$ and $\langle\langle q_2, p\rangle\rangle$ occur at distinct nodes of $\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p])$,*

(3) *$\langle\langle q_2, p\rangle\rangle$ occurs in $\hat{M}_{q_2}(s_0/u_0[u_1 \leftarrow p])$, and*

(4) *$p = h(s_0/u_0) = h(s_0/u_0u_1)$.*

The following pumping lemma can be viewed as a generalization of Lemma 4.2 of [1] from top-down tree transducers to MTTs.

LEMMA 6.5. *Let M be a nondeleting $MTT^R_{fnest,fcp}$. If M is not fci, then it is input pumpable.*

Proof. Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$. We first define some auxiliary notions. Let $t \in T_\Sigma$ and $u, v \in V(t)$ such that u is an ancestor of v, i.e., $v = uv'$ for some $v' \in \mathbb{N}^*$, and let $p_v = h(t/v)$. For $q \in Q$, if $n = \#_{\langle\langle Q,\{p_v\}\rangle\rangle}(\hat{M}_q(t/u[v' \leftarrow p_v]))$, then we say that q *contributes* n *states at* u *to* v. If $n \geq 1$, then we say that q contributes at u to v. For $q, q' \in Q$ we write $q \rightarrow_{u,v} q'$ if $\langle\langle q', p_v\rangle\rangle$ occurs in $\hat{M}_q(t/u[v' \leftarrow p_v])$. For $r_1, r_2 \in Q$ we write $q \rightarrow_{u,v} r_1, r_2$ if $\langle\langle r_1, p_v\rangle\rangle$ and $\langle\langle r_2, p_v\rangle\rangle$ occur at distinct nodes of $\hat{M}_q(t/u[v' \leftarrow p_v])$. Observe the following easy properties:

(P0) $q \rightarrow_{v,v} q'$ if and only if $q = q'$; q contributes one state at v to v.

(P1) $q_0 \rightarrow_{\varepsilon,v} q$ if and only if q occurs in $\mathrm{sts}_M(t, v)$; q_0 contributes $|\mathrm{sts}_M(t, v)|$ states at ε to v.

(P2) q contributes at u to v if and only if there is a $q' \in Q$ such that $q \rightarrow_{u,v} q'$.

Let w be a node of t that is a descendant of u and an ancestor of v.

(P3) If $q \rightarrow_{u,w} q''$ and $q'' \rightarrow_{w,v} q'$, then $q \rightarrow_{u,v} q'$.

(P4) If $q \rightarrow_{u,v} q'$, then there is a $q'' \in Q$ such that $q \rightarrow_{u,w} q''$ and $q'' \rightarrow_{w,v} q'$.

Note that (P3) and (P4) can be proved using Lemma 4.14: Let $w', v'' \in \mathbb{N}^*$ such that $w = uw'$ and $v = wv''$ (and so v' above equals $w'v''$), and let $p_w = h(t/w)$. For (P3), the number $\#_{\langle\langle q'',p_w\rangle\rangle}(\hat{M}_q(t/u[w' \leftarrow p_w]))$ is ≥ 1 because $q \rightarrow_{u,w} q''$, and $\#_{\langle\langle q',p_v\rangle\rangle}(\hat{M}_{q''}(t/w[v'' \leftarrow p_v]))$ is ≥ 1 because $q'' \rightarrow_{w,v} q'$; hence the product of these two numbers is ≥ 1 and so the sum S of Lemma 4.14 is ≥ 1. Thus, by part (i) of that lemma, $\#_{\langle\langle q',p_v\rangle\rangle}(\hat{M}_q(t/u[v' \leftarrow p_v])) \geq 1$, i.e., $q \rightarrow_{u,v} q'$. For (P4), $q \rightarrow_{u,v} q'$ implies that the sum in $(\times)$ of the proof of Lemma 4.14 is ≥ 1, and thus there is an occurrence of some $\langle\langle q'', p_w\rangle\rangle \in \langle\langle Q, \{p_w\}\rangle\rangle$ in $\hat{M}_q(s/u[w' \leftarrow p_w])$ with $\#_{\langle\langle q',p_v\rangle\rangle}(\hat{M}_{q''}(t/w[v'' \leftarrow p_v])) \geq 1$; i.e., there is a $q'' \in Q$ such that $q \rightarrow_{u,w} q''$ and $q'' \rightarrow_{w,v} q'$.

(P5) q contributes ≥ 2 states at u to v if and only if there are $r_1, r_2 \in Q$ such that $q \rightarrow_{u,v} r_1, r_2$.

(P6) Let $r_1', r_2' \in Q$ and w as above. If $q \rightarrow_{u,w} r_1, r_2$ and $r_i \rightarrow_{w,v} r_i'$ for $i \in [2]$, then $q \rightarrow_{u,v} r_1', r_2'$.

Let us prove property (P6). If $r_1' \neq r_2'$, then by (P3), $q \rightarrow_{u,v} r_1'$ and $q \rightarrow_{u,v} r_2'$, which means that $q \rightarrow_{u,v} r_1', r_2'$. Now assume that $r_1' = r_2'$. By Lemma 4.14(i), $\#_{\langle\langle r_1',p_v\rangle\rangle}(\hat{M}_q(t/u[v' \leftarrow p_v]))$ is greater than or equal to

$$(*)\qquad \sum_{r\in Q} \#_{\langle\langle r_1',p_v\rangle\rangle}(\hat{M}_r(t/w[v'' \leftarrow p_v])) \cdot \#_{\langle\langle r,p_w\rangle\rangle}(\hat{M}_q(t/u[w' \leftarrow p_w])),$$

where p_w, w', and v'' are as in the proof of (P3). We distinguish the following two cases:

(i) $r_1 \neq r_2$: For $r = r_1$ and $r = r_2$, $\#_{\langle\langle r,p_w\rangle\rangle}(\hat{M}_q(t/u[w' \leftarrow p_w])) \geq 1$, because $q \rightarrow_{u,w} r_1, r_2$. Thus, the sum in $(*)$ is $\geq \#_{\langle\langle r_1',p_v\rangle\rangle}(\hat{M}_{r_1}(t/w[v'' \leftarrow p_v])) + \#_{\langle\langle r_1',p_v\rangle\rangle}(\hat{M}_{r_2}(t/w[v'' \leftarrow p_v]))$, which is ≥ 2, because $r_i \rightarrow_{w,v} r_i'$ for $i \in [2]$.

(ii) $r_1 = r_2$: For $r = r_1$, $\#_{\langle\langle r,p_w\rangle\rangle}(\hat{M}_q(t/u[w' \leftarrow p_w])) \geq 2$, because $q \rightarrow_{u,w} r_1, r_1$. Thus, the sum in $(*)$ is $\geq \#_{\langle\langle r_1',p_v\rangle\rangle}(\hat{M}_{r_1}(t/w[v'' \leftarrow p_v])) \cdot 2$, which is ≥ 2, because $r_1 \rightarrow_{w,v} r_1'$.

In terms of the $\rightarrow$ notation the four conditions of input pumpability (cf. Definition 6.4) say that there are states q_1 and q_2, a tree $s_0 \in T_\Sigma$, and nodes u_0 and u_0u_1 of s_0 such that

(1) $q_0 \rightarrow_{\varepsilon,u_0} q_1$,

(2) $q_1 \rightarrow_{u_0,u_0u_1} q_1, q_2$,

(3) $q_2 \rightarrow_{u_0,u_0u_1} q_2$, and

(4) $h(s_0/u_0) = h(s_0/u_0u_1)$.

Since M is not fci, arbitrary long state sequences can be generated. Thus, for every $m \geq 1$ there are $t \in T_\Sigma$ and $v \in V(t)$ such that $|\mathrm{sts}_M(t, v)| > m$, which, by (P1), means that q_0 contributes more than m states at ε to v. In Claim 1 below we will show that if a state q contributes "many" states at u to v, then there must be an intermediate node w (a descendant of u and ancestor of v) such that q contributes at least two states at u to w that contribute at w to v, and at least one of these states still contributes "many" states at w to v. The application of this claim can be iterated to show the existence of a sequence of intermediate nodes w, which will eventually lead to an appropriate repetition of states (and look-ahead states) that allows us to define s_0 and nodes u_0, u_0u_1 for which (1)–(4) hold.

Let $\kappa \geq 1$ be an upper bound for the number of occurrences of elements of $\langle Q, \{x_i\}\rangle$ for an $i \geq 1$ in the right-hand side of any rule of R, i.e., $\kappa \geq \#_{\langle Q,\{x_i\}\rangle}(\mathrm{rhs}(\rho))$ for every $\rho \in R$ and $i \geq 1$. Let η be the maximal height of the right-hand side of any rule in R, i.e., $\eta = \max\{\mathrm{height}(\mathrm{rhs}(\rho)) \mid \rho \in R\}$. Let $N \geq 1$ be a parameter copying bound for M and let $B \geq 1$ be a nesting bound for M.

Claim 1. Let $\langle\langle q, p\rangle\rangle \in \langle\langle Q, P\rangle\rangle$ be reachable, $t \in T_\Sigma$, and $u, v \in V(t)$ such that $t/u \in L_p$ and u is an ancestor of v. Let $c \geq 1$. If q contributes more than $(\kappa N^{2B+\eta}) \cdot c$ states at u to v, then there is a proper descendant w of u which is an ancestor of v and there are states $r, r' \in Q$ such that

(a) $q \to_{u,w} r, r'$,

(b) r contributes more than c states at w to v, and

(c) r' contributes at w to v.

Proof of Claim 1. Let w be the first (shortest) descendant of u and ancestor of v such that there are $r_1, r_2 \in Q$ with $q \to_{u,w} r_1, r_2$ and r_1,r_2 contribute at w to v. Clearly such a w exists, because q contributes ≥ 2 states at u to v, and thus, by (P5), there are $r_1, r_2 \in Q$ such that $q \to_{u,v} r_1, r_2$, and, by (P0), r_1, r_2 contribute at v to v. By (P0), q contributes exactly one state at u to u and therefore $w \neq u$. It remains to show that there is an $r \in Q$ such that $q \to_{u,w} r$ and r contributes more than c states at w to v; then r' is chosen to be one of the r_1, r_2 such that (a) holds.

In (sub)Claim 2 below we will show that q contributes at most $\kappa \cdot N^{B+\eta}$ states r at u to w that contribute at w to v. We now show that the number of states that q contributes at u to v is at most N^B times the sum of the contributions of the states r at w to v, and hence that at least one of these r must contribute $> c$ states.

Let $w', v', v'' \in \mathbb{N}^*$ such that $w = uw'$ and $v = uv' = wv''$. Let $p_w = h(t/w)$ and $p_v = h(t/v)$. By assumption, q contributes $> (\kappa N^{2B+\eta}) \cdot c$ states at u to v, i.e., $(\kappa N^{2B+\eta}) \cdot c$ is smaller than $\#_{\langle\langle Q,\{p_v\}\rangle\rangle}(\hat{M}_q(t/u[v' \leftarrow p_v]))$, which, by Lemma 4.14(ii) (using the fact that $\langle\langle q, h(t/u)\rangle\rangle$ is reachable, and summing over all elements of $\langle\langle Q, \{p_v\}\rangle\rangle$), is

$$\leq N^B \cdot \sum_{r \in Q} \#_{\langle\langle Q,\{p_v\}\rangle\rangle}(\hat{M}_r(t/w[v'' \leftarrow p_v])) \cdot \#_{\langle\langle r,p_w\rangle\rangle}(\hat{M}_q(t/u[w' \leftarrow p_w])).$$

If $\#_{\langle\langle Q,\{p_v\}\rangle\rangle}(\hat{M}_r(t/w[v'' \leftarrow p_v])) \neq 0$, then r contributes at w to v. Thus, we can restrict the above sum to states in $Q_{w,v} = \{r \in Q \mid r \text{ contributes at } w \text{ to } v\}$. Now let $r \in Q_{w,v}$ be such that $q \to_{u,w} r$ (i.e., $\#_{\langle\langle r,p_w\rangle\rangle}(\hat{M}_q(t/u[w' \leftarrow p_w])) \geq 1$) and the number of states it contributes at w to v is maximal; i.e., for all $r' \neq r$ with $q \to_{u,w} r'$, $\#_{\langle\langle Q,\{p_v\}\rangle\rangle}(\hat{M}_{r'}(t/w[v'' \leftarrow p_v])) \leq \#_{\langle\langle Q,\{p_v\}\rangle\rangle}(\hat{M}_r(t/w[v'' \leftarrow p_v]))$. Then the above number is

$$\leq N^B \cdot \#_{\langle\langle Q,\{p_v\}\rangle\rangle}(\hat{M}_r(t/w[v'' \leftarrow p_v])) \cdot \#_{\langle\langle Q_{w,v},\{p_w\}\rangle\rangle}(\hat{M}_q(t/u[w' \leftarrow p_w])),$$

which, by Claim 2, is $\leq N^B \cdot \#_{\langle\langle Q,\{p_v\}\rangle\rangle}(\hat{M}_r(t/w[v'' \leftarrow p_v])) \cdot (\kappa N^{B+\eta})$. Thus we get

$c < \#_{\langle\!\langle Q,\{p_v\}\rangle\!\rangle}(\hat{M}_r(t/w[v'' \leftarrow p_v]))$, i.e., r contributes more than c states at w to v, which concludes the proof of Claim 1.

Claim 2. $\#_{\langle\!\langle Q_{w,v},\{p_w\}\rangle\!\rangle}(\hat{M}_q(t/u[w' \leftarrow p_w])) \leq \kappa \cdot N^{B+\eta}$.

Proof of Claim 2. Since $w \neq u$ it follows that $w' \neq \varepsilon$; i.e., there are $i \geq 1$ and $\omega' \in \mathbb{N}^*$ such that $w' = \omega' i$. Let $\omega = u\omega'$; i.e., w is the ith child of ω. In the remainder of this proof we will always write ωi in place of w and $\omega' i$ in place of w', in particular, $p_{\omega i} = p_w$ and $Q_{\omega i,v} = Q_{w,v}$. Let $p_\omega = h(t/\omega)$. Using the fact that $\langle\!\langle q, h(t/u)\rangle\!\rangle$ is reachable, we can apply Lemma 4.14(ii) to t and $u, \omega, \omega i \in V(t)$, summing over all $\langle\!\langle q', p_{\omega i}\rangle\!\rangle$ in $\langle\!\langle Q_{\omega i,v}, \{p_{\omega i}\}\rangle\!\rangle$, to get that $\#_{\langle\!\langle Q_{\omega i,v},\{p_{\omega i}\}\rangle\!\rangle}(\hat{M}_q(t/u[\omega' i \leftarrow p_{\omega i}]))$ is

$$\leq N^B \cdot \sum_{r \in Q} \#_{\langle\!\langle Q_{\omega i,v},\{p_{\omega i}\}\rangle\!\rangle}(\hat{M}_r(t/\omega[i \leftarrow p_{\omega i}])) \cdot \#_{\langle\!\langle r,p_\omega\rangle\!\rangle}(\hat{M}_q(t/u[\omega' \leftarrow p_\omega])).$$

If $\#_{\langle\!\langle Q_{\omega i,v},\{p_{\omega i}\}\rangle\!\rangle}(\hat{M}_r(t/\omega[i \leftarrow p_{\omega i}])) \neq 0$, then there is an occurrence of some $\langle\!\langle r', p_{\omega i}\rangle\!\rangle$ in $\hat{M}_r(t/\omega[i \leftarrow p_{\omega i}])$, i.e., $r \rightarrow_{\omega,\omega i} r'$, and r' contributes at ωi to v, i.e., $r' \rightarrow_{\omega i,v} r''$ for some $r'' \in Q$. Thus, by (P3), $r \rightarrow_{\omega,v} r''$, which means by (P2) that r contributes at ω to v. By the definition of the node ωi there is at most one occurrence of a $\langle\!\langle q', p_\omega\rangle\!\rangle \in \langle\!\langle Q, \{p_\omega\}\rangle\!\rangle$ in $\hat{M}_q(t/u[\omega' \leftarrow p_\omega])$ such that q' contributes at ω to v, and since q contributes at u to v, by (P4) there is at least one such occurrence. Hence, in the above sum there is only one nonzero product, namely, for $r = q'$, and $\#_{\langle\!\langle q',\{p_\omega\}\rangle\!\rangle}(\hat{M}_q(t/u[\omega' \leftarrow p_\omega])) = 1$. We get

$$N^B \cdot \#_{\langle\!\langle Q_{\omega i,v},\{p_{\omega i}\}\rangle\!\rangle}(\hat{M}_{q'}(t/\omega[i \leftarrow p_{\omega i}])) \leq N^B \cdot \#_{\langle\!\langle Q,\{p_{\omega i}\}\rangle\!\rangle}(\hat{M}_{q'}(t/\omega[i \leftarrow p_{\omega i}])).$$

By Lemma 4.3 with $s = t/\omega$ and $u = \varepsilon$, and since $\hat{M}_{q'}(t/\omega[\varepsilon \leftarrow p_\omega]) = \langle\!\langle q', p_\omega\rangle\!\rangle$, the tree $\hat{M}_{q'}(t/\omega[i \leftarrow p_{\omega i}])$ equals $\text{rhs}_M(q', \sigma, \langle p_1, \ldots, p_k\rangle)[\![..]\!][\![i]\!]$, where $[\![..]\!] = [\![\langle r', x_j\rangle \leftarrow M_{r'}(t/\omega j) \mid r' \in Q, j \in [k] - \{i\}]\!]$ and $[\![i]\!] = [\![\langle r', x_i\rangle \leftarrow \langle\!\langle r', p_{\omega i}\rangle\!\rangle \mid r' \in Q]\!]$ with $t[\omega] = \sigma \in \Sigma^{(k)}$, $k \geq 1$, and $p_j = h(t/\omega j)$ for each $j \in [k]$. Thus, $N^B \cdot \#_{\langle\!\langle Q,\{p_{\omega i}\}\rangle\!\rangle}(\hat{M}_{q'}(t/\omega[i \leftarrow p_{\omega i}]))$ equals $N^B \cdot \#_{\langle\!\langle Q,\{p_{\omega i}\}\rangle\!\rangle}(\text{rhs}_M(q', \sigma, \langle p_1, \ldots, p_k\rangle)[\![..]\!][\![i]\!])$, which, avoiding the relabeling $[\![i]\!]$, can be written as

$$N^B \cdot \#_{\langle Q,\{x_i\}\rangle}(\text{rhs}_M(q', \sigma, \langle p_1, \ldots, p_k\rangle)[\![\langle r', x_j\rangle \leftarrow M_{r'}(t/\omega j) \mid r' \in Q, j \neq i]\!]).$$

The application of Lemma 2.6 and the fact that the trees $M_{r'}(t/\omega j)$ do not contain elements of $\langle Q, \{x_i\}\rangle$ gives the number $N^B \cdot \sum_{\tilde{u} \in V_{\langle Q,\{x_i\}\rangle}(\zeta)} \prod F^{[\![..]\!]}_{\zeta,\tilde{u}}$, where $\zeta = \text{rhs}_M(q', \sigma, \langle p_1, \ldots, p_k\rangle)$. Since the height of ζ is at most η, $\prod F^{[\![..]\!]}_{\zeta,\tilde{u}} \leq N^\eta$, and thus the above number is $\leq N^{B+\eta} \cdot |V_{\langle Q,\{x_i\}\rangle}(\zeta)|$, which is $\leq \kappa \cdot N^{B+\eta}$ by the definition of κ. This ends the proof of Claim 2.

Let $\gamma = \kappa N^{2B+\eta}$. Since M is not fci, for every $n \geq 1$ there are $t_n \in T_\Sigma$ and $v_n \in V(t_n)$ such that $|\text{sts}_M(t_n, v_n)| > \gamma^n$. Let $r_0 = q_0$ and $w_0 = \varepsilon$. We now apply Claim 1 for $i = 0, \ldots, n-1$ to $q = r_i$, $p = h(t_n/w_i)$, $t = t_n$, $u = w_i$, $v = v_n$, and $c = \gamma^{n-i-1}$. For $i = 0$ this is possible because $\langle\!\langle q_0, h(t_n)\rangle\!\rangle$ is reachable, and by (P1), q_0 contributes more than γ^n states at ε to v_n. We obtain that there exists a proper descendant w_{i+1} of w_i and states r_{i+1}, r'_{i+1} such that $r_i \rightarrow_{w_i,w_{i+1}} r_{i+1}, r'_{i+1}$, the state r_{i+1} contributes more than γ^{n-i-2} states at w_{i+1} to v_n, and r'_{i+1} contributes at w_{i+1} to v_n. Note that since $q_0 \rightarrow_{\varepsilon,w_{i+1}} r_{i+1}$ and $q_0 \rightarrow_{\varepsilon,w_{i+1}} r'_{i+1}$ by (P3), both r_{i+1} and r'_{i+1} occur in $\text{sts}_M(t_n, w_{i+1})$ by (P1) (and thus $\langle\!\langle r_{i+1}, h(t_n/w_{i+1})\rangle\!\rangle$ is reachable). For an ancestor w of v_n let $\text{csts}(w)$ denote $\text{sts}_M(t_n, w)$ restricted to the states q which contribute at w to v_n (i.e., all states that do not contribute to v_n are erased from

$\mathrm{sts}_M(t_n, w)$). Hence, r occurs in $\mathrm{csts}(w)$ if and only if $q_0 \to_{\varepsilon,w} r \to_{w,v} q$ for some state q. In particular, r_{i+1} and r'_{i+1} occur in $\mathrm{csts}(w_{i+1})$. Figure 6.1 shows the nodes w_i and the corresponding sequences $\mathrm{csts}(w_i)$ with the states r_i, r'_i; the arrows mean $\to_{w_i,w_{i+1}}$.

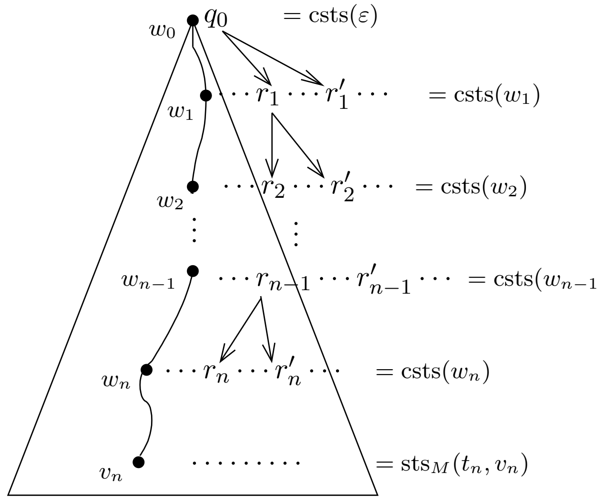

FIG. 6.1. *The tree t_n with contributing states.*

Now take $n = |Q| \cdot |P| \cdot 2^{|Q|}$ and let t_n, v_n, w_i, r_i, and r'_i be as above for $0 \le i \le n$. Clearly this means that there are indices $0 \le i < j \le n$ such that

- $r_i = r_j$,
- $p = h(t_n/w_i) = h(t_n/w_j)$, and
- $\{r \in Q \mid r \text{ occurs in } \mathrm{csts}(w_i)\} = \{r \in Q \mid r \text{ occurs in } \mathrm{csts}(w_j)\}$,

because there are exactly $|Q| \cdot |P| \cdot 2^{|Q|}$ different possibilities (r_i, p, S) for $r_i \in Q$, $p \in P$, and $S \subseteq Q$. Let $q'_1 = r_i$ and let $q'_2 \in Q$ such that $r'_{i+1} \to_{w_{i+1},w_j} q'_2$ and q'_2 occurs in $\mathrm{csts}(w_j)$. Such a q'_2 exists by the fact that r'_{i+1} contributes at w_{i+1} to v_n, using property (P4) (and also (P2) and (P3)). Since $r_{i+1} \to_{w_{i+1},w_j} r_i$, we can apply (P6) to get $q'_1 \to_{w_i,w_j} q'_1, q'_2$. Thus, conditions (1), (2), and (4) of input pumpability hold for $q_1 = q'_1$, $q_2 = q'_2$, $s_0 = t_n$, $u_0 = w_i$, and $u_0u_1 = w_j$. Clearly, if $q'_1 = q'_2$, then also (3) holds, which proves the lemma for that case. Thus, from now on we assume that $q'_1 \ne q'_2$. To realize (3), we will pump the tree $t_n/w_i[w'_j \leftarrow p]$ in t_n, where $w_j = w_iw'_j$.

For every $r \in Q$ that occurs in $\mathrm{csts}(w_i)$, there is an $r' \in Q$ with $r \to_{w_i,w_j} r'$, and r' occurs in $\mathrm{csts}(w_j)$ by (P4). Since the same states appear in $\mathrm{csts}(w_i)$ and $\mathrm{csts}(w_j)$, this means that r' also occurs in $\mathrm{csts}(w_i)$. Thus, there is a sequence

$$q'_1 \to_{w_i,w_j} q'_2 \to_{w_i,w_j} q'_3 \to_{w_i,w_j} \cdots \to_{w_i,w_j} q'_m \to_{w_i,w_j} q'_{m-\nu},$$

where $2 \le m \le |Q|$, $0 \le \nu < m$, and $q'_1, \ldots, q'_m$ are pairwise different states that occur in $\mathrm{csts}(w_i)$. Hence, after $m - \nu - 1$ steps of $\to_{w_i,w_j}$, starting at q'_1, states will repeat with period $\nu + 1$. Let d be a multiple of $\nu + 1$ with $d \ge m - \nu - 1$. Then there is a $\mu \in \{m - \nu, \ldots, m\}$ such that after d steps of $\to_{w_i,w_j}$, q'_1 reaches q'_μ and q'_μ reaches q'_μ.

Let $q_1 = q'_1$, $q_2 = q'_\mu$,

$$s_0 = (t_n[w_i \leftarrow p]) \bullet (t_n/w_i[w'_j \leftarrow p])^d \bullet (t_n/w_j),$$

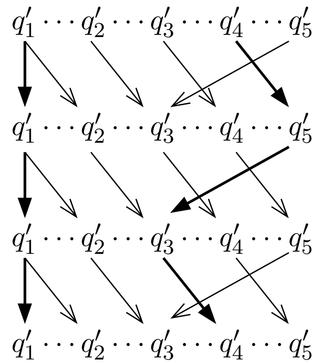

FIG. 6.2. *Conditions* (2) *and* (3) *of input pumpability for* $q_1 = q_1'$ *and* $q_2 = q_4'$.

$u_0 = w_i$, and $u_1 = (w_j')^d$. Then $h(s_0/w_i(w_j')^\gamma) = p$ for all $0 \le \gamma \le d$, which easily follows by induction, using the fact that $\hat{h}(t_n/w_i[w_j' \leftarrow p]) = \hat{h}(t_n/w_i[w_j' \leftarrow h(t_n/w_j)]) = h(t_n/w_i) = p$. In particular $h(s_0/u_0) = h(s_0/u_0u_1) = p$; i.e., condition (4) of input pumpability holds. Clearly, for $0 \le \gamma < d$, $q \rightarrow_{w_i,w_j} q'$ in the tree t_n if and only if $q \rightarrow_{w_i(w_j')^\gamma, w_i(w_j')^{\gamma+1}} q'$ in the tree s_0, and similarly $q \rightarrow_{w_i,w_j} q', q''$ in the tree t_n if and only if $q \rightarrow_{w_i(w_j')^\gamma, w_i(w_j')^{\gamma+1}} q', q''$ in the tree s_0; this is true because $s_0/w_i(w_j')^\gamma[w_j' \leftarrow p] = t_n/w_i[w_j' \leftarrow p]$. Thus, in s_0, $q_2 \rightarrow_{u_0,u_0u_1} q_2$ by the definition of q_μ' (using (P3)), which proves condition (3) of the input pumpable property. To show condition (2) we use (P6): Since $q_1' \rightarrow_{w_i,w_iw_j'} q_1', q_2'$, also $q_1' \rightarrow_{w_i,w_iw_j'} q_1'$ and thus, by the above and by (P3), $q_1' \rightarrow_{w_iw_j', w_i(w_j')^d} q_1'$ holds in s_0. By the definition of q_μ', $q_2' \rightarrow_{w_iw_j', w_i(w_j')^d} q_\mu'$. Therefore, by (P6), $q_1 \rightarrow_{u_0,u_0u_1} q_1, q_2$. Clearly, (1) of input pumpability holds because $q_0 \rightarrow_{\varepsilon,w_i} r_i$ in t_n by the definition of r_i, $s_0[u_0 \leftarrow p] = t_n[w_i \leftarrow p]$, and consequently $q_0 \rightarrow_{\varepsilon,u_0} r_i = q_1$ holds in s_0. Figure 6.2 outlines the choice of q_2 for $m = 5$ and $\nu = 2$ (thus $d = 3$ and $\mu = 4$). ∎

LEMMA 6.6. *Let M be a proper MTT^R. If M is input pumpable, then it is not lsi.*

Proof. Let $M = (Q, \Sigma, \Delta, q_0, R, P, h)$ be input pumpable; i.e., there are $q_1, q_2 \in Q$, $s_0 \in T_\Sigma$, $u_0 \in V(s_0)$, $u_1 \in V(s_0/u_0)$, and $p \in P$ such that (1)–(4) of Definition 6.4 hold. Assume now that M is lsi; i.e., there is a $c \in \mathbb{N}$ such that for every input tree $t \in T_\Sigma$,

$$(*) \qquad \mathrm{size}(\tau_M(t)) \le c \cdot \mathrm{size}(t).$$

In what follows we will derive a contradiction by constructing an input tree t such that $\mathrm{size}(\tau_M(t)) > c \cdot \mathrm{size}(t)$. Note first that if we replace in s_0 the subtree at u_0u_1 by any tree s in L_p, then (1)–(4) still hold. Similar to the proof of Theorem 6.3, the idea of constructing t is as follows: Consider input trees t_i obtained by i times pumping the tree $s_0/u_0[u_1 \leftarrow p]$ in the tree $s_0[u_0u_1 \leftarrow s]$. Then the trees t_i grow at most linearly with constant $\mathrm{size}(s_0/u_0[u_1 \leftarrow p])$. In the output tree $\tau_M(t_i)$ there are at least i occurrences of the tree $M_{q_2}(s)$. Hence, the trees $\tau_M(t_i)$ grow at least linearly with constant $\mathrm{size}(M_{q_2}(s))$. Thus, if we choose s in such a way that $\mathrm{size}(M_{q_2}(s))$ is larger than the product of c and the size of $s_0/u_0[u_1 \leftarrow p]$, then $\mathrm{size}(\tau_M(t_i))$ grows faster than $c \cdot \mathrm{size}(t_i)$; i.e., we can find an i such that $(*)$ does not hold for $t = t_i$.

In order to choose the tree s appropriately, we need the set $\mathrm{Out}(q_2, p) = \{M_{q_2}(s) \mid s \in L_p\}$ to be infinite, i.e., to contain trees with arbitrarily many output symbols.

This is guaranteed by i-properness (cf. point (i) of Definition 5.1) if (a) $\langle\!\langle q_2, p\rangle\!\rangle$ is reachable and (b) $q_2 \neq q_0$.

(a) Clearly, $\langle\!\langle q_2, p\rangle\!\rangle$ is reachable because it occurs in $\hat{M}_{q_0}(s_0[u_0u_1 \leftarrow p])$; this follows from (1) and (2) using Lemma 4.14(i) (analogous to the proof of (P3) in the proof of Lemma 6.5; in fact, using the $\rightarrow$ notation of the proof of that lemma, it follows from (1) and (2) by (P3) that $q_0 \rightarrow_{\varepsilon,u_0u_1} q_2$, which means that $\langle\!\langle q_2, p\rangle\!\rangle$ occurs in $\hat{M}_{q_0}(s_0[u_0u_1 \leftarrow p])$).

(b) By (2), $\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p]) \neq \langle\!\langle q_1, p\rangle\!\rangle = \hat{M}_{q_1}(s_0/u_0[\varepsilon \leftarrow p])$, and thus $u_1 \neq \varepsilon$; i.e., $u_1 = u_1'i$ for some $u_1' \in \mathbb{N}^*$ and $i \geq 1$. Also by (2), $\langle\!\langle q_2, p\rangle\!\rangle$ occurs in $\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p])$. Hence (by Lemma 4.3 applied to q_1, s_0/u_0, and u_1'), $\langle q_2, x_i\rangle$ occurs in the right-hand side of a rule of M. By (ii) of i-properness this implies that $q_2 \neq q_0$.

We now pump the tree $s_0/u_0[u_1 \leftarrow p]$ in the tree $s_0[u_0u_1 \leftarrow s] = (s_0[u_0 \leftarrow p]) \bullet (s_0/u_0[u_1 \leftarrow p]) \bullet s$: for $i \geq 0$, let $t_i = (s_0[u_0 \leftarrow p]) \bullet (s_0/u_0[u_1 \leftarrow p])^i \bullet s$. It follows from (1)–(4) that for every $i \geq 0$, $\mathrm{sts}_M(t_i, u_0u_1^i)$ contains at least one occurrence of q_1 and at least i occurrences of q_2; this is sketched in Figure 6.3 and formalized in the following claim.

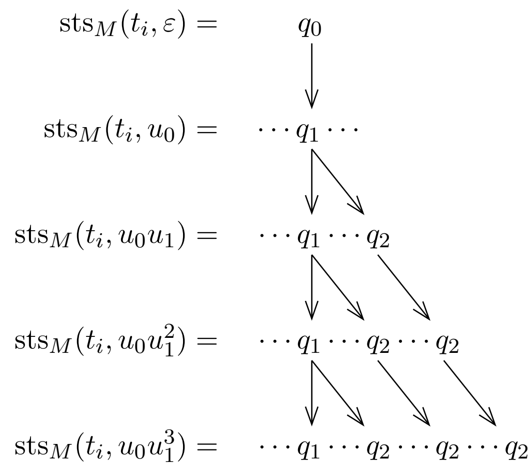

FIG. 6.3. *States that appear in state sequences of t_i.*

Claim. For all $i \geq 0$, $\#_{\langle\!\langle q_1,p\rangle\!\rangle}(\xi_i) \geq 1$ and $\#_{\langle\!\langle q_2,p\rangle\!\rangle}(\xi_i) \geq i$, where ξ_i is the tree $\hat{M}_{q_0}(t_i[u_0u_1^i \leftarrow p])$.

The proof of the claim is by induction on i. For $i = 0$, $t_i[u_0u_1^i \leftarrow p] = s_0[u_0 \leftarrow p]$ and by (1), $\#_{\langle\!\langle q_1,p\rangle\!\rangle}(\hat{M}_{q_0}(s_0[u_0 \leftarrow p])) \geq 1$. For $i+1$ we apply Lemma 4.14(i) to t_{i+1}, $u = \varepsilon$, $w = u_0u_1^i$, $v = u_0u_1^{i+1}$, and $q = q_0$. Since $h(t_{i+1}/u_0u_1^i) = h(s_0/u_0[u_1 \leftarrow s]) = \hat{h}(s_0/u_0[u_1 \leftarrow p]) = p$ by (4) and the fact that $s \in L_p$, $h(t_{i+1}/u_0u_1^{i+1}) = h(s) = p$, and $t_{i+1}[u_0u_1^i \leftarrow p] = t_i[u_0u_1^i \leftarrow p]$, we get

$$\#_{\langle\!\langle q',p\rangle\!\rangle}(\xi_{i+1}) \geq \sum_{r\in Q} \#_{\langle\!\langle q',p\rangle\!\rangle}(\hat{M}_r(s_0/u_0[u_1 \leftarrow p])) \cdot \#_{\langle\!\langle r,p\rangle\!\rangle}(\xi_i).$$

Let $q' = q_1$. Surely restricting the above sum to $r = q_1$ does not increase the result. Thus, the sum is $\geq \#_{\langle\!\langle q_1,p\rangle\!\rangle}(\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p])) \cdot \#_{\langle\!\langle q_1,p\rangle\!\rangle}(\xi_i)$. This is ≥ 1 because $\#_{\langle\!\langle q_1,p\rangle\!\rangle}(\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p])) \geq 1$ by (2), and $\#_{\langle\!\langle q_1,p\rangle\!\rangle}(\xi_i) \geq 1$ by induction.

Let $q' = q_2$. Now restrict the sum to $r \in \{q_1, q_2\}$. If $q_1 = q_2$, then the sum is $\geq \#_{\langle\langle q_2,p\rangle\rangle}(\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p])) \cdot \#_{\langle\langle q_1,p\rangle\rangle}(\xi_i)$; this is $\geq 2 \cdot \max\{1, i\} \geq i + 1$, because, by (2), $\#_{\langle\langle q_2,p\rangle\rangle}(\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p])) \geq 2$, and by induction $\#_{\langle\langle q_1,p\rangle\rangle}(\xi_i) = \#_{\langle\langle q_2,p\rangle\rangle}(\xi_i) \geq \max\{1, i\}$. If $q_1 \neq q_2$, then the sum is $\geq \#_{\langle\langle q_2,p\rangle\rangle}(\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p])) \cdot \#_{\langle\langle q_1,p\rangle\rangle}(\xi_i) + \#_{\langle\langle q_2,p\rangle\rangle}(\hat{M}_{q_2}(s_0/u_0[u_1 \leftarrow p])) \cdot \#_{\langle\langle q_2,p\rangle\rangle}(\xi_i)$; this is $\geq i+1$, because $\#_{\langle\langle q_2,p\rangle\rangle}(\hat{M}_{q_1}(s_0/u_0[u_1 \leftarrow p])) \geq 1$ by (2), $\#_{\langle\langle q_2,p\rangle\rangle}(\hat{M}_{q_2}(s_0/u_0[u_1 \leftarrow p])) \geq 1$ by (3), and, by induction, $\#_{\langle\langle q_1,p\rangle\rangle}(\xi_i) \geq 1$ and $\#_{\langle\langle q_2,p\rangle\rangle}(\xi_i) \geq i$. This ends the proof of the claim.

Since $\#_{\langle\langle q_2,p\rangle\rangle}(\xi_i) \geq i$, we obtain $\text{size}(\tau_M(t_i)) \geq i \cdot \#_\Delta(M_{q_2}(s))$ as follows. By Lemma 4.2 and the fact that $t_i/u_0u_1^i = s$, $\tau_M(t_i) = M_{q_0}(t_i) = \xi_i[\![\dots]\!]$ with $[\![\dots]\!] = [\![\langle\langle q,p\rangle\rangle \leftarrow M_q(s) \mid q \in Q]\!]$. By Lemma 2.6 (summing for all $\delta \in \Delta$), $\text{size}(\tau_M(t_i)) = \#_\Delta(\xi_i[\![\dots]\!]) = S_1^\Delta + S_2^\Delta \geq S_2^\Delta = \sum_{u \in V_{\langle\langle q,p\rangle\rangle}(\xi_i), q \in Q} \#_\Delta(M_q(s)) \cdot \prod F_{\xi_i,u}^{[\![\dots]\!]}$. Since M is nondeleting, it follows from Lemma 3.11(1) that $\#_{y_j}(M_q(s)) \geq 1$ for all $q \in Q^{(m)}$ and $j \in [m]$, and thus $\prod F_{\xi_i,u}^{[\![\dots]\!]} \geq 1$. We get $S_2^\Delta \geq \sum_{u \in V_{\langle\langle q,p\rangle\rangle}(\xi_i), q \in Q} \#_\Delta(M_q(s)) \geq \sum_{u \in V_{\langle\langle q_2,p\rangle\rangle}(\xi_i)} \#_\Delta(M_{q_2}(s)) \geq i \cdot \#_\Delta(M_{q_2}(s))$.

Now let $s \in L_p$ such that

$$\#_\Delta(M_{q_2}(s)) > c \cdot c_1,$$

where $c_1 = \text{size}(s_0/u_0[u_1 \leftarrow p]) - 1$. Then $\text{size}(\tau_M(t_i)) \geq i \cdot (cc_1 + 1) = icc_1 + i$. Let $i > c(c_0 + c_2)$, where $c_0 = \text{size}(s_0[u_0 \leftarrow p]) - 1$ and $c_2 = \text{size}(s)$. Since $\text{size}(t_i) = c_0 + ic_1 + c_2$ this means that $\text{size}(\tau_M(t_i)) > c \cdot \text{size}(t_i)$ because $\text{size}(\tau_M(t_i)) > icc_1 + c(c_0 + c_2) = c(c_0 + ic_1 + c_2) = c \cdot \text{size}(t_i)$. This contradicts $(*)$ and concludes the proof. ∎

We are now ready to prove step (III).

THEOREM 6.7. *Let M be a proper $MTT^R_{fnest,fcp}$. If M is lsi, then it is fci.*

Proof. If M is not fci, then, by Lemma 6.5, M is input pumpable and thus, by Lemma 6.6, M is not lsi. ∎

6.3. From lsi to fnest (I).

In Lemma 6.6 it was proved that if a proper MTT^R M is input pumpable, then it is not lsi. So, in order to prove that M is not lsi if it is not fnest, we would like to show that if M is not fnest, then it is input pumpable. This could be done by proving a pumping argument that works on the paths of trees $\hat{M}_{q_0}(s[u \leftarrow p])$. We have chosen the following alternative: We can associate with M a top-down tree transducer A (with the same regular look-ahead as M) in such a way that

(i) the number of elements $\langle\langle q', p\rangle\rangle$ of $\langle\langle Q, \{p\}\rangle\rangle$ that appear on a path of $\hat{M}_q(s[u \leftarrow p])$ is bounded by the number of such elements that appear in $\hat{A}_q(s[u \leftarrow p])$; and

(ii) if there are n occurrences of $\langle\langle q', p\rangle\rangle$ in $\hat{A}_q(s[u \leftarrow p])$, then there are at least n occurrences of $\langle\langle q', p\rangle\rangle$ in $\hat{M}_q(s[u \leftarrow p])$.

Thus, (i) implies that if M is not fnest, then A is not fci, and (ii) implies that if A is input pumpable, then so is M. Hence we need to show that if A is not fci, then A is input pumpable. This is exactly what the application of Lemma 6.5 to A gives. (The lemma is applicable because, obviously, every top-down tree transducer is nondeleting, fnest with nesting bound 1, and fcp.)

In order to prove (i) and (ii) we merely need to require that T^R A have the same states as M (but of rank zero) and that every rule of A have the same number of occurrences of each element of $\langle Q, X\rangle$ as the corresponding rule of M.

DEFINITION 6.8 (associated $\mathrm{T^R}$, globally fci). *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R. The T^R $A = (Q_A, P, \Sigma, \Delta, q_0, R_A, h)$ is* associated with M *if $Q_A = \{q^{(0)} \mid q \in Q\}$ and for every $q, q' \in Q$, $\sigma \in \Sigma^{(k)}$, $k \geq 0$, $i \in [k]$, and $p_1, \dots, p_k \in P$,*

$$\#_{\langle q', x_i\rangle}(\mathrm{rhs}_A(q, \sigma, \langle p_1, \dots, p_k\rangle)) = \#_{\langle q', x_i\rangle}(\mathrm{rhs}_M(q, \sigma, \langle p_1, \dots, p_k\rangle)).$$

The MTT^R M is globally fci *(for short, gfci) if every T^R associated with M is fci.*

We use the subscript "gfci" for classes of translations of $\mathrm{MTT^R}$s to denote that the corresponding transducers are gfci. Note that for $\mathrm{T^R}$s A_1 and A_2 associated with M, $\mathrm{sts}_{A_1}(s, u)$ is a permutation of $\mathrm{sts}_{A_2}(s, u)$ (cf. Lemma 6.9 of [19]). Hence, M is gfci if and only if there *exists* a $\mathrm{T^R_{fci}}$ associated with M. For every $\mathrm{MTT^R}$ M there is (effectively) an associated $\mathrm{T^R}$ A; it can be obtained from M by simply changing every right-hand side of M into an arbitrary right-hand side in $T_{\langle Q_A, X_k\rangle \cup \Delta}$ while preserving the number of occurrences of $\langle q, x_i\rangle$ for every $\langle q, x_i\rangle \in \langle Q, X_k\rangle$.

Let us first prove property (ii) mentioned above.

LEMMA 6.9. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be a nondeleting MTT^R and $A = (Q_A, P, \Sigma, \Delta, q_0, R_A, h)$ a T^R associated with M. For every $q, q' \in Q$, $s \in T_\Sigma$, $u \in V(s)$, and $p \in P$, $\#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{M}_q(s[u \leftarrow p])) \geq \#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{A}_q(s[u \leftarrow p]))$.*

Proof. The proof is by induction on the structure of s. Let $s = \sigma(s_1, \dots, s_k)$ with $\sigma \in \Sigma^{(k)}$ and $k \geq 0$. Let $m = \mathrm{rank}_Q(q)$.

If $u = \varepsilon$, then $\#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{M}_q(s[u \leftarrow p])) = \#_{\langle\!\langle q', p\rangle\!\rangle}(\langle\!\langle q, p\rangle\!\rangle(y_1, \dots, y_m))$, which equals (now with $q \in Q_A^{(0)}$) $\#_{\langle\!\langle q', p\rangle\!\rangle}(\langle\!\langle q, p\rangle\!\rangle) = \#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{A}_q(s[u \leftarrow p]))$.

Otherwise $u = iv$ with $i \in [k]$ and $v \in V(s_i)$. Thus $\hat{M}_q(s[u \leftarrow p])$ equals $\hat{M}_q(\sigma(\tilde{s}_1, \dots, \tilde{s}_k))$, where $\tilde{s}_\nu = s_\nu$ for $\nu \in [k] - \{i\}$ and $\tilde{s}_i = s_i[v \leftarrow p]$. For $\nu \in [k]$ let $p_\nu = \hat{h}(\tilde{s}_\nu)$. By Lemma 3.5, $\hat{M}_q(\sigma(\tilde{s}_1, \dots, \tilde{s}_k)) = t[\![\dots]\!]$, where $t = \mathrm{rhs}_M(q, \sigma, \langle p_1, \dots, p_k\rangle)$ and $[\![\dots]\!] = [\![\langle r, x_\nu\rangle \leftarrow \hat{M}_r(\tilde{s}_\nu) \mid \langle r, x_\nu\rangle \in \langle Q, X_k\rangle]\!]$. Applying Lemma 2.6 we obtain that $\#_{\langle\!\langle q', p\rangle\!\rangle}(t[\![\dots]\!])$ equals

$$\sum_{\substack{w \in V_{\langle r, x_\nu\rangle}(t),\\ \langle r, x_\nu\rangle \in \langle Q, X_k\rangle}} \#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{M}_r(\tilde{s}_\nu)) \cdot \prod F_{t,w}^{[\![\dots]\!]}.$$

Since M is nondeleting, by Lemma 3.11(1), $\#_{y_j}(\hat{M}_r(\tilde{s}_\nu)) \geq 1$ for all $r \in Q^{(n)}$, $j \in [n]$, and $\nu \in [k]$. This implies that $\prod F_{t,w}^{[\![\dots]\!]} \geq 1$. Hence,

$$(*) \qquad \#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{M}_q(s[u \leftarrow p])) \geq \sum_{\substack{w \in V_{\langle r, x_\nu\rangle}(t),\\ \langle r, x_\nu\rangle \in \langle Q, X_k\rangle}} \#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{M}_r(\tilde{s}_\nu)).$$

By induction, $\#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{M}_r(\tilde{s}_i)) \geq \#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{A}_r(\tilde{s}_i))$. For $\nu \in [k] - \{i\}$, $\tilde{s}_\nu \in T_\Sigma$, and therefore $\#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{M}_r(\tilde{s}_\nu)) = \#_{\langle\!\langle q', p\rangle\!\rangle}(M_r(\tilde{s}_\nu)) = 0 = \#_{\langle\!\langle q', p\rangle\!\rangle}(A_r(\tilde{s}_\nu)) = \#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{A}_r(\tilde{s}_\nu))$. Thus, the sum in $(*)$ is $\geq \sum_{w \in V_{\langle r, x_\nu\rangle}(t), \langle r, x_\nu\rangle \in \langle Q, X_k\rangle} \#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{A}_r(\tilde{s}_\nu))$. Since A is associated with M, $|V_{\langle r, x_\nu\rangle}(\zeta)| = |V_{\langle r, x_\nu\rangle}(t)|$ for every $\langle r, x_\nu\rangle \in \langle Q, X_k\rangle$, where $\zeta = \mathrm{rhs}_A(q, \sigma, \langle p_1, \dots, p_k\rangle)$. Therefore the above sum does not change if we replace t by ζ. Then by Lemma 2.4 we get $\#_{\langle\!\langle q', p\rangle\!\rangle}(\zeta[\dots])$ with $[\dots] = [\langle r, x_\nu\rangle \leftarrow \hat{A}_r(\tilde{s}_\nu) \mid \langle r, x_\nu\rangle \in \langle Q_A, X_k\rangle]$. By Lemma 3.5 and the fact that $\hat{A}$ is a $\mathrm{T^R}$, this equals $\#_{\langle\!\langle q', p\rangle\!\rangle}(\hat{A}_q(s[u \leftarrow p]))$. ∎

For a nondeleting $\mathrm{MTT^R}$ M it follows immediately from Lemma 6.9 and Definition 6.4 that if a $\mathrm{T^R}$ A associated with M is input pumpable, then M is also input pumpable.

LEMMA 6.10. *Let M be a nondeleting MTT^R and A a T^R associated with M. If A is input pumpable, then so is M.*

From Lemma 6.9 it also follows that gfci is a generalization of fci: if $\#_{\langle\langle Q,\{p\}\rangle\rangle}(\hat{M}_{q_0}(s[u \leftarrow p]))$ is bounded by some N, then so is $\#_{\langle\langle Q,\{p\}\rangle\rangle}(\hat{A}_{q_0}(s[u \leftarrow p]))$; i.e., if M is fci, then it is gfci. However, the converse is not true: there are $\mathrm{MTT^R}$s which are gfci but not fci. In fact, even for fcp $\mathrm{MTT^R}$s, gfci does *not* imply fci. To see this consider an MTT M which contains the following rules (and trivial look-ahead $P = \{p\}$):

$$\begin{array}{lll}\langle q_0, \sigma(x_1, x_2)\rangle & \to & \langle q, x_1\rangle(\langle q_0, x_2\rangle),\\ \langle q_0, \alpha\rangle & \to & \alpha,\\ \langle q, \sigma(x_1, x_2)\rangle(y_1) & \to & \sigma(y_1, y_1),\\ \langle q, \alpha\rangle(y_1) & \to & \sigma(y_1, y_1).\end{array}$$

Now let $s_0 = \alpha$ and for $n \geq 0$ let $s_{n+1} = \sigma(\alpha, s_n)$. Then

$$\begin{aligned}\langle q_0, s_n\rangle &\Rightarrow_M \langle q, \alpha\rangle(\langle q_0, s_{n-1}\rangle)\\ &\Rightarrow_M \sigma(\langle q_0, s_{n-1}\rangle, \langle q_0, s_{n-1}\rangle)\\ &\Rightarrow_M^* \sigma(\sigma(\langle q_0, s_{n-2}\rangle, \langle q_0, s_{n-2}\rangle), \sigma(\langle q_0, s_{n-2}\rangle, \langle q_0, s_{n-2}\rangle)).\end{aligned}$$

Hence, $\hat{M}_{q_0}(s_n[2^n \leftarrow p])$ is a full binary tree of height n with all leaves labeled $\langle\langle q_0, p\rangle\rangle$. Thus $\mathrm{sts}_M(s_n, 2^n) = q_0^{2^n}$, which means that M is not fci. However, M is gfci and fcp, with bounds 1 and 2, respectively. To see that M is gfci, consider the $\mathrm{T^R}$ A with right-hand side $\sigma(\langle q, x_1\rangle, \langle q_0, x_2\rangle)$ for the (q_0, σ)-rule and right-hand side α for all other rules. Now A is associated with M, and it is linear in the input variables x_i; i.e., A is fci with bound 1. Moreover, M is not lsi (because $\tau_M(s_n)$ is a full binary tree of height n). Thus, gfci plus fcp cannot be taken as an alternative to the definition of finite copying: $MTT^R_{\mathrm{fci,fcp}} \subsetneq MTT^R_{\mathrm{gfci,fcp}}$.

As illustrated by the example above, a gfci $\mathrm{MTT^R}$ M need not be fci, and thus the number of occurrences of elements of $\langle\langle Q, \{p\}\rangle\rangle$ in $\hat{M}_{q_0}(s[u \leftarrow p])$ is in general unbounded due to parameter copying (in the example above by the rules with right-hand side $\sigma(y_1, y_1)$). However, the number of such elements that appear on *one path* in $\hat{M}_{q_0}(s[u \leftarrow p])$ is bounded, and thus M is fnest. To see this intuitively, consider a label path π in a tree in $T_{\langle Q, T_\Sigma\rangle \cup \Delta}$. The application of a rule r of an $\mathrm{MTT^R}$ does not copy any states on the path π; thus, it increases the number of occurrences of q' on π by at most $\#_{\langle\{q'\},X\rangle}(\mathrm{rhs}(r))$, which equals $\#_{\langle\{q'\},X\rangle}(\mathrm{rhs}(r'))$ for the corresponding rule r' of a $\mathrm{T^R}$ associated with M. We now give a formal proof of property (i) mentioned above.

LEMMA 6.11. *Let $M = (Q, P, \Sigma, \Delta, q_0, R, h)$ be an MTT^R and $A = (Q_A, P, \Sigma, \Delta, q_0, R_A, h)$ a T^R associated with M. For every $q, q' \in Q$, $s \in T_\Sigma$, $u \in V(s)$, $p \in P$, and every label path π in $\hat{M}_q(s[u \leftarrow p])$, $\#_{\langle\langle q',p\rangle\rangle}(\pi) \leq \#_{\langle\langle q',p\rangle\rangle}(\hat{A}_q(s[u \leftarrow p]))$.*

Proof. The proof is by induction on the length of u.

For $u = \varepsilon$, $\#_{\langle\langle q',p\rangle\rangle}(\pi) = \#_{\langle\langle q',p\rangle\rangle}(\langle\langle q, p\rangle\rangle) = \#_{\langle\langle q',p\rangle\rangle}(\hat{A}_q(s[u \leftarrow p]))$.

For $u = u'i$ it follows from Lemma 4.3 that $\hat{M}_q(s[u \leftarrow p]) = t[\![i]\!][\![..]\!]$ with $t = \hat{M}_q(s[u' \leftarrow p'])[\![\mathrm{rhs}]\!]$, $p' = \hat{h}(s/u'[i \leftarrow p])$, and the substitutions $[\![\mathrm{rhs}]\!]$, $[\![..]\!]$, and $[\![i]\!]$ defined as in Lemma 4.3 (with u' instead of u, p' instead of p, and p instead of p_i). By Lemma 2.3(i) applied to $t'[\![..]\!]$ with $t' = t[\![i]\!]$, the label path π is of the form $w_0 v_1 w_1 \cdots v_m w_m$, $m \geq 0$, where $\pi' = w_0\langle r_1, x_{\nu_1}\rangle w_1 \cdots \langle r_m, x_{\nu_m}\rangle w_m$ is a label path in t', and for $j \in [m]$, $r_j \in Q$, $\nu_j \in [k] - \{i\}$, v_j is a label path in $M_{r_j}(s/u'\nu_j)$, and

$w_0, \dots, w_m$ do not contain elements of $\langle Q, X_k - \{x_i\}\rangle$. Since $M_{r_j}(s/u'\nu_j) \in T_\Delta(Y)$, $\#_{\langle\langle q',p\rangle\rangle}(v_j) = 0$ for all $j \in [m]$, which means that $\#_{\langle\langle q',p\rangle\rangle}(\pi) = \#_{\langle\langle q',p\rangle\rangle}(\pi')$.

Clearly, by the definition of $[\![i]\!]$, $\#_{\langle\langle q',p\rangle\rangle}(\pi') = \#_{\langle q',x_i\rangle}(\pi'')$ for some label path π'' in t. Hence, it remains to show that $\#_{\langle q',x_i\rangle}(\pi'') \le \#_{\langle\langle q',p\rangle\rangle}(\hat{A}_q(s[u \leftarrow p])) = \#_{\langle\langle q',p\rangle\rangle}(\xi[\mathrm{rhs}][..][i]) = \#_{\langle q',x_i\rangle}(\xi[\mathrm{rhs}])$, where $\xi = \hat{A}_q(s[u' \leftarrow p'])$ and [rhs], [..], $[i]$ are the (corresponding first-order variants of the) substitutions of Lemma 4.3.

By Lemma 2.3(i) applied to the tree $t = \hat{M}_q(s[u' \leftarrow p'])[\![\mathrm{rhs}]\!]$, π'' is of the form $w_0v_1w_1 \cdots v_mw_m$, $m \ge 0$, where $\rho = w_0\langle\langle r_1, p'\rangle\rangle w_1 \cdots \langle\langle r_m, p'\rangle\rangle w_m$ is a label path in $\hat{M}_q(s[u' \leftarrow p'])$ and for $j \in [m]$, $r_j \in Q$, v_j is a label path in $\mathrm{rhs}_M(r_j, \sigma, \langle p_1, \dots, p_k\rangle)$, and $w_0, \dots, w_m$ contain no elements of $\langle\langle Q, \{p'\}\rangle\rangle$ (i.e., w_j is a string over $\Delta \cup Y$). Thus, $\#_{\langle q',x_i\rangle}(\pi'') = \sum_{j\in[m]} \#_{\langle q',x_i\rangle}(v_j)$. Since, for $j \in [m]$, v_j is a label path in $\mathrm{rhs}_M(r_j, \sigma, \langle p_1, \dots, p_k\rangle)$, this sum is surely

$$\le \sum_{j\in[m]} \#_{\langle q',x_i\rangle}(\mathrm{rhs}_M(r_j, \sigma, \langle p_1, \dots, p_k\rangle)) = \sum_{j\in[m]} \#_{\langle q',x_i\rangle}(\mathrm{rhs}_A(r_j, \sigma, \langle p_1, \dots, p_k\rangle)),$$

which can be written as

$$\sum_{r\in Q} \#_{\langle\langle r,p'\rangle\rangle}(\rho) \cdot \#_{\langle q',x_i\rangle}(\mathrm{rhs}_A(r, \sigma, \langle p_1, \dots, p_k\rangle)).$$

By induction this is $\le \sum_{r\in Q} \#_{\langle\langle r,p'\rangle\rangle}(\xi) \cdot \#_{\langle q',x_i\rangle}(\mathrm{rhs}_A(r, \sigma, \langle p_1, \dots, p_k\rangle))$, which equals $\#_{\langle q',x_i\rangle}(\xi[\mathrm{rhs}])$ by Lemma 2.4. ∎

It follows immediately from Lemma 6.11, by taking $q = q_0$ and summing over all $q' \in Q$, that if A is fci, then M is fnest, with the same bound. This is stated in the next lemma.

LEMMA 6.12. *If an MTT*$^{\mathrm{R}}$ *is gfci, then it is fnest.*

We are now ready to prove step (I), i.e., that for a proper MTT$^{\mathrm{R}}$, lsi implies fnest.

THEOREM 6.13. *Let M be a proper MTT*$^{\mathrm{R}}$*. If M is lsi, then it is fnest.*

Proof. If M is not fnest, then by Lemma 6.12 it is not gfci. By the definition of gfci this means that any T$^{\mathrm{R}}$ A associated with M is not fci. The application of Lemma 6.5 to A gives that A is input pumpable, and thus by Lemma 6.10 M is input pumpable. Now Lemma 6.6 implies that M is not lsi. ∎

From Theorems 6.13, 6.3, and 6.7 we obtain the main result of this section: the converse of Theorem 4.19 for proper MTT$^{\mathrm{R}}$s.

THEOREM 6.14. *Let M be a proper MTT*$^{\mathrm{R}}$*. If M is lsi, then it is finite copying.*

Recall from section 4.3 the notion of finite contribution. By Lemma 4.18, every finite copying MTT$^{\mathrm{R}}$ is of finite contribution, and by the discussion before Theorem 4.19, every MTT$^{\mathrm{R}}$ of finite contribution is lsi. Together with Theorem 6.14 this shows that a proper MTT$^{\mathrm{R}}$ is finite copying if and only if it is of finite contribution. It can be proved that this even holds for a productive MTT$^{\mathrm{R}}$ that satisfies (ii) of Definition 5.6 (of p-properness). Thus, the notions of finite copying and finite contribution are closely related.

7. Main results and consequences. In this final section we prove our main results: (i) a translation is MSO definable if and only if it is a macro tree translation of linear size increase, and (ii) for a given MTT M it is decidable whether or not τ_M is MSO definable. Then we discuss some consequences of these results for top-down tree transducers, attributed tree transducers, and context-free graph grammars. At last some open problems and further research topics are mentioned.

THEOREM 7.1. *Let M be an MTT^R. Then the following statements are equivalent:*

(1) τ_M *is MSO definable.*

(2) τ_M *is lsi.*

(3) *prop*(M) *is finite copying.*

Proof. Since every MSO definable tree translation is lsi (see section 2.5), (1) $\Rightarrow$ (2). Note that this can also be proved using the results from section 4: If τ_M is MSO definable, then by Lemma 4.9, $\tau_M \in MTT^R_{\text{fc}}$ and thus, by Theorem 4.19, τ_M is lsi. To show (2) $\Rightarrow$ (3), let τ_M be lsi. By Theorem 5.9, there is a proper MTT^{R} prop(M) with $\tau_{\text{prop}(M)} = \tau_M$; i.e., $\tau_{\text{prop}(M)}$ is lsi. By Theorem 6.14, prop(M) is finite copying. Finally, if prop(M) is finite copying, then, by Lemma 4.9, $\tau_M = \tau_{\text{prop}(M)}$ is MSO definable. Thus (3) $\Rightarrow$ (1). ∎

Note that, as discussed at the end of section 6, we could have included "(4) prop(M) is of finite contribution" as another equivalent statement in Theorem 7.1.

Theorem 7.1 shows that the class *MSOTT* of MSO definable tree translations can be characterized as those macro tree translations that are lsi. Recall (from section 2.5) that *LSI* denotes the class of all lsi tree translations.

THEOREM 7.2. $MSOTT = MTT \cap LSI$.

Proof. If $\tau \in MTT \cap LSI$, then there is an MTT M such that $\tau_M = \tau$ is lsi. By Theorem 7.1 τ_M is MSO definable, and thus $MTT \cap LSI \subseteq MSOTT$. If $\tau \in MSOTT$, then by Lemma 4.9 there is an MTT^{R} M with $\tau_M = \tau$. By Theorem 7.1 τ_M is lsi, and thus $MSOTT \subseteq MTT^R \cap LSI$. By Lemma 3.4, $MTT^R = MTT$. ∎

By Theorem 7.1, the proper normal form prop(M) (which can be constructed by Theorem 5.9) of an MTT M is finite copying if and only if τ_M is MSO definable. Since the finite copying property is decidable (Lemma 4.10) this implies that for M it is decidable whether or not τ_M is MSO definable. If prop(M) is finite copying, then an MSO tree transducer that realizes τ_M can be constructed, because the equality $MSOTT = MTT^R_{\text{fc}}$ of Lemma 4.9 is effective (cf. the discussion following Lemma 4.10).

THEOREM 7.3. *It is decidable for an MTT M whether or not τ_M is MSO definable, and if it is, then an MSO tree transducer for τ_M can be constructed.*

7.1. Top-down tree transducers. A top-down tree transducer can translate a monadic tree (of height n) into a full binary tree (of height n). This translation is of exponential size increase, and hence it is not MSO definable. On the other hand, there are MSO definable tree translations that cannot be realized by top-down tree transducers: consider the translation that associates with a tree its yield (i.e., the left-to-right sequence of the labels of its leaves), seen as a monadic tree. This translation is MSO definable (cf. Example 1(6, yield) of [3]) but it cannot be realized by a top-down tree transducer, because it is of exponential height increase (viz. it translates a full binary tree of height n into its yield, a monadic tree of height 2^n), whereas top-down tree translations are of linear height increase (cf. Lemma 3.27 of [28]). Now, which translations realized by top-down tree transducers (with regular look-ahead) are MSO definable? By our results, they are exactly the translations realized by finite copying T^{R}s.

THEOREM 7.4. $T^R \cap MSOTT = T^R_{fc}$.

Proof. Let M be a T^{R} such that τ_M is MSO definable. By Theorem 7.1, prop(M) is finite copying. By Theorem 5.9, prop(M) is a T^{R}. Thus, $\tau_M = \tau_{\text{prop}(M)} \in T^R_{\text{fc}}$. Hence, $T^R \cap MSOTT \subseteq T^R_{\text{fc}}$. The inclusion $T^R_{\text{fc}} \subseteq T^R \cap MSOTT$ is immediate from Lemma 4.9. ∎

It is shown in Theorem 7.4 of [19] that $T^R_{\text{fc}} = MSOTT_{\text{dir}}$: the so-called direction preserving MSO definable tree translations. An MSO tree transducer (see section 2.5) is direction preserving if $s \models \chi_{i,c,d}(x, y)$ implies that y is a descendant of x in the input tree s. Thus, the descendant relation in the output tree can only hold between nodes that are also (copies of) descendants in the input tree.

Note that it follows immediately from Theorem 7.1 that $T^R \cap MSOTT = T^R \cap LSI$. Thus, $T^R_{\text{fc}} = T^R \cap LSI$. Since $\mathrm{T}^{\mathrm{R}}_{\mathrm{fc}}$s are closely related to tree-walking transducers (see Theorem 4.9 of [22]), this may be viewed as the result of [1] that the translations realized by tree-walking transducers are exactly the generalized syntax-directed translations of linear size increase.

7.2. Attributed tree transducers. Attributed tree transducers [27, 28] serve as a formal model for attribute grammars [37]. As argued in [3], adding the feature of look-ahead to them yields a better model of attribute grammars and a more robust class of tree translations. Let ATT^R denote the class of translations realized by attributed tree transducers with look-ahead (see [3, 19]) and let the subscript "sur" denote that the transducers are "single-use restricted" (cf. section 5 in [19]); i.e., for every input symbol σ, each outside attribute is used at most once in the set of rules for σ. It is proved in Theorem 17 of [3] that $MSOTT = ATT^R_{\text{sur}}$. Hence $MSOTT \subseteq ATT^R \cap LSI$. Equality of these classes now follows from Theorem 7.2 and the fact that $ATT^R \subseteq MTT$. (The latter inclusion can be proved as follows: By definition, ATT^R consists of all translations that can be realized by the composition of an attributed relabeling, followed by an attributed tree translation. It follows from Theorem 4.4 of [19] that attributed relabelings can be realized by T^{R}s. Thus, $ATT^R \subseteq T^R \circ ATT$, where ATT denotes the class of translations realized by attributed tree transducers. By Lemma 5.11 of [19], $ATT \subseteq MTT^R$ and so $T^R \circ ATT \subseteq T^R \circ MTT^R$, which, by Lemma 3.4, equals $T^R \circ MTT$. Since regular look-ahead can be realized by first running a finite state relabeling, i.e., applying a translation in $DBQREL$ (cf. Theorem 2.6 of [14]), we get the inclusion in $DBQREL \circ T \circ MTT$, which is $\subseteq DBQREL \circ MTT$ by Corollary 4.10 of [24], and thus we have the inclusion in $MTT^R = MTT$.)

THEOREM 7.5. $MSOTT = ATT^R \cap LSI$.

From the fact that $ATT^R \subseteq MTT$ (effectively) together with Theorem 7.3 and the fact that $MSOTT = ATT^R_{\text{sur}}$ (effectively), we obtain the following decidability result for attributed tree transducers.

THEOREM 7.6. *For an ATT^R A it is decidable whether or not there exists an equivalent single-use restricted ATT^R A', and if so, A' can be constructed.*

The interpretation of Theorem 7.6 in terms of classical attribute grammars involves a technical detail: roughly speaking, the look-ahead part of an $\mathrm{ATT}^{\mathrm{R}}$ corresponds to the underlying context-free grammar of an attribute grammar. If we want to apply Theorem 7.6 to an attribute grammar G, then we first have to turn G into an equivalent $\mathrm{ATT}^{\mathrm{R}}$ A, i.e., into an $\mathrm{ATT}^{\mathrm{R}}$ that realizes the same tree-to-tree translation as G (translating the non-derivation-trees of G into some error symbol). Now assume that for A there is an equivalent single-use restricted $\mathrm{ATT}^{\mathrm{R}}$ A'. In general the look-ahead of A' will be different from the one of A, which implies that an attribute grammar G' equivalent to A' does *not* have the same underlying context-free grammar as G, and hence the tree-to-tree translation realized by G' is different from the one realized by G. This problem can be avoided by adding boolean-valued attributes to G' (cf. the introduction of [3]), which simulate the look-ahead part of A'. In this way G' and G have the same underlying context-free grammar and they realize the same tree-to-tree translation (however, the boolean-valued attributes are, in general, not single-use restricted).

7.3. Context-free graph grammars. A context-free graph grammar (see, e.g., [16]) generates a graph language. If the graphs are restricted to trees, then we obtain a tree language. As discussed in the introduction of [19], the class of tree languages that can be generated by context-free graph grammars (either by hyperedge replacement (HR), or by node replacement (NR); cf. Section 6 of [16]) can be obtained by applying the MSO definable tree translations to the regular tree languages. By Theorem 7.2 it means that this class of tree languages can be obtained by the application of lsi macro tree translations to the regular tree languages. This is just a straightforward variation of similar statements in the literature: for single-use restricted ATTs in Corollary 19 of [3], for "single-use restricted" MTTs and for finite copying MTTs in Corollary 7.3 of [19], and for nondeleting MTTs that are finite copying and linear in the parameters in Theorem 5 of [20] (based on Theorem 8.1 of [10]).

THEOREM 7.7. *The output tree languages of MTTs of linear size increase applied to the regular tree languages are the tree languages generated by (HR or NR) context-free graph grammars.*

7.4. Open problems and further research topics. We have proved that for an MTT it is decidable whether or not the translation it realizes is MSO definable. What is the complexity of this problem? In fact, the complexity of deciding the finiteness of ranges of (compositions of) MTTs [12] (cf. Lemma 3.8) is not known, and our decidability proof is based on this result. Generally speaking, complexity issues have not yet been studied in the area of MTTs. A first result in that direction is [42], which shows that for a composition τ of deterministic macro tree translations and an input tree s, the corresponding output tree $\tau(s)$ can be computed in time linear in the sum of the sizes of s and t.

Is it decidable for an MSO tree transducer whether its translation is in T^R, i.e., whether it can be simulated by a top-down tree transducer with regular look-ahead (see section 7.1)?

It would be interesting to find a classification of the possible size increases of MTTs. For top-down tree transducers such a classification is given in [1], and it is shown that the size increase of every top-down tree transducer is either polynomial or exponential. For MTTs it could be the case that every size increase is either polynomial, exponential, or double exponential.

Is polynomial size increase decidable for MTTs? If so, what is the complexity? For top-down tree transducers it is shown in [11] that this problem is NLOGSPACE-complete. It is not clear how MSO definability could be generalized in order to obtain the class of polynomial size increase macro tree translations. (Note that there are well-established models of polynomial size increase transducers based on first-order logic; see, e.g., [13, 34].)

Composition of MTTs yields a proper hierarchy; i.e., there are translations which can be realized by the composition of $m+1$ MTTs but not by the composition of m MTTs (Theorem 4.16 of [24]). Now, what happens if we restrict our attention to translations that are of linear size increase? Maybe then composition does *not* yield a proper hierarchy, but rather it remains the class of MSO definable tree translations, i.e., is $LSI \cap \bigcup_n MTT^n = MSOTT$. Since compositions of MTTs can be realized by high-level tree transducers (and vice versa) [25] this question is equivalent to the following: Are lsi high-level tree translations MSO definable? Again, this question could also be considered for polynomial instead of linear size increase.

Recently we have shown that the k-pebble tree transducer, introduced in [45] as a formal model for XML query languages, can be simulated by the composition of MTTs [21]. Thus, if, as discussed in the previous paragraph, our results would even hold for compositions of MTTs, then they would also hold for k-pebble tree transducers.

For both MTTs and MSO transducers there are nondeterministic variants (cf. [24] and [5], respectively). We would like to know whether our result carries over to the nondeterministic case, i.e., whether the nondeterministic macro tree translations of linear size increase are precisely the nondeterministic MSO definable tree translations.

Last but not least: Given an MTT M, is it decidable whether the translation τ_M realized by M can be realized by an attributed tree transducer (with look-ahead), i.e., is it decidable whether $\tau_M \in ATT$ (or ATT^R)? This question is of interest because, as shown in [3], ATT^R is the class of tree translations that are MSO definable (by an MSO tree-to-graph transducer; see [5]) when subtrees can be shared in the output tree. Of course, if τ_M is MSO definable (without sharing of subtrees), which can be decided by Theorem 7.3, then the answer to the above question is positive, because $MSOTT = ATT^R_{\text{sur}}$ by the result of [3] (other positive criteria are discussed in [8, 9, 29]). On the other hand, note that ATT^{R}s are of linear size-to-height increase (cf., e.g., Lemma 5.40 of [28]). Denote by *LSHI* the class of all translations of linear size-to-height increase. Probably it can be proved (by methods similar to those in this paper) that $MTT \cap LSHI = MTT^R_{\text{fnest}}$ and that $\tau_M \in LSHI$ if and only if $\text{prop}(M)$ is fnest, which is decidable. Thus, it would be decidable for an MTT whether or not it is of linear size-to-height increase. If it is not, then it cannot be realized by an ATT^{R}. But these are only partial answers.

Acknowledgment. We thank the referees for their constructive comments.

REFERENCES

[1] A. V. Aho and J. D. Ullman, *Translations on a context-free grammar*, Inform. and Control, 19 (1971), pp. 439–475.
[2] G. J. Bex, S. Maneth, and F. Neven, *A formal model for an expressive fragment of XSLT*, Inform. Systems, 27 (2002), pp. 21–39.
[3] R. Bloem and J. Engelfriet, *A comparison of tree transductions defined by monadic second order logic and by attribute grammars*, J. Comput. System Sci., 61 (2000), pp. 1–50.
[4] B. Courcelle, *Fundamental properties of infinite trees*, Theoret. Comput. Sci., 25 (1983), pp. 95–169.
[5] B. Courcelle, *Monadic second-order definable graph transductions: A survey*, Theoret. Comput. Sci., 126 (1994), pp. 53–75.
[6] B. Courcelle, *Structural properties of context-free sets of graphs generated by vertex replacement*, Inform. and Comput., 116 (1995), pp. 275–293.
[7] B. Courcelle and J. Engelfriet, *A logical characterization of the sets of hypergraphs defined by hyperedge replacement grammars*, Math. Systems Theory, 28 (1995), pp. 515–552.
[8] B. Courcelle and P. Franchi-Zannettacci, *Attribute grammars and recursive program schemes.* I, Theoret. Comput. Sci., 17 (1982), pp. 163–191.
[9] B. Courcelle and P. Franchi-Zannettacci, *Attribute grammars and recursive program schemes.* II, Theoret. Comput. Sci., 17 (1982), pp. 235–257.
[10] F. Drewes, *A characterization of the sets of hypertrees generated by hyperedge-replacement graph grammars*, Theory Comput. Systems, 32 (1999), pp. 159–208.
[11] F. Drewes, *The complexity of the exponential output size problem for top-down and bottom-up tree transducers*, Inform. and Comput., 169 (2001), pp. 264–283.
[12] F. Drewes and J. Engelfriet, *Decidability of finiteness of ranges of tree transductions*, Inform. and Comput., 145 (1998), pp. 1–50.
[13] H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer-Verlag, Berlin, 1995.

[14] J. Engelfriet, *Top-down tree transducers with regular look-ahead*, Math. Systems Theory, 10 (1977), pp. 289–303.
[15] J. Engelfriet, *Some open questions and recent results on tree transducers and tree languages*, in Formal Language Theory: Perspectives and Open Problems, R. Book, ed., Academic Press, New York, 1980, pp. 241–286.
[16] J. Engelfriet, *Context-free graph grammars*, in Handbook of Formal Languages, Volume 3, G. Rozenberg and A. Salomaa, eds., Springer-Verlag, Berlin, 1997, pp. 125–213.
[17] J. Engelfriet and G. Filè, *The formal power of one-visit attribute grammars*, Acta Inform., 16 (1981), pp. 275–302.
[18] J. Engelfriet and H. J. Hoogeboom, *MSO definable string transductions and two-way finite-state transducers*, ACM Trans. Comput. Log., 2 (2001), pp. 216–254.
[19] J. Engelfriet and S. Maneth, *Macro tree transducers, attribute grammars, and MSO definable tree translations*, Inform. and Comput., 154 (1999), pp. 34–91.
[20] J. Engelfriet and S. Maneth, *Tree languages generated by context-free graph grammars*, in Proceedings of the 6th International Workshop on Theory and Application of Graph Transformations (TAGT'98), H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, eds., Lecture Notes in Comput. Sci. 1764, Springer-Verlag, New York, 2000, pp. 15–29.
[21] J. Engelfriet and S. Maneth, *A comparison of pebble tree transducers with macro tree transducers*, Acta Inform., to appear.
[22] J. Engelfriet, G. Rozenberg, and G. Slutzki, *Tree transducers, L systems, and two-way machines*, J. Comput. System Sci., 20 (1980), pp. 150–202.
[23] J. Engelfriet and V. van Oostrom, *Logical description of context-free graph languages*, J. Comput. System Sci., 55 (1997), pp. 489–503.
[24] J. Engelfriet and H. Vogler, *Macro tree transducers*, J. Comput. System Sci., 31 (1985), pp. 71–146.
[25] J. Engelfriet and H. Vogler, *High level tree transducers and iterated pushdown tree transducers*, Acta Inform., 26 (1988), pp. 131–192.
[26] J. Engelfriet and H. Vogler, *The translation power of top-down tree-to-graph transducers*, J. Comput. System Sci., 49 (1994), pp. 258–305.
[27] Z. Fülöp, *On attributed tree transducers*, Acta Cybernet., 5 (1981), pp. 261–279.
[28] Z. Fülöp and H. Vogler, *Syntax-Directed Semantics—Formal Models based on Tree Transducers*, Monogr. Theoret. Comput. Sci. EATCS Ser., W. Brauer, G. Rozenberg, and A. Salomaa, eds., Springer-Verlag, Berlin, 1998.
[29] Z. Fülöp and H. Vogler, *A characterization of attributed tree transformations by a subclass of macro tree transducers*, Theory Comput. Systems, 32 (1999), pp. 649–676.
[30] H. Ganzinger, *Increasing modularity and language-independency in automatically generated compilers*, Sci. Comput. Programming, 3 (1983), pp. 223–278.
[31] F. Gécseg and M. Steinby, *Tree Automata*, Akadémiai Kiadó, Budapest, 1984.
[32] F. Gécseg and M. Steinby, *Tree automata*, in Handbook of Formal Languages, Volume 3, G. Rozenberg and A. Salomaa, eds., Springer-Verlag, Berlin, 1997, pp. 1–68.
[33] R. Giegerich, *Composition and evaluation of attribute coupled grammars*, Acta Inform., 25 (1988), pp. 355–423.
[34] N. Immerman, *Descriptive Complexity*, Springer-Verlag, New York, 1999.
[35] E. Irons, *A syntax directed compiler for ALGOL* 60, Comm. ACM, 4 (1961), pp. 51–55.
[36] N. Klarlund and M. I. Schwartzbach, *Graphs and decidable transductions based on edge constraints*, in Proceedings of the 19th Colloquium on Trees in Algebra and Programming—CAAP 94, S. Tison, ed., Lecture Notes in Comput. Sci. 787, Springer-Verlag, New York, 1994, pp. 187–201.
[37] D. Knuth, *Semantics of context-free languages*, Math. Systems Theory, 2 (1968), pp. 127–145. (Corrections in Math. Systems Theory, 5 (1971), pp. 95–96.)
[38] H.-P. Kolb, J. Michaelis, U. Mönnich, and F. Morawietz, *An operational and denotational approach to non-context-freeness*, Theoret. Comput. Sci., to appear.
[39] A. Kühnemann, *Benefits of tree transducers for optimizing functional programs*, in Proceedings of the 18th Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'98), V. Arvind and R. Ramanujam, eds., Lecture Notes in Comput. Sci. 1530, Springer-Verlag, New York, 1998, pp. 146–157.
[40] A. Kühnemann and H. Vogler, *Attributgrammatiken*, Vieweg-Verlag, Braunschweig, Wiesbaden, Germany, 1997.
[41] A. Kühnemann and J. Voigtländer, *Tree Transducer Composition as Deforestation Method for Functional Programs*, Tech. Report TUD-FI01-07, Department of Computer Science, Technical University of Dresden, 2001.
[42] S. Maneth, *The complexity of compositions of deterministic tree transducers*, in Proceedings

of the 22nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2002), M. Agrawal and A. Seth, eds., Lecture Notes in Comput. Sci. 2556, Springer-Verlag, New York, 2002, pp. 265–276.

[43] S. Maneth and F. Neven, *Recursive structured document transformations*, in Research Issues in Structured and Semistructured Database Programming—Revised Papers DBPL'99, R. Connor and A. Mendelzon, eds., Lecture Notes in Comput. Sci. 1949, Springer-Verlag, New York, 2000, pp. 80–98.

[44] J. Michaelis, U. Mönnich, and F. Morawietz, *On minimalist attribute grammars and macro tree transducers*, in Linguistic Form and Its Computation, C. Rohrer, A. Rossdeutscher, and H. Kamp, eds., CSLI Publications, Stanford, CA, 2001, pp. 287–326.

[45] T. Milo, D. Suciu, and V. Vianu, *Typechecking for XML transformers*, in Proceedings of the 19th ACM Symposium on Principles of Database Systems (PODS'2000), ACM, New York, 2000, pp. 11–22.

[46] W. Rounds, *Mappings and grammars on trees*, Math. Systems Theory, 4 (1970), pp. 257–287.

[47] J. Stoy, *Denotational Semantics*, MIT Press, Cambridge, MA, 1977.

[48] J. Thatcher, *Generalized^2 sequential machine maps*, J. Comput. System Sci., 4 (1970), pp. 339–367.

[49] A. van Deursen, J. Heering, and P. Klint, eds., *Language Prototyping*, AMAST Series in Computing 5, World Scientific, River Edge, NJ, 1996.

[50] A. van Deursen, P. Klint, and F. Tip, *Origin tracking and its applications*, in Language Prototyping, World Scientific, River Edge, NJ, 1996, pp. 249–294.

[51] V. Vianu, *A web odyssey: From Codd to XML*, in Proceedings of the 20th ACM Symposium on Principles of Database Systems (PODS'2001), ACM, New York, 2001, pp. 1–15.

[52] H. Vogler, *Functional description of the contextual analysis in block-structured programming languages: A case study of tree transducers*, Sci. Comput. Programming, 16 (1991), pp. 251–275.

[53] R. Wilhelm and D. Maurer, *Compiler Design*, Addison-Wesley, Reading, MA, 1995.