

Acknowledgments

First of all, my heartfelt thanks go to my supervisor Professor Zoltán Fülöp. He encouraged me to apply for a Ph.D. scholarship and introduced me to the beauty of the theory of transducers. I also owe him many valuable suggestions and comments, and much from them.

I am grateful the Hungarian Research and Development in Informatics for giving me a fellowship after my Ph.D. studies. I also thank the Institute of Informatics for providing me excellent working conditions.

I would like to thank Professor Róbert Veszteg for letting me to work at the Dresden University of Technology for four weeks. He provided me a pleasant atmosphere to work, most part of the Thesis was written down during that stay. Also, many thanks for his valuable suggestions concerning the Thesis.

Thanks to my colleagues, Balázs Veszteg, Péter Tóth, and Tamás Vinkó, who are also my friends, for the nice conversations about life, try (and not so big) questions of life. Many thanks to Loránd Stankov and Boglárka Tóth for their valuable comments on a preliminary version of the Thesis.

I am very thankful to my parents for their love and support during my studies. Without their encouragement I could not have done this work, which therefore I dedicate to them.

Finally, my special thanks go to my wife and three kids for love, tolerance, and understanding.

Supervisor: Prof. Zoltán Fülöp

University of Szeged
Department of Foundations of
Computer Science

2006



Acknowledgments

First of all, my heartfelt thanks go to my supervisor Professor Zoltán Fülöp. He encouraged me to apply for a PhD studentship and introduced me to the beauty of the theory of tree transducers. I am grateful for his guidance, inspiration, and many valuable suggestions. I am also thankful for the common works, I could learn so much from them.

I am grateful the Foundation for Teaching and Research in Informatics for giving me a fellowship after my PhD student years, and the Institute of Informatics for providing me excellent working conditions.

I would like to thank Professor Heiko Vogler inviting me to work at the Dresden University of Technology for four month in 2005. He provided me a pleasant atmosphere to work, most part of the Thesis was written down during this stay. Also, many thanks for his valuable suggestions concerning the Thesis.

Thanks to my colleagues, Balázs Szörényi, Boglárka Tóth, and Tamás Vinkó, who are also my friends, for the nice conversations about the big (and not so big) questions of life. Many thanks to Lóránd Muzamel and Boglárka Tóth for their valuable comments on a preliminary version of the Thesis.

I am very thankful to my parents for their love and support during my studies. Without their encouragement I could not have done this work, which therefore I dedicate to them.

Finally, my special thank goes to my girlfriend Bogi for her love, tolerance, and understanding.

Contents

1	Introduction	1
1.1	Syntax Directed Semantics	1
1.2	Top-Down and Bottom-Up Tree Transducers	5
1.3	Shape Preserving Tree Transducers	8
1.4	Characterizing Semantical Properties by the Syntax	9
1.5	The Outline of the Thesis	13
2	Preliminaries	15
2.1	Definitions, Notations, and Preliminary Results	15
2.2	Two Examples of Shape Preserving Tree Transducers	22
3	Characterizing of Shape Preserving Tree Transducers	25
3.1	Useful Properties of Shape Preserving Tree Transducers	25
3.1.1	The Top-Down Case	25
3.1.2	The Bottom-Up Case	29
3.2	Transforming Tree Transducers	45
3.2.1	Transforming Shape Preserving Top-Down Tree Transducers into Equivalent Quasirelabelings	45
3.2.2	Constructing Frame Transducers of Transformable Bottom-Up Tree Transducers	52
3.2.3	Properties of Frame Transducers	68
3.3	Characterizing the Shape Preserving Property	77
3.3.1	The Top-Down Case	77
3.3.2	The Bottom-Up Case	82
4	Decidability Results	87
4.1	Decidability of the Shape Preserving Property	87
4.1.1	The Top-Down Case	87
4.1.2	The Bottom-Up Case	91
4.2	Decidability of the Equivalence	104

5 Conclusions	107
Summary	109
Összefoglalás (Summary in Hungarian)	113
Bibliography	117
Glossary	123
Index	125

Chapter 1

Introduction

Tree transducers are formal models of program compilers. Indeed, their theory was significantly motivated by the, so-called, syntax-directed compilation (or syntax-directed translation). To demonstrate this, we give a short description of the syntax-directed translation in the first section. In Section 1.2 we describe two particular tree transducers, namely top-down tree transducers and bottom-up tree transducers, and discuss their behavior because these transducers will be in the focus of our interest. In the next section we introduce the shape preserving versions of the above transducers. Moreover, we point out an important difference between the syntax of shape preserving top-down tree transducers and shape preserving bottom-up tree transducers. Namely, we describe the reason of that in the bottom-up case the transducers can be deleting, however in the top-down case can not.

In Section 1.4 we present the main problem solved in the Thesis which consists of the following questions. Can we characterize shape preserving top-down and bottom-up tree transformations by restrictions on the syntax of these transducers? Is the shape preserving property of these transducers decidable? In the Thesis we will give positive answers to these questions. The last section describes the outline of the Thesis.

1.1 Syntax Directed Semantics

Compilers are computer programs which transform an input program written in a higher level programming language (also called the source language) into an output program of another language, called the target or object language [AU77, Joh79, ASU86]. The target language can usually be understood and executed by the computer.

The earliest program compilers were large hand-written programs which contained the definition (i.e., the syntax) of the source language deeply embedded in the code of the programs. This method made the modification of the source language very

difficult, since it was not obvious how to find those parts of the compiler which might be affected by the modifications. Moreover, it was also difficult to check whether the compiler accepts all those programs which have correct syntax, i.e., which correspond to the definition of the source language, and that it accepts only those programs. Nevertheless, all compilers before 1960 were of this type, since there was not a common method for defining source languages.

In 1956 the linguist Noam Chomsky invented context-free grammars [Cho56]. His intention was to give a formal model which can describe natural languages. It turned out that these grammars are powerful enough to specify the syntax of most programming languages. In fact, the syntax of most programming languages is defined by a notational variant of context-free grammars, called Backus Naur Form (BNF), which was invented, independently from Chomsky, by the computer scientist John W. Backus in 1959 [Bac59]. Soon after this, Edgar T. Irons proposed a new method to define a compiling system [Iro61] and he called his method as syntax-directed compilation. In fact he suggested to separate the definition of the source language and the translation of input programs from the source language into output programs of the target language. That part of compilers which checks whether the input program has the correct syntax according to the definition of the source language is called parsing. This part, as we have mentioned above, can be done using context-free grammars. Moreover, since context free grammars are weak enough, efficient algorithms can be given for the parsing. The second part, i.e., the translation of the input program into the output program, is in fact the semantics of the compilation. By the method of Irons, the translation of the input program is given according to the definition of the source language, i.e., the semantics is defined in terms of the syntax. This is the idea of the, so-called, syntax-directed semantics.

Let us consider now the parsing of an input program. If the program is syntactically correct, then the parsing yields a tree, called the derivation tree (or parsing tree) of the program, which shows the structure of it according to the definition of the programming language (cf. Figure 1.1). A derivation tree is a finite, directed, ordered, acyclic, labeled graph such that each node has at most one incoming edge, and there is exactly one node with no incoming edges, which is called the root node (or just root). Clearly, every node has zero or more, so-called, child nodes. Moreover, every node different from the root has exactly one parent node.

After the parsing, the compiler works on the parsing tree (or input tree), and yields a program in the output language. Nevertheless, the translation of the input program can be done by transforming the input tree into another tree, called the output tree, which represents the structure of the output program in the target language. Then the output program is simply the yield of the output tree. Therefore, from theoretical

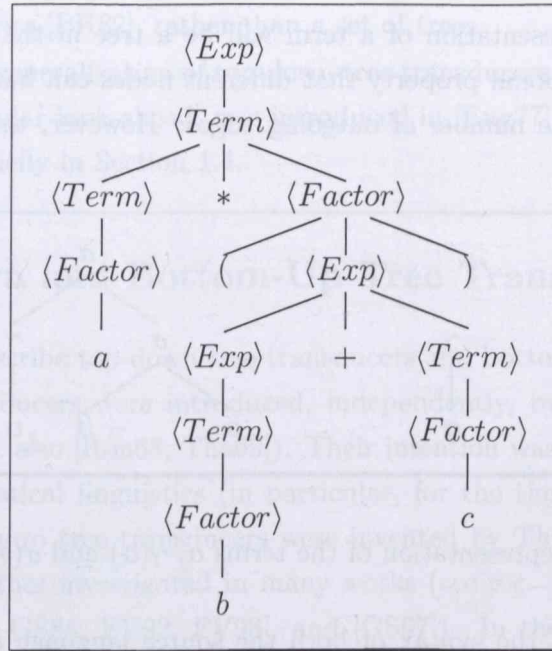


Figure 1.1: The parse tree of the arithmetic expression $a * (b + c)$.

point of view, the translation of the input program is nothing else but a transformation of the input tree. In this way the compiler transforms a parsing tree into an other tree, i.e., computes a function which associates an output tree with every input tree (cf. Figure 1.2).

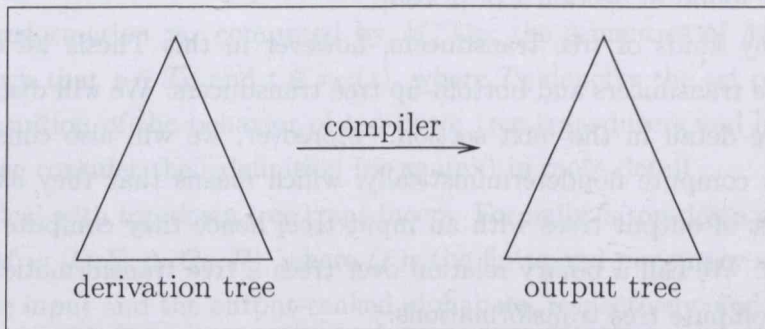


Figure 1.2: A compiler transforms a derivation tree into an output tree.

Since we want to study abstract mathematical properties of the syntax-directed translation, we make the following abstraction steps. First of all, by a tree we mean a well-formed term over a ranked alphabet. A ranked alphabet is a finite nonempty set of symbols such that every symbol in it has a rank (or arity), which is a nonnegative integer. For instance, $\Sigma = \{\alpha^{(0)}, \beta^{(0)}, \gamma^{(1)}, \sigma^{(2)}\}$ is a ranked alphabet, where the upper index of a symbol means its rank, and α , $\gamma(\alpha)$, and $\sigma(\sigma(\alpha, \beta), \gamma(\alpha))$ are trees over Σ .

Note that the graph representation of a term will be a tree in the above sense, see Figure 1.3, with the additional property that different nodes can have the same label only if they have the same number of outgoing edges. However, this latter is just a technical matter.

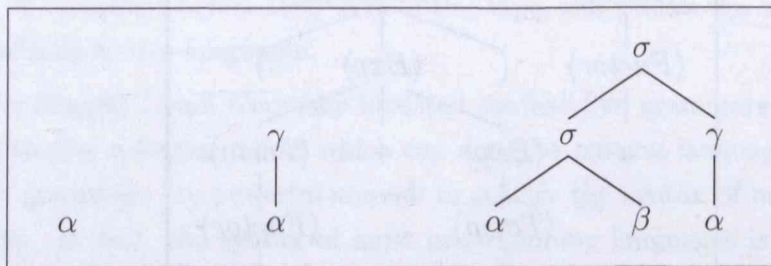


Figure 1.3: The tree representation of the terms α , $\gamma(\alpha)$ and $\sigma(\sigma(\alpha, \beta), \gamma(\alpha))$.

Now we abstract from the syntax of both the source language and of the target language and we allow trees over ranked alphabets to be the input tree and the output tree. Moreover, we replace the machine-oriented computation paradigm of the compiler with a more abstract computation paradigm which is called term rewriting. As a result of these big abstractions steps, we obtain an abstract computation device which takes an input tree over a ranked alphabet, then, by manipulating on the input tree as a term rewrite system, computes an output tree over another ranked alphabet. Such an abstract device is called a tree transducer. A more detailed description of this abstraction can be found in Section 1 of [FV98].

There are many kinds of tree transducers, however in this Thesis we deal only with top-down tree transducers and bottom-up tree transducers. We will discuss them informally in more detail in the next section. Moreover, we will also consider tree transducers which compute nondeterministically, which means that they associate a (maybe empty) set of output trees with an input tree, hence they compute a binary relation over trees. We call a binary relation over trees a tree transformation. Hence tree transducers compute tree transformations.

We note that top-down and bottom up tree transducers can model only certain basic types of syntax-directed semantics. However, there are tree transducers with greater expressive power (e.g. attributed tree transducers [Fül81] which are formal models of Knuth's attribute grammars [Knu68], and macro tree transducers [EV85]) which can model syntax-directed semantics in general (cf. also [FV98]). Moreover, we note also that many generalizations of the above transducers have been introduced, for example, modular tree transducers [EV91], which are generalizations of macro tree transducers; top-down tree-to-graph transducers [EV94]; higher-order attributed tree transducers [NV00]; tree series transducers [Kui99, EFV02, ÉK03, FV03] which transform a tree

into a formal tree series [BR82], rather than a set of trees.

A rather natural generalization of top-down tree transducers, namely top-down tree transducers with regular look-ahead, was introduced in [Eng77]. We will discuss these latter transducers briefly in Section 1.4.

1.2 Top-Down and Bottom-Up Tree Transducers

In this section we describe top-down tree transducers and bottom-up tree transducers. Top-down tree transducers were introduced, independently, by Rounds [Rou70] and Thatcher [Tha70] (cf. also [Rou68, Tha69]). Their intention was to give formal models for parts of mathematical linguistics (in particular, for the theory of syntax-directed translation). Bottom-up tree transducers were invented by Thatcher [Tha73]. These transducers were further investigated in many works (see e.g. [Eng75, Bak78, Bak79, Ési80, Eng82, Ési83, GS84, FV92, FV98], and [GS97]). In the Thesis we follow the terminology of [Eng75]. The reader can find a deep introduction to the theory of top-down and bottom-up tree transducers in [GS84, GS97], where these transducers are called root-to-frontier and frontier-to-root tree transducers, respectively.

A tree transducer M is a finite state machine which processes a tree s , called input tree, over the input ranked alphabet Σ and computes a finite set $\tau_M(s)$ of trees, called output trees, over the output ranked alphabet Δ . Top-down tree transducers process an input tree s from its root towards its leaves. Bottom-up tree transducers work in the opposite direction, i.e., they process an input tree s from its leaves towards its root. The tree transformation τ_M computed by M , i.e., the semantics of M , is the set of pairs (s, t) such that $s \in T_\Sigma$ and $t \in \tau_M(s)$, where T_Σ denotes the set of trees over Σ . To give an intuition of the behavior of top-down tree transducers and bottom-up tree transducers we consider their definition (or syntax) in more detail.

First we deal with top-down tree transducers. Formally, a top-down tree transducer is a system $M = (Q, \Sigma, \Delta, Q_0, R)$, where Q is the finite and nonempty set of states; Σ and Δ are the input and the output ranked alphabets, respectively, and $Q_0 \subseteq Q$ is the set of initial states. States are considered to be unary symbols. Moreover, R is a finite set of (rewriting) rules of the form $q(\sigma(x_1, \dots, x_k)) \rightarrow r$, where $q \in Q$, σ is an input symbol of rank k from Σ , x_1, \dots, x_k are variables, and r is a tree over Δ which may contain also subtrees as $p(x_i)$, where p is a state and $1 \leq i \leq k$. Variables are zero-ary symbols.

In fact M is a term rewrite system which rewrites an input tree starting at the root of the tree and proceeding towards its leafs. In more detail, a general step of the rewriting by M can be described as follows. Assume that u is an intermediate tree over $Q \cup \Sigma \cup \Delta$ such that above the states in u there are only symbols from Δ while below

them only symbols from Σ . Assume also that there is a subtree $q(\sigma(u_1, \dots, u_k))$ of u , where $q \in Q$, $k \geq 0$, $\sigma \in \Sigma$ with rank k , and $u_1, \dots, u_k \in T_\Sigma$ (see Figure 1.4, where the gray and white parts of u are consisting of symbols from Δ and Σ respectively, and in-between the two parts states are from Q). If there is a rule in R of the form $q(\sigma(x_1, \dots, x_k)) \rightarrow r$, then M rewrites u into another tree v , which fact we denote by $u \Rightarrow_M v$, where v is the tree obtained by replacing the subtree $q(\sigma(u_1, \dots, u_k))$ of u by $r[u_1, \dots, u_k]$. Here $r[u_1, \dots, u_k]$ is the tree yielded by replacing the variables x_1, \dots, x_k in r by the trees u_1, \dots, u_k , respectively.

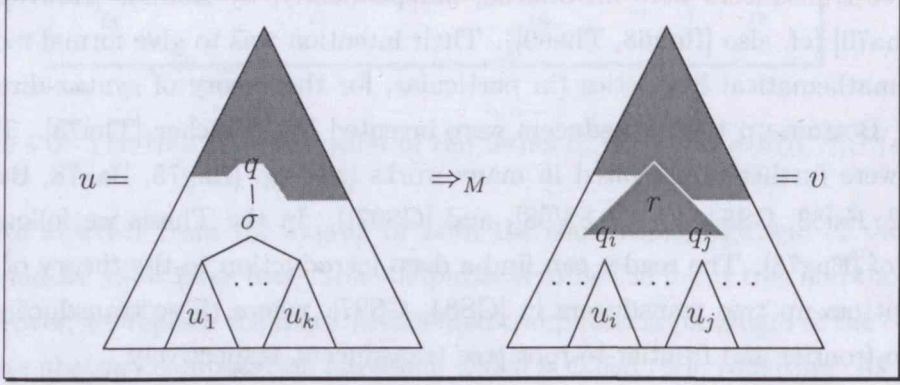


Figure 1.4: Application of a rewriting rule of a top-down tree transducer.

If r does not contain a variable x_i for some $1 \leq i \leq k$, then the tree u_i will not show up in the tree v . In this case we say that M deletes u_i .

Now take an input tree $s \in T_\Sigma$. Applying rules in R to the tree $q_0(s)$ as far as possible, where $q_0 \in Q_0$ is an initial state, the rewriting process may terminate in an output tree $t \in T_\Delta$. If this is the case, then we write $q_0(s) \Rightarrow_M^* t$. In this way M computes the tree transformation $\tau_{M,q_0} = \{(s, t) \in T_\Sigma \times T_\Delta \mid q_0(s) \Rightarrow_M^* t\}$ in state q_0 . Then the tree transformation τ_M computed by M is the relation $\cup_{q \in Q_0} \tau_{M,q}$. A tree transformation computed by a top-down tree transducer is called a top-down tree transformation.

Now we describe the behavior of bottom-up tree transducers. A bottom-up tree transducer is a system $M = (Q, \Sigma, \Delta, Q_0, R)$, where Q , Σ and Δ are the same as in the top-down case, and Q_0 is the set of final states. Moreover, R is a finite set of rules of the form $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$, where $q, q_1, \dots, q_k \in Q$, σ is an input symbol of rank k from Σ , and r is a tree over Δ which may contain also variables as its leaves from the set $\{x_1, \dots, x_k\}$.

Bottom-up tree transducers also can be regarded as term rewrite systems. However, M starts to rewrite an input tree at the leaves and proceeds towards the root of the tree. A general step of a rewriting process of an input tree can be described as follows.

Assume that u is an intermediate tree over $Q \cup \Sigma \cup \Delta$ such that above the states in u there are only symbols from Σ while below them only symbols from Δ . Assume also that there is a subtree $\sigma(q_1(v_1), \dots, q_k(v_k))$ of u , where $k \geq 0$, $\sigma \in \Sigma$ with rank k , $q_1, \dots, q_k \in Q$, and $v_1, \dots, v_k \in T_\Delta$ (see Figure 1.5, where the gray and white parts of u are consisting of symbols from Δ and Σ respectively, and between the two parts states are from Q). If M has a rule of the form $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$, then M can apply it to rewrite u into a tree v , which we denote by $u \Rightarrow_M v$. In fact, M replaces the subtree $\sigma(q_1(v_1), \dots, q_k(v_k))$ of u by $q(r[v_1, \dots, v_k])$, where $r[v_1, \dots, v_k]$ is the tree resulted by replacing the variables x_1, \dots, x_k in r to the trees v_1, \dots, v_k , respectively.

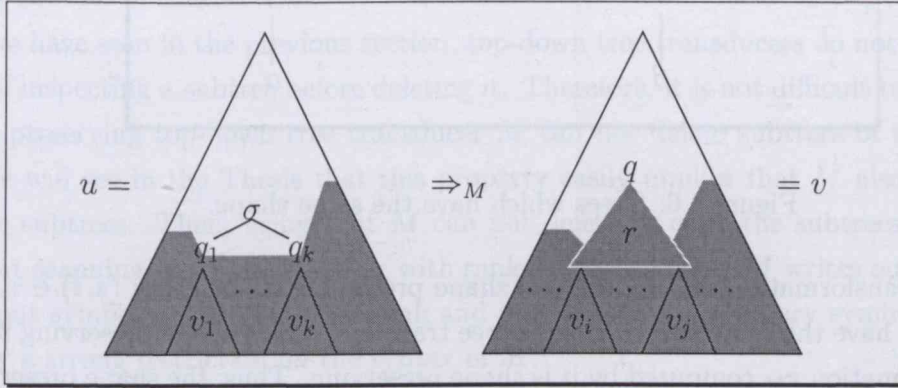


Figure 1.5: Application of a rewriting rule of a bottom-up tree transducer.

Again, as in the top-down case, r may not contain a variable x_i , i.e., M deletes the tree u_i . However, in contrast to the top-down case, before the deletion M rewrites u_i to a tree $q_i(v_i)$.

Applying rewriting steps as far as possible, M may rewrite an input tree $s \in T_\Sigma$ into a tree $q_0(t)$, where $q_0 \in Q_0$ and $t \in T_\Delta$. If this is the case, then we write $s \Rightarrow_M^* q_0(t)$. Similarly as in the top-down case, M computes the tree transformation $\tau_{M,q_0} = \{(s, t) \in T_\Sigma \times T_\Delta \mid s \Rightarrow_M^* q_0(t)\}$ in state q_0 . Then the tree transformation τ_M computed by M is $\cup_{q \in Q_0} \tau_{M,q}$. The concept of a bottom-up tree transformation is defined analogously with the top-down one.

In [Eng75] it was pointed out that there are important differences between the behavior of top-down and bottom-up tree transducers. One of them concerns the problem of deleting. In fact, as we have seen above, a top-down tree transducer can not inspect a subtree before deleting it, while a bottom-up tree transducer can. A consequence of this difference is that there are bottom-up tree transformations which can not be computed by any top-down tree transducer. (Note that the analogous statement, which we obtain by exchanging bottom-up and top-down, also holds because of other differences.) In the next section we discuss how the above difference in deleting affects the syntax of shape preserving top-down and bottom-up tree transducers.

1.3 Shape Preserving Tree Transducers

We say that two trees have the same shape, if they are isomorphic modulo labeling, i.e., they differ only in the labels of their nodes (cf. Figure 1.6).

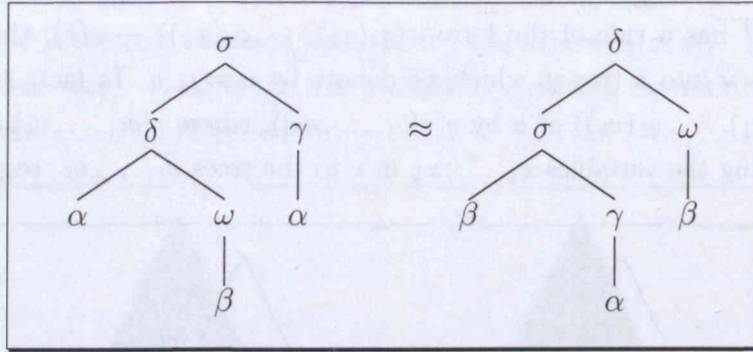


Figure 1.6: Trees which have the same shape.

A tree transformation $\tau \subseteq T_\Sigma \times T_\Delta$ is shape preserving if, for every $(s, t) \in \tau$, the trees s and t have the same shape. Now, a tree transducer M is shape preserving if the tree transformation τ_M computed by it is shape preserving. Thus, the shape preserving property of a tree transducer M is a property of the tree transformation τ_M computed by M , hence it is a dynamic property in the following sense. Assume, that M is a shape preserving (top-down or bottom-up) tree transducer and consider an input tree s of M . Assume also that s has a subtree s' and during the rewriting process of s into an output tree t , s' is rewritten into a tree t' . Then it can be very well that the shapes of s' and t' are not the same, however, since M is shape preserving, s and t have the same shape.

We can give examples of shape preserving tree transducers easily. For this, we recall the concept of *finite state relabelings* from [Eng75]. However, in this Thesis, for the sake of brevity, we drop the attribute “finite state” from the name of finite state relabelings, and write just *relabeling* instead. (We note that in [Eng75] the name “relabeling” refers to a simpler class of transducers.) A top-down tree transducer is a top-down relabeling if each of its rules has the form $q(\sigma(x_1, \dots, x_k)) \rightarrow \delta(q_1(x_1), \dots, q_k(x_k))$, where σ and δ are input and output symbols, respectively, of the same rank k . Similarly, a bottom-up tree transducer is a bottom-up relabeling if each of its rules has the form $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(\delta(x_1, \dots, x_k))$. It was shown in [Eng75] (cf. the discussion after Definition 3.14) that both bottom-up and top-down relabeling tree transducers compute the same tree transformation class, which we denote by *QREL*.

Now it can be seen that both top-down and bottom-up relabeling tree transducers are shape preserving. This is because, in both the top-down and the bottom-up case,

a statical property, the syntax, assures that the dynamic property holds, i.e., that the shape of the input tree and the output tree are the same. In fact, scanning an input symbol $\sigma \in \Sigma$, a relabeling tree transducer writes out exactly one output symbol $\delta \in \Delta$ such that σ and δ have the same rank. Moreover, both tree transducers keep the order of the subtrees symbolized by the variables x_1, \dots, x_k .

On the other hand, it is easy to give top-down and bottom-up shape preserving tree transducers which are not relabeling tree transducers. Nevertheless, the shape preserving property causes a rather strong restriction on the syntax of top-down tree transducers, while in the bottom-up case this does not hold. We discuss this in more detail in the rest of this section.

As we have seen in the previous section, top-down tree transducers do not have the ability of inspecting a subtree before deleting it. Therefore, it is not difficult to see that a shape preserving top-down tree transducer M can not delete subtrees of the input tree. We will see in the Thesis that this property easily implies that M also can not copy the subtrees. Then, using that M can not delete or copy the subtrees, we will show that scanning an input symbol σ with rank different to one, M writes out exactly one output symbol δ with the same rank and maybe some more unary symbols. This is clearly a strong restriction on the syntax of M .

Now let $M = (Q, \Sigma, \Delta, Q_0, R)$ be a shape preserving bottom-up tree transducer and let $s \in T_\Sigma$ such that $s \Rightarrow_M^* q_0(t)$, where $q_0 \in Q_0$ and $t \in T_\Delta$. Assume that there is a subtree $\sigma(s_1, \dots, s_k)$ of s where $k \geq 1$ and $\sigma \in \Sigma$ with rank k . Moreover, assume that s_i , for some $1 \leq i \leq k$ has been rewritten into the term $q_i(t_i)$, where $q_i \in Q$ and $t_i \in T_\Delta$. During the rewriting process of s_i , M can also collect information about it and can store this information in the state q_i . For example, this information can be the shape of s_i . Therefore, even if M deletes the subtree s_i , it can reproduce its shape in the output tree using the information stored in the state q_i . This clearly implies that a shape preserving bottom-up tree transducer can delete, and therefore also copy, subtrees of the input tree.

1.4 Characterizing Semantical Properties by the Syntax

In this Section first we describe a problem class concerning tree transducers as follows. Take a tree transformation class C computed by tree transducers of a certain kind. Moreover, let us consider a (natural) subclass C' of C . Now we ask whether a (natural) syntactic restriction of the model can be given such that a tree transformation τ belongs to C' if and only if there is a tree transducer M which obeys that syntactic restriction and for which $\tau_M = \tau$. If such a syntactic restriction can be given, then we have characterized a semantic restriction made on C by a syntactic restriction made on the

tree transducer model. Moreover, we ask whether the following problem is decidable: given a tree transducer M of that certain kind, is the tree transformation τ_M in C' ?

In the rest of the section we give three “positive” instances of this problem class: for each instance, the semantic restriction can be characterized by a syntactic restriction and the elementship problem is decidable. Finally we present the main problem which we have solved in the thesis as a fourth positive instance of that problem class.

The first example concerns top-down tree transducers with regular look-ahead. In order to present it, we need some preparation.

A subset L of T_Σ , for a ranked alphabet Σ , is called a tree language. Tree automata [Don65, Don70] and [TW65, TW68] are finite state devices which recognize tree languages (cf. also [GS84, GS97]). If a tree language L is recognizable by a tree automaton, then we say that L is a recognizable tree language.

A top-down tree transducer with regular look-ahead [Eng77] is a five-tuple $M = (Q, \Sigma, \Delta, Q_0, R)$, where Q, Σ, Δ and Q_0 are the same as for top-down tree transducers, and R is a finite set of the rules of the form $\langle q(\sigma(x_1, \dots, x_k)) \rightarrow r, L_1, \dots, L_k \rangle$, where $q(\sigma(x_1, \dots, x_k)) \rightarrow r$ is a top-down tree transducer rule, and L_1, \dots, L_k , the look-ahead, are recognizable tree languages. The above rule can be used to rewrite a subtree $q(\sigma(s_1, \dots, s_k))$ of the input tree only if the inclusion $s_i \in L_i$ holds for every $1 \leq i \leq k$. In this way the transducer has the ability of inspecting a subtree of a node before deleting it (note that bottom-up tree transducers also have this property, cf. the discussion at the end of Section 1.2).

We say that M is deterministic, if Q_0 is a singleton, and if $\langle q(\sigma(x_1, \dots, x_k)) \rightarrow r_1, L_1, \dots, L_k \rangle$ and $\langle q(\sigma(x_1, \dots, x_k)) \rightarrow r_2, L'_1, \dots, L'_k \rangle$ are two different rules in R with the same left-hand side, then there is an $1 \leq i \leq k$, such that $L_i \cap L'_i = \emptyset$. It is clear that “being deterministic” is a restriction on the syntax of the transducer and that deterministic top-down tree transducers with regular look-ahead compute tree transformations which are partial mappings.

Now we consider the first example. Let us denote the class TOP^R of tree transformations which can be computed by top-down tree transducers with regular look-ahead. Let TOP_f^R be the subclass of TOP^R which consists of transformations which are partial mappings. (As we have mentioned, a deterministic top-down tree transducer with regular look-ahead computes a tree transformation which is in TOP_f^R .) On the other hand, it was shown in [Eng78] (cf. Theorem on page 171) that a tree transformation τ is in TOP_f^R if and only if it can be computed by a deterministic top-down tree transducer with regular look-ahead. Therefore a restriction on the semantics (i.e, on the set TOP^R) is characterized by a restriction on the syntax.

Moreover, it is decidable if a tree transformation computed by a top-down tree transducer with regular look-ahead is in TOP_f^R or not. This can be seen as follows.

Let M be a top-down tree transducer with regular look-ahead. Then, by Theorem 2.6 of [Eng77], $\tau_M = \tau_{M_1} \circ \tau_{M_2}$, where M_1 is a deterministic bottom-up relabeling tree transducer, i.e M_1 does not have different rules with the same left-hand side, M_2 is a top-down tree transducer, and \circ is the composition of the relations τ_{M_1} and τ_{M_2} . Now, since τ_{M_1} is a partial mapping (note that M_1 is deterministic), τ_M is a partial mapping if and only if the restriction of τ_{M_2} to the tree language $\tau_{M_1}(\text{dom}(\tau_M))$, where $\text{dom}(\tau_M)$ denotes the domain of τ_M , is a partial mapping. By Corollary 2.7 of [Eng77], $\tau_{M_1}(\text{dom}(\tau_M))$ is a recognizable language. Then, by Theorem 8 of [Ési80], it is decidable if the restriction of τ_{M_2} to the tree language $\tau_{M_1}(\text{dom}(\tau_M))$ is a partial mapping or not. This implies that it is also decidable whether τ_M is a partial mapping.

Now we prepare our second example. A deterministic top-down tree transducer $M = (Q, \Sigma, \Delta, Q_0, R)$ with regular look-ahead has the single-use restriction if the following holds. Let σ be an input symbol in Σ with rank $k \geq 0$ and let $\mu_1 = \langle q_1(\sigma(x_1, \dots, x_k)) \rightarrow r_1, L_1, \dots, L_k \rangle$ and $\mu_2 = \langle q_2(\sigma(x_1, \dots, x_k)) \rightarrow r_2, L_1, \dots, L_k \rangle$ be rules in R . Then, for every term $q(x_i)$ where $q \in Q$ and $1 \leq i \leq k$, if $q(x_i)$ occurs in both r_1 and r_2 , then $\mu_1 = \mu_2$. This means that for this symbol σ and look-ahead L_1, \dots, L_k , no $q(x_i)$ occurs more than once in the right-hand sides of those rules which scan σ and have the look-ahead L_1, \dots, L_k . Clearly, the single-use restriction is a restriction on the syntax of the model.

On the other hand, a tree transformation τ has linear size increase if there is a constant c such that, for every $(s, t) \in \tau$, $\text{size}(t)$ is bounded by $c \cdot \text{size}(s)$ (by the size of a tree we mean the number of its nodes).

The second, more recent example is the following. Let $TOP_{f, \text{lin}}^R$ denote the subclass of TOP_f^R consisting of tree transformations which have linear size increase. It was shown that $TOP_{f, \text{lin}}^R$ is exactly the set of tree transformation which can be computed by deterministic top-down tree transducers with regular look-ahead and single-use restriction (see Theorems 7.4 and 7.2 in [EM03] and Theorem 7.4 in [EM99]). Therefore, again, a semantic restriction on the tree transformation class is characterized by a syntactic restriction on the model itself. Moreover, by Theorem 2 of [Man03], it is decidable, whether a given deterministic top-down tree transducer with regular look-ahead computes a tree transformation of linear size increase or not.

The third example concerns recognizable tree languages and bottom-up tree transducers. We say that a bottom-up tree transducer $M = (Q, \Sigma, \Delta, Q_0, R)$ is linear if, for every rule $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R and $1 \leq i \leq k$, x_i occurs at most once in r . Clearly, being linear is a restriction on the syntax of tree transducers. Moreover, a tree transformation $\tau \subseteq T_\Sigma \times T_\Delta$ preserves recognizability if, for every recognizable tree language $L \subseteq T_\Sigma$, the tree language $\tau(L) = \cup_{s \in L} \tau(s)$ is also recognizable. Let BOT denote the class of bottom-up tree transformations and BOT_{pr} denote the subclass of

BOT consisting of tree transformations which preserve recognizability.

By Corollary 3.11 of [Eng75], linear bottom-up tree transformations preserve recognizability. On the other hand, by Theorem 1 of [Géc81], a bottom up tree transformation preserves recognizability, if and only if it can be computed by a linear bottom-up tree transducer. By these results, again, a semantical property is characterized by a restriction on the syntax of the transducers. Moreover, by Theorem 2 of [Géc81], it is decidable whether an arbitrary bottom-up tree transducer computes a tree transformation which is in BOT_{pr} .

In the Thesis we show a result of similar type. In fact, we characterize shape preserving top-down and bottom-up tree transformations by relabeling tree transducers. Let *SHAPE* be the set of the tree transformations computed by shape preserving (top-down or bottom-up) tree transducers. Certainly, $SHAPE \subseteq TOP \cup BOT$, where *TOP* is the class of top-down tree transformations. We have seen that a relabeling tree transducer computes a shape preserving tree transformation, thus $QREL \subseteq SHAPE$. As the main results of the Thesis, we will show that every top-down tree transformation $\tau \in SHAPE$ can be computed by a relabeling tree transducer (Theorem 3.64), and that every bottom-up tree transformation $\tau \in SHAPE$ can also be computed by a relabeling tree transducer (Theorem 3.70). Thus a top-down or bottom-up tree transformation τ is in *SHAPE* if and only if it can be computed by a relabeling tree transducer. Therefore, we characterize the semantical restriction “being shape preserving” by the syntactic restriction “being relabeling”. Moreover, we also show that the shape preserving property is decidable for top-down tree transducers (Theorem 4.4) as well as for bottom-up tree transducers (Theorem 4.19).

We should mention that our result, namely that $SHAPE = QREL$, is a natural generalization of a result concerning generalized sequential machines. We discuss this result in the following. A generalized sequential machine (gsm, cf. [Gin66], see also [Eil74, Ber79]) is a system $M = (Q, \Sigma, \Delta, q_0, \delta)$, where Q is the set of states; Σ and Δ are the input and the output alphabets, respectively; q_0 is the initial state; and δ , the transition function, is a mapping from $Q \times \Sigma$ to $Q \times \Delta^*$. Then δ extends from $Q \times \Sigma^*$ to the $Q \times \Delta^*$ in a standard way and the translation defined by M is the set $\tau_M = \{(x, y) \in \Sigma^* \times \Delta^* \mid (q, y) \in \delta(q_0, x)\}$. In general the length of an input string $x \in \Sigma^*$ and of the output string $y = \tau_M(x)$ is not the same, however if τ_M has this property then M is called a length preserving gsm. For instance if M is a Mealy automaton [Mea55], i.e., δ maps to $Q \times \Delta$, then M is length preserving. On the other hand, it was shown that if a gsm is length preserving then it is equivalent to a Mealy automaton (cf. Propositions XI.3.1 and IX.6.2 in [Eil74] and see also Theorem 3 in [Leg81]). One can easily observe that top-down tree transducers and bottom-up tree transducers are both, in fact, generalizations of gsm’s from strings to trees. Thus, the shape preserving

property of tree transducers corresponds to the length preserving property of gsm's, and the characterization of shape preserving tree transducers by relabeling tree transducers is the generalization of the above result, namely the characterization of gsm's by Mealy automata.

1.5 The Outline of the Thesis

The Thesis has the following structure. In Chapter 2 the necessary definitions and preliminary results are given. Here we also provide examples of shape preserving top-down and bottom-up tree transducers. We will use these examples many times in the Thesis as running examples.

In Chapter 3 we show the above discussed characterization of shape preserving tree transducers, namely that $SHAPE = QREL$. The results of this chapter are from [FG03] which paper discusses the top-down case and from [Gaz06b] which deals with the bottom-up one.

In Section 3.1 we give some useful properties of shape preserving tree transducers. In the top-down case (Subsection 3.1.1) we show that every shape preserving top-down tree transducer is necessarily a, so-called, permutation quasirelabeling (Lemma 3.5). Permutation quasirelabelings have the property that scanning an input symbol with rank $k \neq 1$ they write out exactly one output symbol with rank k and maybe some more unary symbols. Moreover, they can permute the order of the subtrees of an input tree during the computation of an output tree.

In Subsection 3.2.1 we give an algorithm which eliminates the permutation rules from shape preserving permutation quasirelabelings. In this way we show that every shape preserving top-down tree transducer is equivalent to a, so-called, quasirelabeling (Lemma 3.31).

In Subsection 3.3.1 we give an equivalent relabeling to a shape preserving quasirelabeling. Summarizing the above results we can show that shape preserving top-down tree transducers are equivalent to relabelings (Theorem 3.64)

On the other hand, in Subsection 3.1.2 we show useful properties of shape preserving bottom-up tree transducer. Using these properties, in Subsection 3.2.2, we show that every shape preserving bottom-up tree transducer, provided that it computes an infinite tree transformation, is a, so-called, transformable tree transducer (Lemma 3.36). Then we introduce the concept of frame transducers of transformable tree transducers (Definition 3.38). In Corollary 3.44 we show that, for a transformable bottom-up tree transducer M and its frame transducer $fr(M)$, a strong connection holds, namely that $\tau_M = g^{-1} \circ \tau_{fr(M)} \circ h$, where g and h are special functions, called tree homomorphisms (cf. e.g. Definition II.4.13 in [GS84]).

In Subsection 3.2.3 we show that the frame transducer $fr(M)$ of a shape preserving bottom-up tree transducer M is also shape preserving (Lemma 3.52). Then, in Subsection 3.3.2, using Theorem 3.64, we define the relabeling frame transducer $rfr(M)$ of M which is a relabeling equivalent to $fr(M)$ (Definition 3.65). Finally, we give an equivalent relabeling to a shape preserving bottom-up tree transducer M using its relabeling frame transducer $rfr(M)$ (Theorem 3.70).

In Chapter 4 we show that it is decidable whether a given (top-down or bottom-up) tree transducer computes a shape preserving tree transformation or not. Subsection 4.1.1 deals with the top-down case. Here we show that the shape preserving property of top-down tree transducers is decidable (Theorem 4.4).

In Subsection 4.1.2, we show that it is decidable whether a transformable tree transducer is shape preserving (Lemma 4.5). Moreover, we show that it is also decidable if a bottom-up tree transducer M is transformable, provided that M satisfies certain decidable conditions (Lemma 4.18). Using these results we can prove that the shape preserving property of bottom-up tree transducers is decidable (Theorem 4.19).

In this chapter we also show that, for every shape preserving (top-down or bottom-up) tree transducer, an equivalent relabeling can be effectively given. It was shown in [AB93] that the equivalence problem for relabeling tree transducers is decidable. Now using these results we can easily show that the equivalence of shape preserving tree transducers is also decidable (Corollary 4.20).

The corresponding papers in this chapter are [FG03] and [Gaz06a]. In [FG03] it was shown that the shape preserving property for top-down tree transducers is decidable. In [Gaz06a] the analogous result was given for bottom-up tree transducers.

Finally, in Chapter 5 we draw some conclusions.

Chapter 2

Preliminaries

2.1 Definitions, Notations, and Preliminary Results

Let A and B be sets. $A \subseteq B$ denotes that A is a subset of B . Proper inclusion is denoted by $A \subset B$, the cardinality of A by $||A||$, and the empty set by \emptyset . The set of nonnegative integers is denoted by \mathbb{N} , and for every $k \in \mathbb{N}$, $[k]$ is the set $\{1, \dots, k\}$. A subset ρ of $A \times B$ is called a (*binary*) *relation* from A to B . In case $B = A$ we also say that ρ is a relation over A . Sometimes we express that $(a, b) \in \rho$ by writing apb . We define the *domain* and the *range* of ρ , denoted by $dom(\rho)$ and $ran(\rho)$, respectively, by $dom(\rho) = \{a \in A \mid \exists b \in B \text{ such that } (a, b) \in \rho\}$ and $ran(\rho) = \{b \in B \mid \exists a \in A \text{ such that } (a, b) \in \rho\}$.

For every $a \in A$, $\rho(a)$ denotes the set $\{b \mid (a, b) \in \rho\}$. If $\rho(a) = \{b\}$, for some $b \in B$, then sometimes $\rho(a)$ denotes the element b rather than the set $\{b\}$. If, for every $a \in A$, $||\rho(a)|| = 1$ (resp. $||\rho(a)|| \leq 1$), then ρ is a *mapping* (resp. *partial mapping*) from A to B . As usual, we also write $\rho : A \rightarrow B$ to denote that ρ is a (partial) mapping from A to B . Let $\rho : A \rightarrow A$ be a mapping. We say that ρ is a *permutation* (of the set A) if $dom(\rho) = ran(\rho) = A$ and, for every $a, b \in A$ such that $a \neq b$, the condition $\rho(a) \neq \rho(b)$ holds.

Let ρ be a relation over A . The inverse of ρ , denoted by ρ^{-1} , is the set $\{(b, a) \mid (a, b) \in \rho\}$. Let C be a set and $\rho_1 \subseteq A \times B$, $\rho_2 \subseteq B \times C$ relations. The *composition* of ρ_1 and ρ_2 is denoted by $\rho_1 \circ \rho_2$ and defined by

$$\rho_1 \circ \rho_2 = \{(a, c) \mid \exists b \in B : (a, b) \in \rho_1 \text{ and } (b, c) \in \rho_2\}.$$

Moreover, if R_1 and R_2 are classes of relations, then $R_1 \circ R_2$ denotes the class of relations $\{\rho_1 \circ \rho_2 \mid \rho_1 \in R_1 \text{ and } \rho_2 \in R_2\}$.

Let ρ be a relation over A . We say that ρ is *reflexive* (resp. *irreflexive*) if, for every $a \in A$, $(a, a) \in \rho$ (resp. $(a, a) \notin \rho$). We say that ρ is *symmetric* if, for every $(a, b) \in \rho$,



$(b, a) \in \rho$. Finally, ρ is called *transitive* if, for every $(a, b), (b, c) \in \rho$, $(a, c) \in \rho$ also holds. A relation over A is called an *equivalence relation* (resp. a *strict partial order*) on A , if it is reflexive, symmetric and transitive (resp. irreflexive and transitive).

An *alphabet* A is a finite, nonempty set of symbols. We denote by A^* the set of *strings* (or *words*) over A , we let $A^+ = A^* - \{\varepsilon\}$, where ε is the *empty string*. A string $u \in A^*$ is the *prefix* of a $w \in A^*$ if there is a $v \in A^*$ such that $uv = w$. We say that u is a *proper prefix* of w if u is a prefix of w and $u \neq w$. Moreover u and w are *comparable* if u is a prefix of w or w is a prefix of u . If u and v are not comparable, then we call them *incomparable*. The *length* of a string $w \in A^*$ is defined in the usual way and is denoted by $\text{length}(w)$. Moreover, for every $k \in \mathbb{N}$, we put $A^{*,k} = \{w \in A^* \mid \text{length}(w) \leq k\}$. The i th letter of a string w is denoted by $w(i)$.

A *ranked alphabet* is a pair (Σ, rank) , where Σ is an alphabet and rank is a mapping from Σ to \mathbb{N} . For every $k \geq 0$, we denote by $\Sigma^{(k)}$ the set of symbols $\sigma \in \Sigma$ with $\text{rank}(\sigma) = k$ and, for a symbol $\sigma \in \Sigma$ we write $\sigma^{(k)}$ to denote that $\sigma \in \Sigma^{(k)}$.

Let A be a set disjoint with Σ . The set of (*finite, labeled and ordered*) *trees* over Σ indexed by A , denoted by $T_\Sigma(A)$, is the smallest subset T of $(\Sigma \cup A \cup \{(\cdot, \cdot)\} \cup \{, \})^*$, such that (i) $A \subseteq T$ and (ii) if $\sigma \in \Sigma^{(k)}$ with $k \geq 0$ and $s_1, \dots, s_k \in T$, then $\sigma(s_1, \dots, s_k) \in T$. In case $k = 0$, we identify $\sigma(\cdot)$ with σ . Moreover, $T_\Sigma(\emptyset)$ is denoted by T_Σ . It should be clear that $T_\Sigma = \emptyset$ if and only if $\Sigma^{(0)} = \emptyset$.

A *tree language* is a subset of T_Σ and a *tree transformation* is a subset of $T_\Sigma \times T_\Delta$, i.e., a tree transformation is a relation from T_Σ to T_Δ , where Σ and Δ are ranked alphabets.

We will need the set $X = \{x_1, x_2, \dots\}$ of *variable symbols*. For every $k \geq 0$, we define $X_k = \{x_1, \dots, x_k\}$, thus $X_0 = \emptyset$. We use the variables to occur in trees, so we will frequently consider the sets $T_\Sigma(X)$, $T_\Sigma(X_k)$, etc. of trees where Σ is a ranked alphabet. We identify $T_{\Sigma^{(1)}}(X_1)$ with $(\Sigma^{(1)})^*$.

We distinguish a subset $\widehat{T}_\Sigma(X_k)$ of $T_\Sigma(X_k)$ as follows. A tree $t \in T_\Sigma(X_k)$ is in $\widehat{T}_\Sigma(X_k)$ if for every $1 \leq i \leq k$, the variable x_i occurs exactly once in t and, reading the leaves of t from left to right, the variables occur in the order $x_1 < x_2 < \dots < x_k$. Note that $\widehat{T}_{\Sigma^{(1)}}(X_1) = T_{\Sigma^{(1)}}(X_1) = (\Sigma^{(1)})^*$.

The *tree substitution* is defined as follows. Let $t \in T_\Sigma(X_k)$ and let t_1, \dots, t_k be also trees over (maybe other) ranked alphabets. Then $t[t_1, \dots, t_k]$ stands for the tree which is obtained from t by substituting, for every $1 \leq i \leq k$, the tree t_i for every occurrence of x_i . If $\gamma \in (\Sigma^{(1)})^*$, then $\gamma[t]$ is also denoted by γt in order to avoid too many parentheses. Now let $t \in T_\Sigma(X_1)$ and $n \geq 0$. Then t^n is defined as follows.

(i) If $n = 0$, then $t^n = x_1$.

(ii) If $n > 0$, then $t^n = t^{n-1}[t]$.

If Q is a unary ranked alphabet, i.e., the rank of all symbols in Q is 1, and A is a set, then $Q(A)$ stands for the set $\{q(a) \mid q \in Q \text{ and } a \in A\}$.

Now we introduce some characteristics of trees, namely we define the height and the set of occurrences of a tree.

Let Σ be a ranked alphabet and A be a set. For an arbitrary $s \in T_\Sigma(A)$ the *height* of s ($height(s)$) and the set of *occurrence* of s ($occ(s)$) are defined as follows.

- (i) If $s \in \Sigma^{(0)} \cup A$, then $height(s) = 1$, $occ(s) = \{\varepsilon\}$.
- (ii) If $s = \sigma(s_1, \dots, s_k)$ for some $\sigma \in \Sigma^{(k)}$, $k \geq 1$ and $s_1, \dots, s_k \in T_\Sigma(A)$, then
 - $height(s) = 1 + \max\{height(s_i) \mid 1 \leq i \leq k\}$,
 - $occ(s) = \{\varepsilon\} \cup \{w \mid w = iv, 1 \leq i \leq k, v \in occ(s_i)\}$.

Obviously, $height(s) \in \mathbb{N}$, while $occ(s) \subseteq \mathbb{N}^*$, where in this latter case numbers are considered as symbols.

Now, for $s \in T_\Sigma(A)$, and $w \in occ(s)$, we can define the *subtree* of s at w ($stree(s, w)$) as follows.

- (i) If $s \in \Sigma^{(0)} \cup A$ (and thus $w = \varepsilon$), then $stree(s, w) = s$.
- (ii) If $s = \sigma(s_1, \dots, s_k)$ for some $\sigma \in \Sigma^{(k)}$ with $k \geq 1$ and $s_1, \dots, s_k \in T_\Sigma(A)$, then
 - if $w = \varepsilon$, then $stree(s, w) = s$, otherwise,
 - if $w = iv$ for some $1 \leq i \leq k$, then $stree(s, w) = stree(s_i, v)$.

Hence $stree(s, w) \in T_\Sigma(A)$.

For trees $s, t \in T_\Sigma$, we say that t is a *subtree* of s if there is a $w \in occ(s)$ with $stree(s, w) = t$. Moreover t is a *proper subtree* of s if t is a subtree of s and $t \neq s$.

Let $s \in T_\Sigma(X_k)$, for some $k \geq 1$ and let $i \in [k]$ such that the variable x_i occurs exactly once in s . Then we denote by $occ(s, x_i)$ the unique occurrence $w \in occ(s)$ for which $stree(s, w) = x_i$.

Let Σ and Δ be a ranked alphabets. Two trees $s \in \hat{T}_\Sigma(X_k)$ and $t \in \hat{T}_\Delta(X_k)$, where $k \geq 0$, have the same shape, denoted by $s \approx t$, if $occ(s) = occ(t)$, and, for every $i \in [k]$, we have $occ(s, x_i) = occ(t, x_i)$. Certainly $\Delta = \Sigma$ and $k = 0$ is possible, in which case \approx is an equivalence relation over T_Σ . We note that, for a given $s \in \hat{T}_\Sigma(X_k)$, there are only finitely many $t \in \hat{T}_\Sigma(X_k)$ such that $s \approx t$. For instance, if $\Sigma = \{\alpha^{(0)}, \beta^{(0)}, \gamma^{(1)}, \sigma^{(2)}, \delta^{(2)}\}$ and $\Delta = \Sigma$, then $\alpha \approx \beta$, $\sigma(\alpha, \gamma(\beta)) \approx \delta(\beta, \gamma(\alpha))$ and $\sigma(x_1, \delta(\beta, \gamma(x_2))) \approx \delta(x_1, \sigma(\alpha, \gamma(x_2)))$. If s and t do not have the same shape, then we write $s \not\approx t$. A tree transformation $\tau \subseteq T_\Sigma \times T_\Delta$ is *shape preserving* if, for every $(s, t) \in \tau$, we have $s \approx t$.

Next we introduce the concept of *tree homomorphisms*. Let Σ and Δ be ranked alphabets and let $\bar{h} : \Sigma \rightarrow T_\Delta(X)$ be a mapping with the property that if $\sigma \in \Sigma^{(k)}$ for some $k \geq 0$, then $\bar{h}(\sigma) \in T_\Delta(X_k)$ holds. The tree homomorphism induced by \bar{h} is the mapping $h : T_\Sigma(X) \rightarrow T_\Delta(X)$ defined by induction as follows.

- (i) If $s \in X$, then $h(s) = s$.
- (ii) If $s = \sigma(s_1, \dots, s_k)$ for some $k \geq 0$, $\sigma \in \Sigma^{(k)}$ and $s_1, \dots, s_k \in T_\Sigma(X)$, then $h(s) = \bar{h}(\sigma)[h(s_1), \dots, h(s_k)]$.

We say that the tree homomorphism h is *linear* (resp. *nondeleting*), if, for every $k \geq 1$ and $\sigma \in \Sigma^{(k)}$, the condition that $\bar{h}(\sigma)$ contains every variable $x \in X_k$ at most once (resp. at least once) holds.

A *tree transducer* [Rou70, Tha70, Tha73, Eng75] is a system $M = (Q, \Sigma, \Delta, q_0, R)$, where Q is a unary ranked alphabet, called the set of *states*; Σ and Δ are ranked alphabets called the *input* and the *output ranked alphabet*, respectively, satisfying that $Q \cap (\Sigma \cup \Delta) = \emptyset$; $q_0 \in Q$ is the *designated state*; and R is a finite set of *rewriting rules* such that either (i) every rule in R is of the form $q(\sigma(x_1, \dots, x_k)) \rightarrow r$ with $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$ and $r \in T_\Delta(Q(X_k))$ or (ii) every rule in R is of the form $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ with $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q, q_1, \dots, q_k \in Q$ and $r \in T_\Delta(X_k)$. In Case (i) M is called a *top-down tree transducer* and q_0 is the *initial state* while in Case (ii) M is a *bottom-up tree transducer* and q_0 is the *final state*. Now, let $lhs \rightarrow rhs$ be a rule of a tree transducer. We say that lhs and rhs are the *left-hand side* and the *right-hand side* of this rule, respectively. We note, that this definition of tree transducers differs from the one we gave in the introduction, since there tree transducers have a set of designated states, rather than one designated state. However, we can define tree transducers as above without loss of generality, since in this Thesis we consider only nondeterministic tree transducers (i.e., tree transducers which can have different rules with the same left-hand side), and to a nondeterministic tree transducer with a set of designated states, an equivalent one can be constructed, which has only one designated state.

The *derivation relation* induced by a tree transducer M is a binary relation \Rightarrow_M over the set $T_{Q \cup \Sigma \cup \Delta}(X)$ defined as follows. If M is a top-down tree transducer, then for $s, t \in T_{Q \cup \Sigma \cup \Delta}(X)$, we write $s \Rightarrow_M t$ if and only if there is a rule $q(\sigma(x_1, \dots, x_k)) \rightarrow r$ in R and t is obtained from s by replacing an occurrence of a subtree $q(\sigma(s_1, \dots, s_k))$ of s by $r[s_1, \dots, s_k]$, where $s_1, \dots, s_k \in T_\Sigma(X)$. If M is a bottom-up tree transducer, then for $s, t \in T_{Q \cup \Sigma \cup \Delta}(X)$, we write $s \Rightarrow_M t$ if and only if there is a rule $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R and t is obtained from s by replacing an occurrence of a subtree $\sigma(q_1(t_1), \dots, q_k(t_k))$ of s by $q(r[t_1, \dots, t_k])$, where $t_1, \dots, t_k \in T_\Delta(X)$. The

reflexive, transitive closure of \Rightarrow_M is denoted by \Rightarrow_M^* . Then the tree transformation computed by M in a state $q \in Q$ is the relation

$$\tau_{M,q} = \begin{cases} \{(s, t) \in T_\Sigma \times T_\Delta \mid q(s) \Rightarrow_M^* t\}, & \text{if } M \text{ is a top-down tree transducer} \\ \{(s, t) \in T_\Sigma \times T_\Delta \mid s \Rightarrow_M^* q(t)\}, & \text{if } M \text{ is a bottom-up tree transducer.} \end{cases}$$

The tree transformation computed by M is $\tau_M = \tau_{M,q_0}$. Note that, for every state $q \in Q$, $\text{dom}(\tau_{M,q})$ is finite if and only if $\tau_{M,q}$ is finite. Two tree transducers M and M' are *equivalent* if $\tau_M = \tau_{M'}$.

In the rest of the Thesis we will frequently investigate the decomposition of a derivation (or: computation) of tree transducers. In fact, we will use the following, slightly modified versions of decomposition lemmas of [Eng75].

Proposition 2.1 (cf. Lemma 1.2 of [Eng75]) Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a top-down tree transducer. Then the following holds.

1. For every $\sigma \in \Sigma^{(0)}$, $q \in Q$ and $t \in T_\Delta$, if $q(\sigma) \Rightarrow_M^* t$, then the rule $q(\sigma) \rightarrow t$ is in R .
2. For every $k \geq 1$, $t \in T_\Delta(X_k)$ such that every $x \in X_k$ occurs exactly once in t , $q_1, \dots, q_k \in Q$, $s_1, \dots, s_k \in T_\Sigma(X)$, and $t' \in T_\Delta(Q(X))$, if $t[q_1(s_1), \dots, q_k(s_k)] \Rightarrow_M^* t'$, then there are trees $t_1, \dots, t_k \in T_\Delta(Q(X))$ such that $t' = t[t_1, \dots, t_k]$, and, for every $1 \leq i \leq k$, $q_i(s_i) \Rightarrow_M^* t_i$.
3. For every $k \geq 1$, $t \in T_\Delta(X_k)$, $s_1, \dots, s_k \in T_\Sigma(X)$, $q_1, \dots, q_k \in Q$ and $t_1, \dots, t_k \in T_\Delta(Q(X))$, if, for every $1 \leq i \leq k$, $q_i(s_i) \Rightarrow_M^* t_i$, then $t[q_1(s_1), \dots, q_k(s_k)] \Rightarrow_M^* t[t_1, \dots, t_k]$. ■

Proposition 2.2 (cf. Lemma 1.1 of [Eng75]) Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer. Then the following holds.

1. For every $\sigma \in \Sigma^{(0)}$, $q \in Q$ and $t \in T_\Delta$, if $\sigma \Rightarrow_M^* q(t)$, then the rule $\sigma \rightarrow q(t)$ is in R .
2. For every $k \geq 1$, $\sigma \in \Sigma^{(k)}$, $s_1, \dots, s_k \in T_\Sigma(Q(X))$, $q \in Q$ and $t \in T_\Delta(X)$, if $\sigma(s_1, \dots, s_k) \Rightarrow_M^* q(t)$, then there are states $q_1, \dots, q_k \in Q$ and trees $t_1, \dots, t_k \in T_\Delta(X)$ such that $\sigma(s_1, \dots, s_k) \Rightarrow_M^* \sigma(q_1(t_1), \dots, q_k(t_k)) \Rightarrow_M q(t)$ and, for every $1 \leq i \leq k$, $s_i \Rightarrow_M^* q_i(t_i)$.
3. For every $k \geq 1$, $\sigma \in \Sigma^{(k)}$, $s_1, \dots, s_k \in T_\Sigma(Q(X))$, $q_1, \dots, q_k \in Q$ and $t_1, \dots, t_k \in T_\Delta(X)$, if, for every $1 \leq i \leq k$, $s_i \Rightarrow_M^* q_i(t_i)$, then $\sigma(s_1, \dots, s_k) \Rightarrow_M^* \sigma(q_1(t_1), \dots, q_k(t_k))$. ■

Now, let $M = (Q, \Sigma, \Delta, q_0, R)$ be a top-down (resp. bottom-up) tree transducer. A rule $q(\sigma(x_1, \dots, x_k)) \rightarrow r$ (resp. $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$) in R is *useful* if it takes part in a successful derivation. More exactly, the mentioned rule is useful, if there are $u \in \widehat{T}_\Sigma(X_1)$, $v \in \widehat{T}_\Delta(X_1)$, $s_1, \dots, s_k \in T_\Sigma$, $t_1, \dots, t_k \in T_\Delta$ and $t \in T_\Delta$ such that

$$q_0(u[\sigma(s_1, \dots, s_k)]) \Rightarrow_M^* v[q(\sigma(s_1, \dots, s_k))] \Rightarrow_M v[r[q_1(s_1), \dots, q_k(s_k)]] \Rightarrow_M^* t$$

$$(\text{resp. } u[\sigma(s_1, \dots, s_k)] \Rightarrow_M^* u[\sigma(q_1(t_1), \dots, q_k(t_k))] \Rightarrow_M u[q(r[t_1, \dots, t_k])] \Rightarrow_M^* q_0(t)).$$

The useless rules can be eliminated from M by a standard construction. A state $q \in Q$ is useful if it is on the right-hand side (resp. on the left-hand side) of a useful rule. Note that if q is useful, then $\text{dom}(\tau_{M,q}) \neq \emptyset$ and $\text{ran}(\tau_{M,q}) \neq \emptyset$. Throughout the Thesis all tree transducers we consider are assumed to have only useful rules and states.

A tree $s \in T_\Sigma$ is called an *input tree* to M or just an input tree. A tree $t \in T_\Delta$ satisfying $s \Rightarrow_M^* q(t)$ (resp. $q(s) \Rightarrow_M^* t$) for some $s \in T_\Sigma$ and $q \in Q$ is called an *output tree*. Hence input trees and output trees are trees over the input and the output ranked alphabet, respectively.

Next we define some restrictions on tree transducers. Therefore first let $M = (Q, \Sigma, \Delta, q_0, R)$ be a top-down tree transducer. We say that M is

- (a) *linear* (resp. *nondeleting*) if for every rule $q(\sigma(x_1, \dots, x_k)) \rightarrow r$ in R and index $i \in [k]$, x_i occurs at most once (resp. at least once) in r ;
- (b) a *permutation top-down quasirelabeling tree transducer* (or a permutation quasirelabeling to be short) if every rule in R has either of the following forms
 - (1) $q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1 q_1(x_{\pi(1)}), \dots, \gamma_k q_k(x_{\pi(k)}))$, where $k \neq 1$, $q, q_1, \dots, q_k \in Q$, $\sigma \in \Sigma^{(k)}$, $\delta \in \Delta^{(k)}$, $\gamma, \gamma_1, \dots, \gamma_k \in (\Delta^{(1)})^*$ and $\pi : [k] \rightarrow [k]$ is a permutation, and
 - (2) $q(\sigma(x_1)) \rightarrow \gamma p(x_1)$, where $p, q \in Q$, $\sigma \in \Sigma^{(1)}$ and $\gamma \in (\Delta^{(1)})^*$;
- (c) a *top-down quasirelabeling tree transducer* if it is a permutation top-down quasirelabeling tree transducer and, for every rule

$$q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1 q_1(x_{\pi(1)}), \dots, \gamma_k q_k(x_{\pi(k)}))$$

in R , π is the identity permutation, i.e., $\pi(1) = 1, \dots, \pi(k) = k$;

- (d) a *top-down relabeling tree transducer* if every rule in R is of the form

$$\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(\delta(x_1, \dots, x_k)),$$

where $\delta \in \Delta^{(k)}$;

(e) *shape preserving* if τ_M is shape preserving.

Now let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer. We say that M is

- (a) *linear* (resp. *nondeleting*) if for every rule $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R and index $i \in [k]$, x_i occurs at most once (resp. at least once) in r ;
- (b) a *bottom-up quasirelabeling tree transducer* if every rule in R has either of the following forms.
 - (1) $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(\gamma\delta(\gamma_1x_1, \dots, \gamma_kx_k))$, where $k \neq 1$, $\delta \in \Delta^{(k)}$ and $\gamma, \gamma_1, \dots, \gamma_k \in (\Delta^{(1)})^*$, and
 - (2) $\sigma(p(x_1)) \rightarrow q(\gamma(x_1))$, where $p, q \in Q$, $\sigma \in \Sigma^{(1)}$ and $\gamma \in (\Delta^{(1)})^*$;

(c) a *bottom-up relabeling tree transducer* if every rule in R is of the form

$$\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(\delta(x_1, \dots, x_k)),$$

where $\delta \in \Delta^{(k)}$;

(d) *shape preserving* if τ_M is shape preserving.

Clearly, both top-down and bottom-up (quasi)relabeling tree transducers are linear and nondeleting. It is known (Theorem 2.9 in [Eng75]) that linear and nondeleting top-down tree transducers compute the same tree transformation class as the one computed by linear and nondeleting bottom-up tree transducers. Moreover, by the proof of Theorem 2.9 of [Eng75], for every (quasi)relabeling M , a (quasi)relabeling M' can be effectively constructed, such that M' is equivalent to M and the following holds. If M is a top-down tree transducer then M' is a bottom-up tree transducer and vice versa. Therefore, if there is no danger of confusion, we will drop the attribute top-down (resp. bottom-up) from the name top-down (resp. bottom-up) (quasi)relabeling tree transducer, and we will simply write *(quasi)relabeling* instead.

The class of tree transformations computed by relabelings and shape preserving top-down or bottom-up tree transducers is denoted by *QREL* and *SHAPE*, respectively.

We introduce top-down tree automata as special top-down relabeling tree transducers because this will be convenient in what follows. A *top-down tree automaton* is a top-down relabeling tree transducer $T = (Q, \Sigma, \Delta, q_0, R)$ such that $\Sigma = \Delta$ and each rule in R has the form $q(\sigma(x_1, \dots, x_k)) \rightarrow \sigma(q_1(x_1), \dots, q_k(x_k))$. Since the input and the output ranked alphabets are the same we can also write $T = (Q, \Sigma, q_0, R)$. The tree transformation τ_T is a partial mapping from T_Σ to T_Σ . The *tree language recognized by* T , denoted by $L(T)$, is the domain (and hence the range) of τ_T . A tree language L is *recognizable* if there is a top-down tree automaton which recognizes L .

2.2 Two Examples of Shape Preserving Tree Transducers

Now we demonstrate the behavior of shape preserving tree transducers by giving two examples. First we present a shape preserving top-down tree transducer.

Example 2.3 Let $M_t = (Q, \Sigma, \Delta, q_0, R)$ be a top-down tree transducer, where

- $Q = \{q_0, q_1, \dots, q_5\}$,
- $\Sigma = \{\alpha_1^{(0)}, \alpha_2^{(0)}, \alpha_3^{(0)}, \gamma^{(1)}, \sigma^{(3)}\}$,
- $\Delta = \{\beta_1^{(0)}, \beta_2^{(0)}, \beta_3^{(0)}, \omega^{(1)}, \delta^{(3)}\}$,
- R is the set of the rules

$$\begin{aligned}
 \mu_1 &= q_0(\gamma(x_1)) \rightarrow q_1(x_1), \\
 \mu_2 &= q_1(\sigma(x_1, x_2, x_3)) \rightarrow \omega(\delta(\omega(q_2(x_2)), q_3(x_1), q_0(x_3))), \\
 \mu_3 &= q_2(\gamma(x_1)) \rightarrow q_4(x_1), \\
 \mu_4 &= q_3(\gamma(x_1)) \rightarrow \omega(q_5(x_1)), \\
 \mu_5 &= q_4(\alpha_2) \rightarrow \beta_1, \\
 \mu_6 &= q_5(\alpha_1) \rightarrow \beta_2, \\
 \mu_7 &= q_0(\alpha_3) \rightarrow \beta_3.
 \end{aligned}$$

It is easy to see that M_t is a permutation quasirelabeling. To see that M_t is also shape preserving, let $u = \gamma(\sigma(\gamma\alpha_1, \gamma\alpha_2, x_1))$ and $v = \omega\delta(\omega\beta_2, \omega\beta_1, x_1)$. Clearly, $\tau_{M_t} = \{(u^n[\alpha_3], v^n[\beta_3]) \mid n \geq 0\}$ and, for every $n \geq 0$, $u^n[\alpha_3] \approx v^n[\beta_3]$, which means that M_t is shape preserving. Let us consider now the derivation $q_0(u[\alpha_3]) \Rightarrow_{M_t}^* v[\beta_3]$ in more detail.

$$\begin{aligned}
 q_0(u[\alpha_3]) &= q_0(\gamma(\sigma(\gamma\alpha_1, \gamma\alpha_2, \alpha_3))) \\
 &\Rightarrow_M q_1(\sigma(\gamma\alpha_1, \gamma\alpha_2, \alpha_3)) \\
 &\Rightarrow_M \omega\delta(\omega q_2(\gamma\alpha_2), q_3(\gamma\alpha_1), q_0(\alpha_3)) \\
 &\Rightarrow_M^2 \omega\delta(\omega q_4(\alpha_2), \omega q_5(\alpha_1), q_0(\alpha_3)) \\
 &\Rightarrow_M^3 \omega\delta(\omega\beta_1, \omega\beta_2, \beta_3) \\
 &= v[\alpha_3].
 \end{aligned}$$

It can be seen that μ_1 and μ_3 both scan the input symbol γ , but none of them write out any output symbol. The application of μ_2 compensates the deletion made by μ_1 and μ_3 since it writes out more than one output symbols. Actually, μ_2 scans the input symbol σ which has rank three, and writes out the output symbol δ , which also has rank three, and two additional unary symbols.

This example also demonstrates that a shape preserving top-down tree transducer can have a rule such that a real (i.e. not the identity) permutation of the variables occurs in the right-hand side of the rule (cf. the rule μ_2). \square

Next we give an example of a shape preserving bottom-up tree transducer.

Example 2.4 Let $M_b = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer, where

- $Q = \{q_0, q_1, q_2, q_3, q_\alpha\},$
- $\Sigma = \{\alpha^{(0)}, \gamma^{(1)}, \gamma_1^{(1)}, \sigma_1^{(2)}, \sigma^{(3)}\},$
- $\Delta = \{\beta^{(0)}, \beta_1^{(0)}, \beta_2^{(0)}, \omega^{(1)}, \delta_1^{(2)}, \delta^{(3)}\},$
- $R = \{\mu_1, \mu_2, \dots, \mu_8\},$ where

$$\begin{aligned}
 \mu_1 &= \sigma(q_1(x_1), q_\alpha(x_2), q_2(x_3)) \rightarrow q_0(\delta(\delta_1(\beta, x_1), x_2, x_3)), \\
 \mu_2 &= \sigma_1(q_\alpha(x_1), q_0(x_2)) \rightarrow q_1(x_2), \\
 \mu_3 &= \sigma_1(q_\alpha(x_1), q_\alpha(x_2)) \rightarrow q_1(x_2), \\
 \mu_4 &= \gamma(q_3(x_1)) \rightarrow q_2(x_1), \\
 \mu_5 &= \gamma(q_3(x_1)) \rightarrow q_3(\omega(x_1)), \\
 \mu_6 &= \gamma_1(q_\alpha(x_1)) \rightarrow q_3(\omega(\omega(x_1))), \\
 \mu_7 &= \alpha \rightarrow q_\alpha(\beta_1), \\
 \mu_8 &= \alpha \rightarrow q_\alpha(\beta_2).
 \end{aligned}$$

We leave the proof, that M_b is shape preserving to the reader, however we provide some help by giving certain subsets of τ_{M_b, q_0} and τ_{M_b, q_2} . Let $a = \sigma(\sigma_1(\alpha, x_1), \alpha, \gamma\gamma_1\alpha)$ and $b = \delta(\delta_1(\beta, x_1), \beta_2, \omega\omega\beta_1)$. Moreover let, for every $n > 0$, $a_n = a^n[\alpha]$, $b_n = b^n[\beta_1]$, $c_n = \gamma^n\gamma_1\alpha$ and $d_n = \omega^{n+1}\beta_2$. It is easy to see that, for every $n > 0$, $(a_n, b_n) \in \tau_{M_b, q_0}$, $(c_n, d_n) \in \tau_{M_b, q_2}$, $a_n \approx b_n$ and $c_n \approx d_n$.

Next we examine a derivation of M_b in detail in order to analyze the shape preserving property of M_b . Let $s = \sigma(\sigma_1(\alpha, s_1), \alpha, s_2)$ and $t = \delta(\delta_1(\beta, t_1), \beta_2, t_2)$, where $s_1 = \sigma(\sigma_1(\alpha, \alpha), \alpha, s_2)$, $s_2 = \gamma\gamma_1\alpha$, $t_1 = \delta(\delta_1(\beta, \beta_1), \beta_2, t_2)$ and $t_2 = \omega\omega\beta_1$. Since $s_2 \approx t_2$, we have that $s_1 \approx t_1$, and thus $s \approx t$ also holds. It is easy to see that there is a derivation of M_b of the following form.

$$\begin{aligned}
 &(\dagger) \\
 s &= \sigma(\sigma_1(\alpha, s_1), \alpha, s_2) \\
 &\Rightarrow_{M_b}^* \sigma(\sigma_1(q_\alpha(\beta_1), q_0(t_1)), \alpha, s_2) \\
 &\Rightarrow_{M_b} \sigma(q_1(t_1), \alpha, s_2) \quad (\text{rule } \mu_2) \\
 &\Rightarrow_{M_b}^* \sigma(q_1(t_1), q_\alpha(\beta_2), q_2(t_2)) \\
 &\Rightarrow_{M_b} q_0(\delta(\delta_1(\beta, t_1), \beta_2, t_2)) \quad (\text{rule } \mu_1) \\
 &= q_0(t).
 \end{aligned}$$

When μ_2 scans the input symbol σ_1 , it does not write out any output symbol, only keeps the tree t_1 and deletes the subtree β_1 . This behavior of μ_2 does not impact the

shape preserving property of M_b , since later M_b applies the rule μ_1 which writes out more than one output symbols compensating the deletion of β_1 by μ_2 and the fact that μ_2 did not write out any output symbol.

Now let us consider the derivation $s \Rightarrow_{M_b}^* q_0(t)$ in another form. Let $u = \sigma(\sigma_1(\alpha, s_1), \alpha, x_1)$ and $v = \delta(\delta_1(\beta, t_1), \beta_2, x_1)$. Then there is a derivation of M_b of the following form.

$$\begin{array}{lll}
 (\dagger) & & \\
 s & = & u[\gamma\gamma_1\alpha] \\
 & \Rightarrow_{M_b} & u[\gamma\gamma_1q_\alpha(\beta_1)] \\
 & \Rightarrow_{M_b} & u[\gamma q_3(\omega\omega\beta_1)] & \text{(rule } \mu_6) \\
 & \Rightarrow_{M_b} & u[q_2(\omega\omega\beta_1)] & \text{(rule } \mu_4) \\
 & \Rightarrow_{M_b}^* & q_0(v[\omega\omega\beta_1]) \\
 & = & q_0(t).
 \end{array}$$

In this derivation the rule μ_6 writes out two unary symbols, but in the next derivation step μ_4 does not write out any output symbol, preserving the shape of the input subtree. \square

Chapter 3

Characterizing of Shape Preserving Tree Transducers

In this chapter we give a characterization of shape preserving top-down tree transducers as well as of shape preserving bottom-up tree transducers. In fact we show that a (top-down or bottom-up) tree transducer is shape preserving if and only if it is equivalent to a relabeling.

3.1 Useful Properties of Shape Preserving Tree Transducers

In this section we show properties of shape preserving tree transducers which we will use in the forthcoming sections. The shape preserving property is a strong restriction for both bottom-up and top-down tree transducers. As we will see in this section, shape preserving tree transducers can have only rather restricted rules. However, there is an important difference between the rules of these two types of shape preserving tree transducers as follows (cf. also Section 1.3). A rule of a shape preserving top-down tree transducer can not be deleting or copying, i.e. every variable occurring on the left-hand side of the rule, must occur exactly once on the right-hand side. On the other hand, a shape preserving bottom-up tree transducer can be deleting and copying as well. However, certain restrictions on rules of a shape preserving bottom-up tree transducer also can be given.

3.1.1 The Top-Down Case

Here we show that every shape preserving top-down tree transducer is a permutation top-down quasirelabeling tree transducer. By definition a permutation quasirelabeling differs from a relabeling in that the right-hand sides of its rules may contain some

extra unary output symbols and a permutation of the variables is also allowed in the right-hand sides.

We begin our work by proving that every shape preserving top-down tree transducer is nondeleting.

Lemma 3.1 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving top-down tree transducer. Then M is nondeleting.

Proof. We prove by contradiction. Let us assume that there is a rule $q(\sigma(x_1, \dots, x_k)) \rightarrow r$ in R such that $k \geq 1$ and, say, x_i does not occur in r . This rule is also useful, so there are $u \in \widehat{T}_\Sigma(X_1)$, $s_1, \dots, s_k \in T_\Sigma$, $v \in \widehat{T}_\Delta(X_1)$ and $t \in T_\Delta$ such that

$$q_0(u[\sigma(s_1, \dots, s_k)]) \Rightarrow_M^* v[q(\sigma(s_1, \dots, s_k))] \Rightarrow_M v[r[s_1, \dots, s_k]] \Rightarrow_M^* t.$$

Since M is shape preserving, $u[\sigma(s_1, \dots, s_k)] \approx t$ holds. Now let us change the involved occurrence of s_i to a s'_i such that $s_i \not\approx s'_i$. Then certainly $u[\sigma(s_1, \dots, s_k)] \not\approx u[\sigma(s_1, \dots, s'_i, \dots, s_k)]$. On the other hand, since x_i does not occur in r , we have $r[s_1, \dots, s'_i, \dots, s_k] = r[s_1, \dots, s_k]$ and thus

$$\begin{aligned} q_0(u[\sigma(s_1, \dots, s'_i, \dots, s_k)]) &\Rightarrow_M^* v[q(\sigma(s_1, \dots, s'_i, \dots, s_k))] \Rightarrow_M \\ v[r[s_1, \dots, s'_i, \dots, s_k]] &= v[r[s_1, \dots, s_k]] \Rightarrow_M^* t. \end{aligned}$$

Since M is shape preserving, $u[\sigma(s_1, \dots, s'_i, \dots, s_k)] \approx t$, a contradiction. ■

The above lemma is the key in showing that every shape preserving top-down tree transducer is a permutation quasirelabeling. However, we need some further preparations. First we define the branch number and the weighted branch number of a tree.

Definition 3.2 Let Σ be a ranked alphabet. A symbol $\sigma \in \Sigma$ is called a *branch symbol* provided that its rank is greater than 1. The *branch number* $\text{bn}(s)$ and the *weighted branch number* $\text{wbn}(s)$ of a tree $s \in T_\Sigma(X)$ are defined by induction as follows.

- (i) If $s \in \Sigma^{(0)}$ or $s \in X$, then $\text{bn}(s) = 0$ and $\text{wbn}(s) = 0$.
- (ii) If $s = \sigma(s_1, \dots, s_k)$ for some $\sigma \in \Sigma^{(k)}$, $k \geq 1$ and $s_1, \dots, s_k \in T_\Sigma(X)$, then
 - if $k = 1$, then $\text{bn}(s) = \text{bn}(s_1)$ and $\text{wbn}(s) = \text{wbn}(s_1)$,
 - if $k > 1$, then $\text{bn}(s) = 1 + \sum_{i=1}^k \text{bn}(s_i)$ and $\text{wbn}(s) = k + \sum_{i=1}^k \text{wbn}(s_i)$.

□

Hence the branch number of a tree s is the sum of the number of the occurrences of the branch symbols in s . Certainly, if $s \approx t$, then $\text{bn}(s) = \text{bn}(t)$ and $\text{wbn}(s) = \text{wbn}(t)$.

Lemma 3.3 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving top-down tree transducer. Then, for every rule $q(\sigma(x_1, \dots, x_k)) \rightarrow r$ in R , we have $\text{bn}(r) \leq 1$.

Proof. We prove by contradiction. Assume that, there is a rule $q(\sigma(x_1, \dots, x_k)) \rightarrow r \in R$ with $\text{bn}(r) > 1$. This rule can be applied in a successful derivation $q_0(s) \Rightarrow_M^* t$ for some $s \in T_\Sigma$ and $t \in T_\Delta$. Since M is shape preserving, $\text{bn}(s) = \text{bn}(t)$. The application of the above rule increases the branch number of the output with respect to the input, hence another rule is needed to compensate the increase. The only chance to decrease the branch number is to apply a rule of the form $p(\delta(x_1, \dots, x_l)) \rightarrow r$, where $l \geq 1$ and r does not contain some of the variables x_1, \dots, x_l . However, by Lemma 3.1, there are no such rules in R . Thus $\text{bn}(s) > \text{bn}(t)$, which is a contradiction. Hence our statement follows. ■

From Lemmas 3.1 and 3.3 we get the following corollary.

Corollary 3.4 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving top-down tree transducer. Then every rule in R has either of the following forms.

1. $q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1 q_1(x_{i_1}), \dots, \gamma_m q_m(x_{i_m}))$, where $k \neq 1$, $m \geq k$, $q, q_1, \dots, q_m \in Q$, $\sigma \in \Sigma^{(k)}$, $\delta \in \Delta^{(m)}$, $\gamma, \gamma_1, \dots, \gamma_m \in (\Delta^{(1)})^*$ and $\{x_{i_1}, \dots, x_{i_m}\} = X_k$. (Note that some x_j may occur more than once in the right-hand side.)
2. $q(\sigma(x_1)) \rightarrow \gamma p(x_1)$, where $q, p \in Q$, $\sigma \in \Sigma^{(1)}$ and $\gamma \in (\Delta^{(1)})^*$.

Next we are going to show that in 1. of the above corollary even $k = m$ holds, which means that a shape preserving top-down tree transducer is a permutation quasirelabeling.

Lemma 3.5 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving top-down tree transducer. Then M is a permutation quasirelabeling.

Proof. Each rule in R is as in 1. or 2. in Corollary 3.4. It is enough to prove that in Case 1. only $m = k$ is possible. This can be shown easily in the following way. If $m > k$, then the application of that rule increases the weighted branch number, which increase cannot be compensated somewhere else, cf. the proof of Lemma 3.3. ■

Note, that we have seen in Example 2.3 that M_t is shape preserving and a permutation quasirelabeling as well.

Now, we state an important property of shape preserving top-down tree transducers which we will use later.

Lemma 3.6 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving top-down tree transducer, $q \in Q$, and let us consider a derivation $q_0(u) \Rightarrow_M^* v[q(x_1)]$ of M , where $u \in \widehat{T}_\Sigma(X_1)$ and $v \in \widehat{T}_\Delta(X_1)$. If $\text{occ}(u, x_1)$ and $\text{occ}(v, x_1)$ are incomparable, then $\text{dom}(\tau_{M,q})$ and $\text{ran}(\tau_{M,q})$ are uniform.

Proof. Let $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$ and assume that α and β are incomparable. We prove that $\text{dom}(\tau_{M,q})$ and $\text{ran}(\tau_{M,q})$ are uniform by contradiction. Therefore assume that $\text{dom}(\tau_{M,q})$ or $\text{ran}(\tau_{M,q})$ is not uniform. We treat the two possible cases together.

Let us suppose that $\text{dom}(\tau_{M,q})$ (respectively $\text{ran}(\tau_{M,q})$) is not uniform. Then there are trees $u_1, u_2 \in \text{dom}(\tau_{M,q})$ and $v_1, v_2 \in \text{ran}(\tau_{M,q})$ such that $u_1 \not\approx u_2$ (resp. $v_1 \not\approx v_2$), $q(u_1) \Rightarrow_M^* v_1$, and $q(u_2) \Rightarrow_M^* v_2$. Then, clearly, there are derivations $q_0(u[u_1]) \Rightarrow_M^* v[q(u_1)] \Rightarrow_M^* v[v_1]$ and $q_0(u[u_2]) \Rightarrow_M^* v[q(u_2)] \Rightarrow_M^* v[v_2]$ of M and, since M is shape preserving, $u[u_1] \approx v[v_1]$ and $u[u_2] \approx v[v_2]$ (cf. Figure 3.1). Now, since $u_1 \not\approx u_2$ (resp. $v_1 \not\approx v_2$) and α and β are incomparable, we have that

$$\text{stree}(u[u_1], \alpha) \not\approx \text{stree}(u[u_2], \alpha) \approx \text{stree}(v[v_2], \alpha) = \text{stree}(v[v_1], \alpha)$$

$$(\text{resp. } \text{stree}(u[u_1], \beta) = \text{stree}(u[u_2], \beta) \approx \text{stree}(v[v_2], \beta) \not\approx \text{stree}(v[v_1], \beta)).$$

However, this implies that $u[u_1] \not\approx v[v_1]$, which is a contradiction. ■

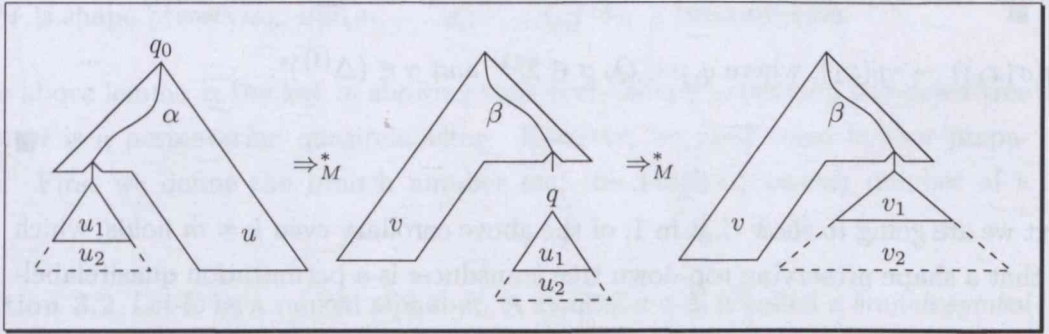


Figure 3.1: If α and β are incomparable, then $\text{dom}(\tau_{M,q})$ and $\text{ran}(\tau_{M,q})$ are uniform.

The next lemma is a consequence of the previous one. We will use it when we construct a quasirelabeling equivalent to a shape preserving permutation quasirelabeling.

Lemma 3.7 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving permutation quasirelabeling. Then for every $k > 1$, permutation rule

$$q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1 q_1(x_{\pi(1)}), \dots, \gamma_k q_k(x_{\pi(k)}))$$

in R , where $q, q_1, \dots, q_k \in Q$, $\sigma \in \Sigma^{(k)}$, $\delta \in \Delta^{(k)}$, $\gamma, \gamma_1, \dots, \gamma_k \in (\Delta^{(1)})^*$ and $\pi : [k] \rightarrow [k]$ is a permutation, and $i \in [k]$, if $\pi(i) \neq i$, then both $\text{dom}(\tau_{M,q_i})$ and $\text{ran}(\tau_{M,q_i})$ are uniform.

Proof. Let $\mu = q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1 q_1(x_{\pi(1)}), \dots, \gamma_k q_k(x_{\pi(k)}))$ be a permutation rule in R with $k > 1$ and assume that for some $i \in [k]$, $\pi(i) \neq i$ holds.

Since the rule is useful, there are $u \in \widehat{T}_\Sigma(X_1)$, $s, s_1, \dots, s_k \in T_\Sigma$, $v \in \widehat{T}_\Delta(X_1)$ and $t, t_1, \dots, t_k \in T_\Delta$ such that

$$\begin{aligned}
 (\dagger) \quad q_0(s) &= q_0(u[\sigma(s_1, \dots, s_k)]) \\
 &\Rightarrow_M^* v[q(\sigma(s_1, \dots, s_k))] \\
 &\Rightarrow_M v[\gamma\delta(\gamma_1 q_1(s_{\pi(1)}), \dots, \gamma_k q_k(s_{\pi(k)}))] \quad (\text{rule } \mu) \\
 &\Rightarrow_M^* v[\gamma\delta(\gamma_1 t_1, \dots, \gamma_k t_k)] \\
 &= t.
 \end{aligned}$$

Let $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$. Now we distinguish three cases.

Case 1: α and β are incomparable. Then, by Lemma 3.6, $\text{dom}(\tau_{M,q})$ and $\text{ran}(\tau_{M,q})$ are uniform. Clearly this implies that the sets $\text{dom}(\tau_{M,q_i})$ and $\text{ran}(\tau_{M,q_i})$ are also uniform.

Case 2: β is a prefix of α . Now $\alpha = \text{occ}(v[\gamma x_1], x_1)$ because M is a shape preserving permutation quasirelabeling and thus the symbol δ written by the application of the rule μ matches the σ being at α in s . Since $\pi(i) \neq i$, the occurrences $\alpha\pi(i)$ and αi are incomparable. Let $u' = u[\sigma(s_1, \dots, s_{\pi(i)-1}, x_1, s_{\pi(i)+1}, \dots, s_k)]$ and $v' = v[\gamma\delta(\gamma_1 t_1, \dots, \gamma_{i-1} t_{i-1}, \gamma_i x_1, \gamma_{i+1} t_{i+1}, \dots, \gamma_k t_k)]$. Clearly, $\alpha\pi(i) = \text{occ}(u', x_1)$, αi is a prefix of $\text{occ}(v', x_1)$. Then $\text{occ}(u', x_1)$ and $\text{occ}(v', x_1)$ are incomparable. Moreover, there is a derivation $q_0(u') \Rightarrow_M^* v'[q_i(x_1)]$ of M , which, by Lemma 3.6, implies that $\text{dom}(\tau_{M,q_i})$ and $\text{ran}(\tau_{M,q_i})$ are uniform.

Case 3: α is a proper prefix of β . Then α is also a proper prefix of $\text{occ}(v[\gamma x_1], x_1)$. On the other hand, similarly as in Case 2, it can be seen that $\alpha = \text{occ}(v[\gamma x_1], x_1)$, which is a contradiction. \blacksquare

3.1.2 The Bottom-Up Case

As we have showed in the previous subsection, a shape preserving top-down tree transducer is nondeleting and linear. Although we can not prove a similar result for shape preserving bottom-up tree transducers, we can show the following. Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and consider a rule $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R . If, for some $i \in [k]$, τ_{M,q_i} is infinite, then x_i occurs exactly once in r . Moreover, for every $i, j \in [k]$ such that $i < j$ and τ_{M,q_i} and τ_{M,q_j} are infinite, the variables x_i and x_j occur in r in the order $x_i < x_j$.

Furthermore, we define the concept of the matching paths and show some of their important properties. These matching paths will be very useful in the forthcoming sections when we show that M is equivalent to a relabeling.

We begin our work by defining the deleting states and the copying states of a bottom-up tree transducer.

Definition 3.8 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer and let q be a state in Q . The state q is *deleting* (resp. *copying*) if there are trees $u \in \widehat{T}_\Sigma(X_1)$ and $v \in T_\Delta(X_1)$ such that there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$, and the variable x_1 does not occur (resp. x_1 occurs more than once) in the tree v . Otherwise q is *nondeleting* (resp. *noncopying*). \square

Now, consider a shape preserving bottom-up tree transducer $M = (Q, \Sigma, \Delta, q_0, R)$ and a state q in Q . We show that if $\tau_{M,q}$ is infinite then q is nondeleting.

Lemma 3.9 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let q be a state in Q such that $\tau_{M,q}$ is infinite. Then q is nondeleting.

Proof. We prove the lemma by contradiction. Let us suppose that q is deleting. Then by Definition 3.8 there are trees $u \in \widehat{T}_\Sigma(X_1)$ and $v \in T_\Delta$ and there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$. Then, since q is useful, there is a derivation $u[u_1] \Rightarrow_M^* u[q(v_1)] \Rightarrow_M^* q_0(v[v_1]) = q_0(v)$, where $u_1 \in T_\Sigma$ and $v_1 \in T_\Delta$. Since $\tau_{M,q}$ is infinite, there is a tree $\bar{u}_1 \in \text{dom}(\tau_{M,q})$ such that $u_1 \not\approx \bar{u}_1$. Let $\bar{v}_1 \in \text{ran}(\tau_{M,q})$ such that $\bar{u}_1 \Rightarrow_M^* q(\bar{v}_1)$. Then there is a derivation $u[\bar{u}_1] \Rightarrow_M^* u[q(\bar{v}_1)] \Rightarrow_M^* q_0(v[\bar{v}_1]) = q_0(v)$. Since M is shape preserving, $u[u_1] \approx v \approx u[\bar{u}_1]$. On the other hand, clearly, $u[u_1] \not\approx u[\bar{u}_1]$, which is a contradiction. \blacksquare

Next we are going to show that if $\tau_{M,q}$ is infinite then q is also noncopying. The proof of this result is more complicated than showing that q is nondeleting. First we show that if the set $\tau_{M,q}$ is infinite then the set $\text{ran}(\tau_{M,q})$ is also infinite.

Lemma 3.10 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and q be a state in Q . If $\tau_{M,q}$ is infinite, then the set $\text{ran}(\tau_{M,q})$ is infinite.

Proof. We prove the statement by contradiction. Therefore let us suppose that $\tau_{M,q}$ is infinite, but the set $\text{ran}(\tau_{M,q})$ is finite.

Then it is easy to see that there are trees $u_1, \bar{u}_1 \in \text{dom}(\tau_{M,q})$ and $v_1, \bar{v}_1 \in \text{ran}(\tau_{M,q})$ such that $u_1 \not\approx \bar{u}_1$, $u_1 \Rightarrow_M^* q(v_1)$, $\bar{u}_1 \Rightarrow_M^* q(\bar{v}_1)$ and $v_1 \approx \bar{v}_1$. Since q is useful, there are derivations of M of the form

$$u[u_1] \Rightarrow_M^* u[q(v_1)] \Rightarrow_M^* q_0(v[v_1]) \quad \text{and} \quad u[\bar{u}_1] \Rightarrow_M^* u[q(\bar{v}_1)] \Rightarrow_M^* q_0(v[\bar{v}_1]),$$

where $u \in \widehat{T}_\Sigma(X_1)$ and $v \in T_\Delta(X_1)$. Clearly, $u[u_1] \not\approx u[\bar{u}_1]$. Moreover, since M is shape preserving, $u[u_1] \approx v[v_1]$ and $u[\bar{u}_1] \approx v[\bar{v}_1]$. Then, since $v \approx \bar{v}_1$, we have that $u[u_1] \approx u[\bar{u}_1]$, which is a contradiction proving the lemma. \blacksquare

We will need the following lemma.

Lemma 3.11 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let q be a state in Q such that $\tau_{M,q}$ is infinite. Let $s = u[u_1] \Rightarrow_M^* u[q(v_1)] \Rightarrow_M^* q_0(v[v_1]) = q_0(t)$ be a derivation of M , where $u \in \widehat{T}_\Sigma(X_1)$, $s, u_1 \in T_\Sigma$, $v \in T_\Delta(X_1)$ such that the variable x_1 occurs in the tree v , and $t, v_1 \in T_\Delta$. Moreover, let $\alpha = \text{occ}(u, x_1)$ and $\beta \in \text{occ}(v)$ such that $\text{stree}(v, \beta) = x_1$. Then α and β are comparable.

Proof. We prove the lemma by contradiction. Let us suppose that α and β are incomparable. Since $\tau_{M,q}$ is infinite, by Lemma 3.10, $\text{ran}(\tau_{M,q})$ is also infinite. Then there is a tree $\bar{v}_1 \in \text{ran}(\tau_{M,q})$ such that $\bar{v}_1 \not\approx v_1$. Let $\bar{u}_1 \in \text{dom}(\tau_{M,q})$ such that $\bar{u}_1 \Rightarrow_M^* q(\bar{v}_1)$. Let us form the input tree $\bar{s} = u[\bar{u}_1]$. Then there is a derivation $\bar{s} = u[\bar{u}_1] \Rightarrow_M^* u[q(\bar{v}_1)] \Rightarrow_M^* q_0(v[\bar{v}_1]) = q_0(\bar{t})$, where $\bar{t} \in T_\Delta$. Since M is shape preserving $s \approx t$ and $\bar{s} \approx \bar{t}$. Then, since α and β are incomparable, we have that $v_1 = \text{stree}(t, \beta) \approx \text{stree}(s, \beta) = \text{stree}(\bar{s}, \beta) \approx \text{stree}(\bar{t}, \beta) = \bar{v}_1$, which contradicts that $v_1 \not\approx \bar{v}_1$. ■

Now we can show that q is noncopying.

Lemma 3.12 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let q be a state in Q such that $\tau_{M,q}$ is infinite. Then q is noncopying.

Proof. We prove the lemma by contradiction. Let us suppose that q is copying. Then by Definition 3.8 there are trees $u \in \widehat{T}_\Sigma(X_1)$ and $v \in T_\Delta(X_1)$ such that there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$ and the variable x_1 occurs more than once in the tree v . Let $\alpha = \text{occ}(u, x_1)$ and $\beta_1, \beta_2 \in \text{occ}(v)$ such that $\beta_1 \neq \beta_2$ and $\text{stree}(v, \beta_1) = \text{stree}(v, \beta_2) = x_1$. Then clearly β_1 and β_2 are incomparable and, by Lemma 3.11, α and β_1 are comparable, and α and β_2 are also comparable. This clearly implies that α is a proper prefix of the occurrences β_1 and β_2 . Since $\tau_{M,q}$ is infinite, there is a tree $\bar{u} \in \text{dom}(\tau_{M,q})$ such that $\bar{u} = u_1[u_2[u_3]]$, where $u_1, u_2 \in \widehat{T}_\Sigma(X_1)$, $u_2 \neq x_1$, $u_3 \in T_\Sigma$ and there is a derivation of M of the following form.

$$u_1[u_2[u_3]] \Rightarrow_M^* u_1[u_2[p(v_3)]] \Rightarrow_M^* u_1[p(v_2[v_3])] \Rightarrow_M^* q(v_1[v_2[v_3]]),$$

where $p \in Q$, $v_1, v_2 \in T_\Delta(X_1)$ and $v_3 \in T_\Delta$. Let, for every $i \in \mathbb{N}$, $u^{(i)} = u_1[u_2^i[u_3]]$, $s^{(i)} = u[u^{(i)}]$, $v^{(i)} = v_1[v_2^i[v_3]]$, $t^{(i)} = v[v^{(i)}]$ and $\alpha_i = \text{occ}(u[u_1[u_2^i]], x_1)$. Clearly, for every $i \in \mathbb{N}$, there is a derivation $u^{(i)} \Rightarrow_M^* q(v^{(i)})$, and therefore

$$s^{(i)} = u[u^{(i)}] \Rightarrow_M^* u[q(v^{(i)})] \Rightarrow_M^* q_0(v[v^{(i)}]) = t^{(i)}.$$

Moreover, since $u_2 \neq x_1$, α_i is a proper prefix of α_{i+1} .

It is easy to see that v_1 and v_2 contain the variable x_1 . Indeed, if one of them does not contain x_1 , then for a sufficiently large $m \in \mathbb{N}$, $\text{height}(s^{(m)}) > \text{height}(t^{(m)})$. On the other hand, since M is shape preserving, $s^{(m)} \approx t^{(m)}$, a contradiction.

It is also not difficult to see that there is a number $n \in \mathbb{N}$ such that either (i) α_n and β_1 are incomparable, or (ii) α_n and β_2 are incomparable. Let us suppose that the case (i) holds (see Figure 3.2, the case (ii) can be considered similarly).

Now, let $\beta'_1 \in \text{occ}(v[v_1[v_2^n[x_1]]])$ such that $\text{stree}(t^{(n)}, \beta'_1) = x_1$ and β_1 is a prefix of β'_1 . Then, clearly, $\alpha^{(n)}$ and β'_1 are incomparable. On the other hand, since $u_2 \neq x_1$, it follows that $\tau_{M,p}$ is infinite. Moreover, it is easy to see, that there is a derivation $s^{(n)} = u[u_1[u_2^n[u_3]]] \Rightarrow_M^* u[u_1[u_2^n[p(v_3)]]] \Rightarrow_M^* q_0(v[v_1[v_2^n[v_3]]]) = t^{(n)}$. Then, by Lemma 3.11, we get that $\alpha^{(n)}$ and β'_1 are comparable, which is a contradiction proving the lemma. ■

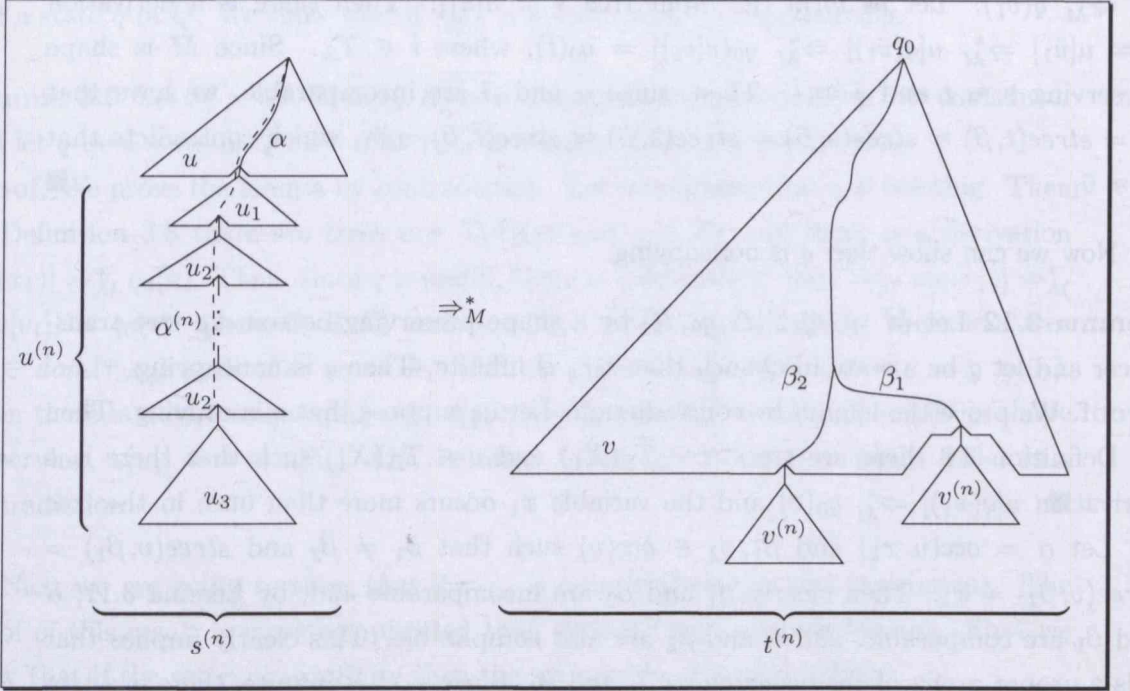


Figure 3.2: If $\tau_{M,q}$ is infinite, then q is noncopying.

Let us consider now the shape preserving bottom-up tree transducer M_b appearing in Example 2.4. It is easy to see that $\{q_i \mid i \in [3]\} \cup \{q_0\}$ is the set of those states in which M_b computes infinite tree transformations. One can also easily observe that these states are both nondeleting and noncopying.

We will use the next corollary frequently when we consider certain derivations of a shape preserving bottom-up tree transducer.

Corollary 3.13 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let q be a state in Q such that $\tau_{M,q}$ is infinite. Then in every derivation $u[u_1] \Rightarrow_M^* u[q(v_1)] \Rightarrow_M^* q_0(v[v_1])$, where $u \in \hat{T}_\Sigma(X_1)$, $u_1 \in T_\Sigma$, $v \in T_\Delta(X_1)$ and $v_1 \in T_\Delta$, the condition $v \in \hat{T}_\Delta(X_1)$ holds.

Proof. Since $\tau_{M,q}$ is infinite, by Lemmas 3.9 and 3.12, the state q is nondeleting and noncopying. Then, by Definition 3.8, in every derivation of M of the form $u[q(x_1)] \Rightarrow_M^* q_0(v)$, the variable x_1 occurs exactly once in the tree v , which proves the statement. ■

In the following definition we associate a set of indexes to every rule of a bottom-up tree transducer.

Definition 3.14 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R . We define

$$\text{inf}(\mu) = \{i \in [k] \mid \tau_{M,q_i} \text{ is infinite}\}.$$

□

We will also use the following corollary.

Corollary 3.15 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R . For every $i \in \text{inf}(\mu)$, the variable x_i occurs exactly once in the tree r .

Proof. Let $i \in \text{inf}(\mu)$. By Definition 3.14, τ_{M,q_i} is infinite, hence $\tau_{M,q}$ is also infinite. Then the statement can be proved easily, using Corollary 3.13. ■

In Subsection 3.2.2 we will construct a so-called frame transducer of a special bottom-up tree transducer, called transformable tree transducer. Later we will show that every shape preserving bottom-up tree transducer M is transformable, provided that τ_M is infinite. The frame transducer computes a tree transformation from which, roughly speaking, we can get back the tree transformation computed by the original transformable tree transducer. Moreover, it has certain good properties which will make the proof of that every shape preserving bottom-up tree transducer is equivalent to a relabeling much easier. One of these properties is that if the original transformable tree transducer is shape preserving, then the frame transducer is also shape preserving. During the construction of the frame transducer and later, when we prove that a shape preserving bottom-up tree transducer has a shape preserving frame transducer, we will need the concept of matching paths. Before the formal definition of the matching paths, we give some intuition with the help of the tree transducer M_b and its derivations (\dagger) and (\ddagger) appearing in Example 2.4.

First we consider the derivation (\dagger) and the application of the rule μ_1 . It is easy to see that the symbol δ written out by μ_1 occurs at the same occurrence in the tree t as the occurrence of σ scanned by μ_1 in the tree s . One can also show that this is true in general, i.e. the following holds. In every derivation $s' \Rightarrow_{M_b}^* q_0(t')$ of M_b , where $s' \in T_\Sigma$

and $t' \in T_\Delta$, such that μ_1 is applied in this derivation, the symbols σ and δ , scanned and written out by μ_1 , occur at the same occurrence in the trees s' and t' , respectively.

We can observe the following. Let $r = \delta(\delta_1(\beta, x_1), x_2, x_3)$, i.e. r is the tree which is written out by μ_1 . The longest common prefix of the occurrences $\{occ(r, x_i) \mid i \in inf(\mu_1)\}$ is ε and the symbol δ is exactly the symbol which occurs at ε in r . This is the intuition of that ε will be called a matching path of μ_1 .

Next we consider the derivation (\dagger) and the application of μ_6 . It can be seen that the second ω written out by μ_6 occurs at the same occurrence in the tree t as the occurrence of γ_1 scanned by μ_6 in the tree s . This is also true in general. For every derivation $s' \Rightarrow_{M_b}^* q_0(t')$ of M_b , where $s' \in T_\Sigma$ and $t' \in T_\Delta$, such that μ_6 is applied in this derivation, the following holds. The symbol γ_1 scanned by μ_6 , and the second ω symbol written out by the same rule, occur at the same occurrence in the trees s' and t' , respectively.

Now, let $\lambda_1 = occ(u[\gamma(x_1)], x_1)$, $\lambda_2 = occ(v, x_1)$ and $r' = \omega\omega(x_1)$, i.e. λ_1 is the occurrence of γ_1 scanned by μ_6 in the tree s , λ_2 is the occurrence of the first ω written out by the same rule in the tree t , and r' is the tree which is written out by μ_6 . Since $\lambda_1 = 31$ and $\lambda_2 = 3$, we have that $\lambda_1 = \lambda_2 1$. This path 1 is exactly the occurrence of the second ω symbol written out by μ_6 in the tree r' . This is the intuition of that 1 will be called a matching path of μ_6 .

In these examples we gave the matching paths of certain rules of a shape preserving bottom-up tree transducer. The exact definition of matching paths is given below.

Definition 3.16 Let $M = (\hat{Q}, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $||inf(\mu)|| \neq 1$, $\tau_{M,q}$ is infinite and, for every $i \in inf(\mu)$, the variable x_i occurs exactly once in r . The *set of matching paths* of μ , denoted by $mp(\mu)$, is defined by case distinction according to the cardinality of the set $inf(\mu)$ in the following way.

Case 1: $||inf(\mu)|| = 0$. Let $mp(\mu)$ be the smallest subset of \mathbb{N}^* satisfying the following condition. If there exists a derivation

$$\begin{aligned}
 s &= u[\sigma(s_1, \dots, s_k)] \\
 &\Rightarrow_M^* u[\sigma(q_1(t_1), \dots, q_k(t_k))] \\
 &\Rightarrow_M u[q(r[t_1, \dots, t_k])] && \text{(rule } \mu) \\
 &\Rightarrow_M^* q_0(v[r[t_1, \dots, t_k]]) \\
 &= q_0(t),
 \end{aligned}$$

of M , where $s \in T_\Sigma$, $u \in \hat{T}_\Sigma(X_1)$, $v \in \hat{T}_\Delta(X_1)$, $s_1, \dots, s_k \in T_\Sigma$, $t_1, \dots, t_k \in T_\Delta$, $t \in T_\Delta$ (cf. Figure 3.3), and, for a string $\gamma \in \mathbb{N}^*$, the condition $\alpha = \beta\gamma$ or $\beta = \alpha\gamma$ holds, where $\alpha = occ(u, x_1)$, $\beta = occ(v, x_1)$, then let $\gamma \in mp(\mu)$. Moreover, if the condition $\alpha = \beta\gamma$

holds, then γ is called a *right matching path* of μ , otherwise $\beta = \alpha\gamma$ and $\gamma \neq \varepsilon$, when γ is called a *left matching path* of μ . Note that in general, γ can be both right and left matching path of μ .

Case 2: $||inf(\mu)|| > 1$. Let $mp(\mu) = \{\gamma\}$, where γ is the longest common prefix of the strings $occ(r, x_i)$ for every $i \in inf(\mu)$. \square

One can observe that we did not define the matching path of a rule μ of a bottom-up tree transducer M in case $||inf(\mu)|| = 1$. The reason of this is that we will not need to use a matching path of such a rule, neither in the definition of the frame transducer nor in those lemmas which we will use to show that the frame transducer of a shape preserving bottom-up tree transducer is also shape preserving.

In the next example we determine those rules of M_b which have matching path, and we give one matching path for every such a rule.

Example 3.17 Let M_b be the tree transducer we gave in Example 2.4. For every state $q \in \{q_0, q_1, q_2, q_3\}$ the set $\tau_{M,q}$ is infinite and the set τ_{M,q_α} is finite. Then $inf(\mu_1) = \{1, 3\}$, $inf(\mu_2) = \{2\}$, $inf(\mu_3) = \emptyset$, $inf(\mu_4) = \{1\}$, $inf(\mu_5) = \{1\}$ and $inf(\mu_6) = inf(\mu_7) = inf(\mu_8) = \emptyset$. By Definition 3.16, the set of matching paths is defined for the rules μ_1, μ_3 and μ_6 and is not defined for the remaining rules in R . (Note that the rules μ_7 and μ_8 have the state q_α on their right-hand side and τ_{M,q_α} is finite.)

We have already seen in the discussion given before Definition 3.16 that $\varepsilon \in mp(\mu_1)$ and $1 \in mp(\mu_6)$. Moreover it is easy to see that, by Case 1 of Definition 3.16, 1 is a right matching path of μ_6 . To compute a matching path of μ_3 , let $u = \sigma(x_1, \alpha, \gamma\gamma_1\alpha)$ and $v = \delta(\delta_1(\beta, x_1), \beta_1, \omega\omega\beta_2)$. Clearly there is a derivation of M_b of the following form.

$$\begin{aligned}
 & (\dagger) \\
 s &= u[\sigma_1(\alpha, \alpha)] \\
 &\Rightarrow_{M_b}^* u[\sigma_1(q_\alpha(\beta_1), q_\alpha(\beta_2))] \\
 &\Rightarrow_{M_b} u[q_1(\beta_2)] && \text{(rule } \mu_3) \\
 &\Rightarrow_{M_b}^* q_0(v[\beta_2]) \\
 &= q_0(t),
 \end{aligned}$$

where $s \in T_{\Sigma_{M_b}}$ and $t \in T_{\Delta_{M_b}}$. Let $\lambda_1 = occ(u, x_1)$ and $\lambda_2 = occ(v, x_1)$. Then $\lambda_1 = 1$, $\lambda_2 = 12$ and $\lambda_2 = \lambda_1 2$. Now, since $||inf(\mu_3)|| = 0$ and τ_{M,q_1} is infinite, by Case 1 of Definition 3.16, $2 \in mp(\mu_3)$ and 2 is a left matching path of μ_3 . \square

We showed in the discussion before Definition 3.16 that for the rule $\mu_6 \in R$, $1 \in mp(\mu_6)$. Moreover, it follows from the same discussion and from Case 1 of Definition 3.16 that even $mp(\mu_6) = \{1\}$ holds. We are going to generalize this observation, i.e. we will show the following.

Let us consider a shape preserving bottom-up tree transducer M and a rule μ of M . If μ is such that $||inf(\mu)|| = 0$ and $\tau_{M,q}$ is infinite, where q is the state occurring on the right-hand side of μ , then $||mp(\mu)|| = 1$ (note that μ_6 satisfies the conditions of this statement). To prove this we will need the following lemma which implies that either every matching path $\gamma \in mp(\mu)$ is a right matching path of μ or every matching path $\gamma \in mp(\mu)$ is a left matching path of μ .

Lemma 3.18 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\tau_{M,q}$ is infinite. Assume there are two derivations (1) and its “primed version” (2) of M as follows.

$$\begin{array}{ll}
 (1) & (2) \\
 s = u[\sigma(s_1, \dots, s_k)] & s' = u'[\sigma(s'_1, \dots, s'_k)] \\
 \Rightarrow_M^* u[\sigma(q_1(t_1), \dots, q_k(t_k))] & \Rightarrow_M^* u'[\sigma(q_1(t'_1), \dots, q_k(t'_k))] \\
 \Rightarrow_M u[q(r[t_1, \dots, t_k])] & \Rightarrow_M u'[q(r[t'_1, \dots, t'_k])] \quad (\text{rule } \mu) \\
 \Rightarrow_M^* q_0(v[r[t_1, \dots, t_k]]) & \Rightarrow_M^* q_0(v'[r[t'_1, \dots, t'_k]]) \\
 = q_0(t), & = q_0(t'),
 \end{array}$$

where $s, s' \in T_\Sigma$, $u, u' \in \widehat{T}_\Sigma(X_1)$, $v, v' \in \widehat{T}_\Delta(X_1)$, $s_1, s'_1, \dots, s_k, s'_k \in T_\Sigma$, $t_1, t'_1, \dots, t_k, t'_k \in T_\Delta$ and $t, t' \in T_\Delta$. Let $\alpha = occ(u, x_1)$, $\alpha' = occ(u', x_1)$, $\beta = occ(v, x_1)$ and $\beta' = occ(v', x_1)$.

Then either

- β is a prefix of α and β' is a prefix of α' or
- α is a proper prefix of β and α' is a proper prefix of β' .

Proof. By Lemma 3.11, α and β are comparable and α' and β' are also comparable. Now we prove the lemma by contradiction. Assume, on the contrary, that

- α is a proper prefix of β or α' is a proper prefix of β' and
- β is a prefix of α or β' is a prefix of α' .

Without loss of generality, assume that α is a proper prefix of β and β' is a prefix of α' . Let $\beta = \alpha\gamma$ and $\alpha' = \beta'\gamma'$, where $\gamma \in \mathbb{N}^+$ and $\gamma' \in \mathbb{N}^*$. Since M is shape preserving, $s' \approx t'$, which implies that $\sigma(s'_1, \dots, s'_k) \approx stree(r[t'_1, \dots, t'_k], \gamma')$. Moreover there is a derivation of M of the following form.

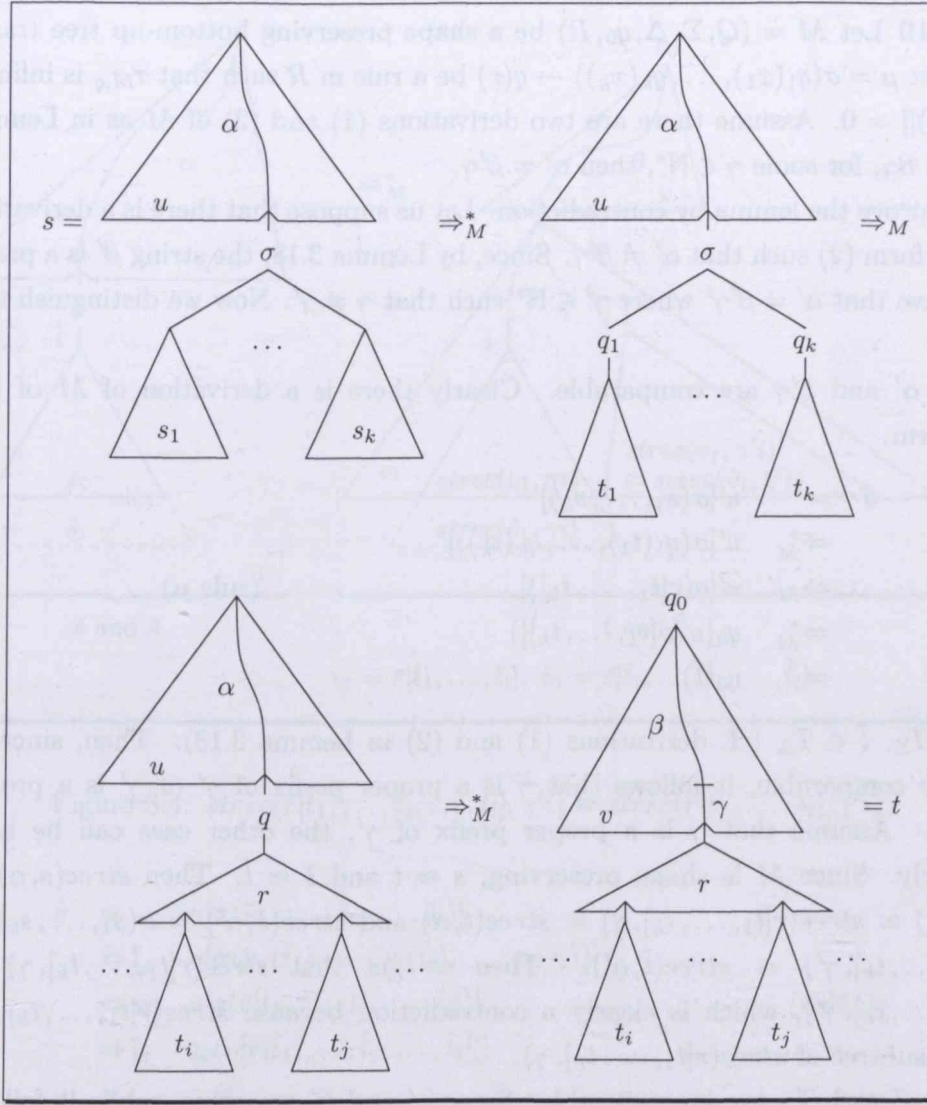


Figure 3.3: The derivation appearing in Case 1 of Definition 3.16 in the case when $\beta = \alpha\gamma$.

$$\begin{aligned}
 \tilde{s} &= u[\sigma(s'_1, \dots, s'_k)] \\
 &\Rightarrow^*_M u[\sigma(q_1(t'_1), \dots, q_k(t'_k))] \\
 &\Rightarrow_M u[q(r[t'_1, \dots, t'_k])] && \text{(rule } \mu) \\
 &\Rightarrow^*_M q_0(v[r[t'_1, \dots, t'_k]]) \\
 &= q_0(\tilde{t}),
 \end{aligned}$$

where $\tilde{s} \in T_\Sigma$ and $\tilde{t} \in T_\Delta$. Since M is shape preserving, $\tilde{s} \approx \tilde{t}$. Furthermore, since $\beta = \alpha\gamma$, it follows that $r[t'_1, \dots, t'_k] \approx \text{stree}(\sigma(s'_1, \dots, s'_k), \gamma)$, which clearly contradicts that $\sigma(s'_1, \dots, s'_k) \approx \text{stree}(r[t'_1, \dots, t'_k], \gamma')$, since $\gamma \neq \varepsilon$. ■

The following two lemmas will imply that even the condition $\|mp(\mu)\| = 1$ holds.

Let us consider a shape preserving bottom-up tree transducer M and a rule μ of M . If μ is such that $\|inf(\mu)\| = 0$ and $\tau_{M,q}$ is infinite, where q is the state occurring on the right-hand side of μ , then $\|mp(\mu)\| = 1$ (note that μ_6 satisfies the conditions of this statement). To prove this we will need the following lemma which implies that either every matching path $\gamma \in mp(\mu)$ is a right matching path of μ or every matching path $\gamma \in mp(\mu)$ is a left matching path of μ .

Lemma 3.18 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\tau_{M,q}$ is infinite. Assume there are two derivations (1) and its “primed version” (2) of M as follows.

$$\begin{array}{ll}
 (1) & (2) \\
 s = & u[\sigma(s_1, \dots, s_k)] \quad s' = \quad u'[\sigma(s'_1, \dots, s'_k)] \\
 \Rightarrow_M^* & u[\sigma(q_1(t_1), \dots, q_k(t_k))] \quad \Rightarrow_M^* \quad u'[\sigma(q_1(t'_1), \dots, q_k(t'_k))] \\
 \Rightarrow_M & u[q(r[t_1, \dots, t_k])] \quad \Rightarrow_M \quad u'[q(r[t'_1, \dots, t'_k])] \quad (\text{rule } \mu) \\
 \Rightarrow_M^* & q_0(v[r[t_1, \dots, t_k]]) \quad \Rightarrow_M^* \quad q_0(v'[r[t'_1, \dots, t'_k]]) \\
 = & q_0(t), \quad = \quad q_0(t'),
 \end{array}$$

where $s, s' \in T_\Sigma$, $u, u' \in \widehat{T}_\Sigma(X_1)$, $v, v' \in \widehat{T}_\Delta(X_1)$, $s_1, s'_1, \dots, s_k, s'_k \in T_\Sigma$, $t_1, t'_1, \dots, t_k, t'_k \in T_\Delta$ and $t, t' \in T_\Delta$. Let $\alpha = occ(u, x_1)$, $\alpha' = occ(u', x_1)$, $\beta = occ(v, x_1)$ and $\beta' = occ(v', x_1)$.

Then either

- β is a prefix of α and β' is a prefix of α' or
- α is a proper prefix of β and α' is a proper prefix of β' .

Proof. By Lemma 3.11, α and β are comparable and α' and β' are also comparable. Now we prove the lemma by contradiction. Assume, on the contrary, that

- α is a proper prefix of β or α' is a proper prefix of β' and
- β is a prefix of α or β' is a prefix of α' .

Without loss of generality, assume that α is a proper prefix of β and β' is a prefix of α' . Let $\beta = \alpha\gamma$ and $\alpha' = \beta'\gamma'$, where $\gamma \in \mathbb{N}^+$ and $\gamma' \in \mathbb{N}^*$. Since M is shape preserving, $s' \approx t'$, which implies that $\sigma(s'_1, \dots, s'_k) \approx stree(r[t'_1, \dots, t'_k], \gamma')$. Moreover there is a derivation of M of the following form.

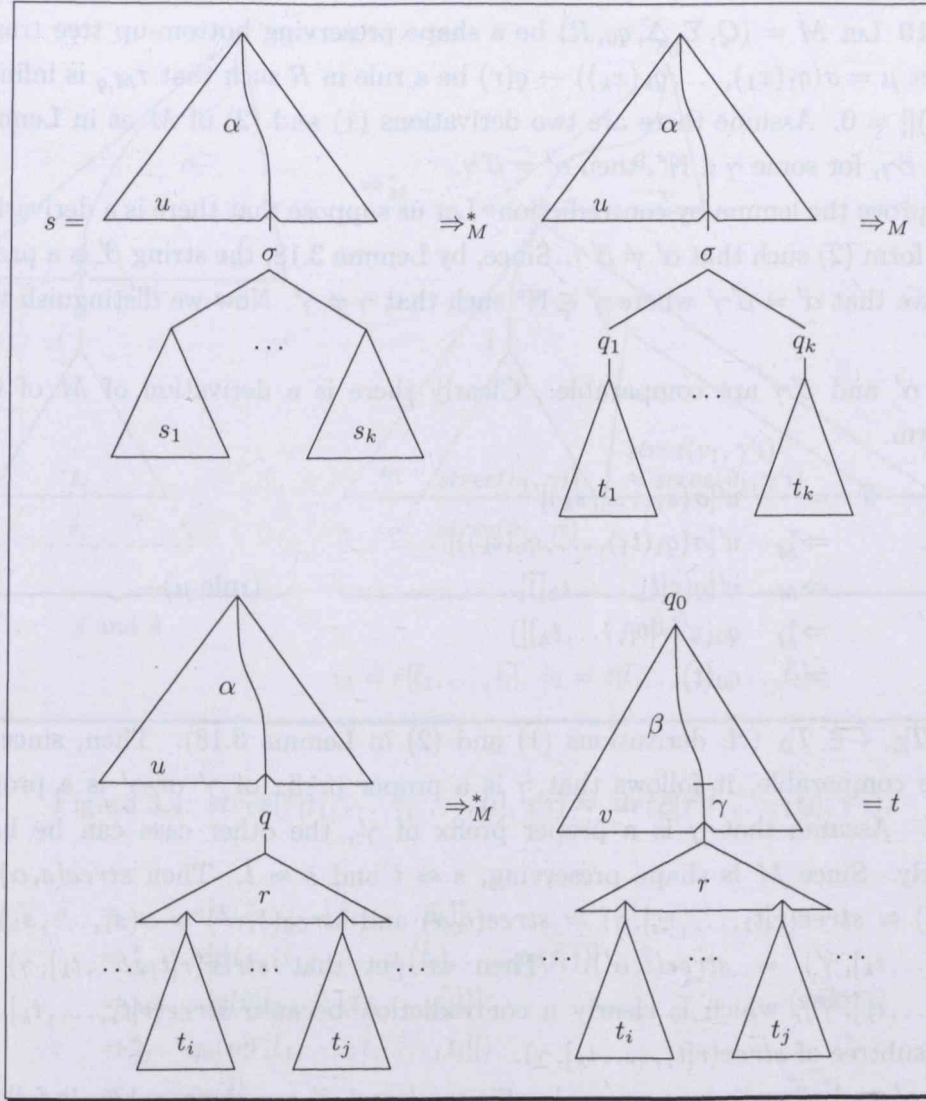


Figure 3.3: The derivation appearing in Case 1 of Definition 3.16 in the case when $\beta = \alpha\gamma$.

$$\begin{aligned}
 \tilde{s} &= u[\sigma(s'_1, \dots, s'_k)] \\
 &\Rightarrow^*_M u[\sigma(q_1(t'_1), \dots, q_k(t'_k))] \\
 &\Rightarrow_M u[q(r[t'_1, \dots, t'_k])] && \text{(rule } \mu) \\
 &\Rightarrow^*_M q_0(v[r[t'_1, \dots, t'_k]]) \\
 &= q_0(\tilde{t}),
 \end{aligned}$$

where $\tilde{s} \in T_\Sigma$ and $\tilde{t} \in T_\Delta$. Since M is shape preserving, $\tilde{s} \approx \tilde{t}$. Furthermore, since $\beta = \alpha\gamma$, it follows that $r[t'_1, \dots, t'_k] \approx stree(\sigma(s'_1, \dots, s'_k), \gamma)$, which clearly contradicts that $\sigma(s'_1, \dots, s'_k) \approx stree(r[t'_1, \dots, t'_k], \gamma')$, since $\gamma \neq \gamma'$. ■

The following two lemmas will imply that even the condition $\|mp(\mu)\| = 1$ holds.

Lemma 3.19 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\tau_{M,q}$ is infinite and $||\inf(\mu)|| = 0$. Assume there are two derivations (1) and (2) of M as in Lemma 3.18. If $\alpha = \beta\gamma$, for some $\gamma \in \mathbb{N}^*$, then $\alpha' = \beta'\gamma$.

Proof. We prove the lemma by contradiction. Let us suppose that there is a derivation of M of the form (2) such that $\alpha' \neq \beta'\gamma$. Since, by Lemma 3.18, the string β' is a prefix of α' , we have that $\alpha' = \beta'\gamma'$ where $\gamma' \in \mathbb{N}^*$ such that $\gamma \neq \gamma'$. Now we distinguish two cases.

Case 1: α' and $\beta'\gamma$ are comparable. Clearly there is a derivation of M of the following form.

$$\begin{aligned}
 \tilde{s} &= u'[\sigma(s_1, \dots, s_k)] \\
 &\Rightarrow_M^* u'[\sigma(q_1(t_1), \dots, q_k(t_k))] \\
 &\Rightarrow_M u'[q(r[t_1, \dots, t_k])] && \text{(rule } \mu) \\
 &\Rightarrow_M^* q_0(v'[r[t_1, \dots, t_k]]) \\
 &= q_0(\tilde{t}),
 \end{aligned}$$

where $\tilde{s} \in T_\Sigma$, $\tilde{t} \in T_\Delta$ (cf. derivations (1) and (2) in Lemma 3.18). Then, since α' and $\beta'\gamma$ are comparable, it follows that γ is a proper prefix of γ' or γ' is a proper prefix of γ . Assume that γ is a proper prefix of γ' , the other case can be handled similarly. Since M is shape preserving, $s \approx t$ and $\tilde{s} \approx \tilde{t}$. Then $\text{stree}(s, \alpha) = \sigma(s_1, \dots, s_k) \approx \text{stree}(r[t_1, \dots, t_k], \gamma) = \text{stree}(t, \alpha)$ and $\text{stree}(\tilde{s}, \alpha') = \sigma(s_1, \dots, s_k) \approx \text{stree}(r[t_1, \dots, t_k], \gamma') = \text{stree}(\tilde{t}, \alpha')$. Then we get that $\text{stree}(r[t_1, \dots, t_k], \gamma) \approx \text{stree}(r[t_1, \dots, t_k], \gamma')$, which is clearly a contradiction because $\text{stree}(r[t_1, \dots, t_k], \gamma')$ is a proper subtree of $\text{stree}(r[t_1, \dots, t_k], \gamma)$.

Case 2: α' and $\beta'\gamma$ are incomparable. Since α' and β' are comparable, it follows that γ and γ' are incomparable. Moreover, since $\tau_{M,q}$ is infinite and $||\inf(\mu)|| = 0$, there is a rule $\bar{\mu} = \bar{\sigma}(p_1(x_1), \dots, p_l(x_l)) \rightarrow q(\bar{r})$ in R such that $||\inf(\bar{\mu})|| > 0$. Then there are derivations of M of the following forms.

$$\begin{aligned}
 \bar{s} &= u[\bar{\sigma}(\bar{s}_1, \dots, \bar{s}_l)] & \bar{s}' &= u'[\bar{\sigma}(\bar{s}_1, \dots, \bar{s}_l)] \\
 &\Rightarrow_M^* u[\bar{\sigma}(p_1(\bar{t}_1), \dots, p_l(\bar{t}_l))] & &\Rightarrow_M^* u'[\bar{\sigma}(p_1(\bar{t}_1), \dots, p_l(\bar{t}_l))] \\
 &\Rightarrow_M u[q(\bar{r}[\bar{t}_1, \dots, \bar{t}_l])] & &\Rightarrow_M u'[q(\bar{r}[\bar{t}_1, \dots, \bar{t}_l])] && \text{(rule } \bar{\mu}) \\
 &\Rightarrow_M^* q_0(v[\bar{r}[\bar{t}_1, \dots, \bar{t}_l]]) & &\Rightarrow_M^* q_0(v'[\bar{r}[\bar{t}_1, \dots, \bar{t}_l]]) \\
 &= q_0(\bar{t}), & &= q_0(\bar{t}'),
 \end{aligned}$$

where $\bar{s}, \bar{s}_1, \dots, \bar{s}_l, \bar{s}' \in T_\Sigma$, $\bar{t}, \bar{t}_1, \dots, \bar{t}_l, \bar{t}' \in T_\Delta$ and u, u', v, v' are the same as in derivations (1) and (2). Since M is shape preserving, $\bar{s} \approx \bar{t}$ and $\bar{s}' \approx \bar{t}'$. Let $i \in \inf(\bar{\mu})$. Since τ_{M,p_i} is infinite, there is a tree $\hat{s}_i \in \text{dom}(\tau_{M,p_i})$ such that $\bar{s}_i \not\approx \hat{s}_i$. Let $\hat{t}_i \in \text{ran}(\tau_{M,p_i})$ such that $\hat{s}_i \Rightarrow_M^* p_i(\hat{t}_i)$. Let us form the input tree $\hat{s} = u[\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l)]$. Then there is a derivation of M of the following form.

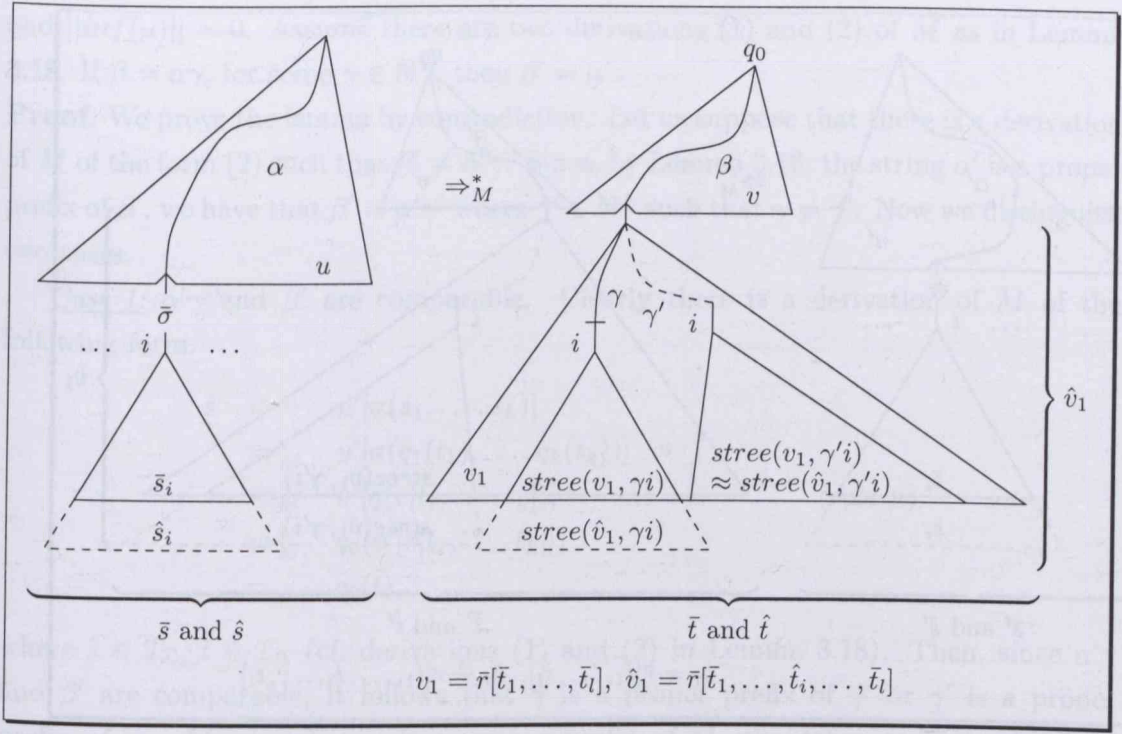


Figure 3.4: $stree(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l], \gamma'i) \approx stree(\bar{r}[\bar{t}_1, \dots, \bar{t}_l], \gamma'i)$.

$$\begin{aligned}
 \hat{s} &= u[\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l)] \\
 &\Rightarrow_M^* u[\bar{\sigma}(p_1(\bar{t}_1), \dots, p_i(\hat{t}_i), \dots, p_l(\bar{t}_l))] \\
 &\Rightarrow_M u[q(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l])] \quad (\text{rule } \bar{\mu}) \\
 &\Rightarrow_M^* q_0(v[\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l]]) \\
 &= q_0(\hat{t}),
 \end{aligned}$$

where $\hat{t} \in T_\Delta$. Since M is shape preserving, $\hat{s} \approx \hat{t}$. Moreover, since γ and γ' are incomparable, we get the following condition (cf. Figure 3.4)

$$\begin{aligned}
 stree(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l], \gamma'i) &= stree(\hat{t}, \beta\gamma'i) \approx stree(\hat{s}, \beta\gamma'i) \\
 &= stree(\bar{s}, \beta\gamma'i) \approx stree(\bar{t}, \beta\gamma'i) = stree(\bar{r}[\bar{t}_1, \dots, \bar{t}_l], \gamma'i),
 \end{aligned}$$

hence

$$stree(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l], \gamma'i) \approx stree(\bar{r}[\bar{t}_1, \dots, \bar{t}_l], \gamma'i). \quad (\dagger)$$

Now let $\hat{s}' = u'[\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l)]$. Then there is a derivation of M of the following form.

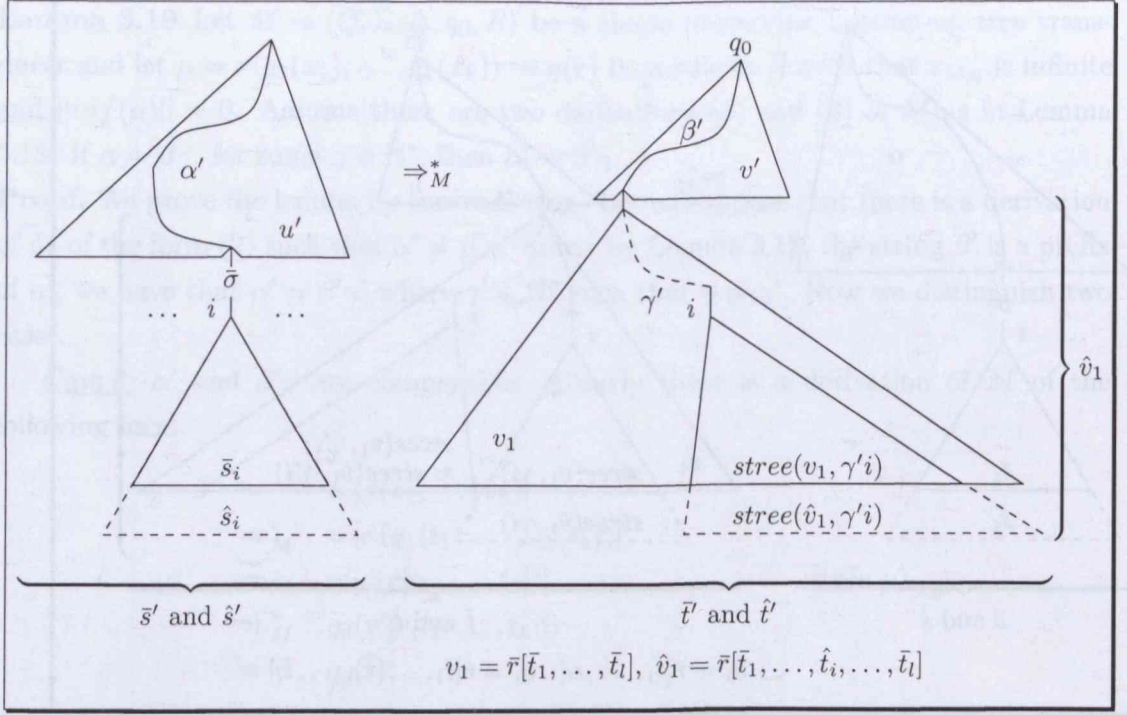


Figure 3.5: $stree(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l], \gamma'i) \not\approx stree(\bar{r}[\bar{t}_1, \dots, \bar{t}_l], \gamma'i)$.

$$\begin{aligned}
 \hat{s}' &= u'[\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l)] \\
 &\Rightarrow_M^* u'[\bar{\sigma}(p_1(\bar{t}_1), \dots, p_i(\hat{t}_i), \dots, p_l(\bar{t}_l))] \\
 &\Rightarrow_M u'[q(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l])] \quad (\text{rule } \bar{\mu}) \\
 &\Rightarrow_M^* q_0(v'[\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l]]) \\
 &= q_0(\hat{t}'),
 \end{aligned}$$

where $\hat{t}' \in T_\Delta$. Since M is shape preserving, $\hat{s}' \approx \hat{t}'$. Then we get the following condition (cf. Figure 3.5)

$$\begin{aligned}
 stree(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l], \gamma'i) &= stree(\hat{t}', \beta'\gamma'i) \approx stree(\hat{s}', \beta'\gamma'i) = \hat{s}_i \\
 &\not\approx \bar{s}_i = stree(\bar{s}', \beta'\gamma'i) \approx stree(\bar{t}', \beta'\gamma'i) = stree(\bar{r}[\bar{t}_1, \dots, \bar{t}_l], \gamma'i),
 \end{aligned}$$

hence

$$stree(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l], \gamma'i) \not\approx stree(\bar{r}[\bar{t}_1, \dots, \bar{t}_l], \gamma'i).$$

This contradicts to (\dagger) , which proves our lemma. ■

The statement and the proof of the next lemma is similar to that of the previous one.

Lemma 3.20 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\tau_{M,q}$ is infinite

and $||inf(\mu)|| = 0$. Assume there are two derivations (1) and (2) of M as in Lemma 3.18. If $\beta = \alpha\gamma$, for some $\gamma \in \mathbb{N}^+$, then $\beta' = \alpha'\gamma$.

Proof. We prove the lemma by contradiction. Let us suppose that there is a derivation of M of the form (2) such that $\beta' \neq \alpha'\gamma$. Since, by Lemma 3.18, the string α' is a proper prefix of β' , we have that $\beta' = \alpha'\gamma'$ where $\gamma' \in \mathbb{N}^+$ such that $\gamma \neq \gamma'$. Now we distinguish two cases.

Case 1: $\alpha'\gamma$ and β' are comparable. Clearly there is a derivation of M of the following form.

$$\begin{aligned}
 \tilde{s} &= u'[\sigma(s_1, \dots, s_k)] \\
 &\Rightarrow_M^* u'[\sigma(q_1(t_1), \dots, q_k(t_k))] \\
 &\Rightarrow_M u'[q(r[t_1, \dots, t_k])] && \text{(rule } \mu) \\
 &\Rightarrow_M^* q_0(v'[r[t_1, \dots, t_k]]) \\
 &= q_0(\tilde{t}),
 \end{aligned}$$

where $\tilde{s} \in T_\Sigma$, $\tilde{t} \in T_\Delta$ (cf. derivations (1) and (2) in Lemma 3.18). Then, since $\alpha'\gamma$ and β' are comparable, it follows that γ is a proper prefix of γ' or γ' is a proper prefix of γ . Assume that γ is a proper prefix of γ' , the other case can be handled similarly. Since M is shape preserving, $s \approx t$ and $\tilde{s} \approx \tilde{t}$. Then $stree(t, \beta) = r[t_1, \dots, t_k] \approx stree(\sigma(s_1, \dots, s_k), \gamma) = stree(s, \beta)$ and $stree(\tilde{t}, \beta') = r[t_1, \dots, t_k] \approx stree(\sigma(s_1, \dots, s_k), \gamma') = stree(\tilde{s}, \beta')$. Then we get that $stree(\sigma(s_1, \dots, s_k), \gamma) \approx stree(\sigma(s_1, \dots, s_k), \gamma')$, which is clearly a contradiction because $stree(\sigma(s_1, \dots, s_k), \gamma')$ is a proper subtree of $stree(\sigma(s_1, \dots, s_k), \gamma)$.

Case 2: $\alpha'\gamma$ and β' are incomparable. Since α' and β' are comparable, it follows that γ and γ' are incomparable. Moreover, since $\tau_{M,q}$ is infinite and $||inf(\mu)|| = 0$, there is a rule $\bar{\mu} = \bar{\sigma}(p_1(x_1), \dots, p_l(x_l)) \rightarrow q(\bar{r})$ in R such that $||inf(\bar{\mu})|| > 0$. Now let us consider the following derivations of M .

$$\begin{aligned}
 \bar{s} &= u[\bar{\sigma}(\bar{s}_1, \dots, \bar{s}_l)] & \bar{s}' &= u'[\bar{\sigma}(\bar{s}_1, \dots, \bar{s}_l)] \\
 &\Rightarrow_M^* u[\bar{\sigma}(p_1(\bar{t}_1), \dots, p_l(\bar{t}_l))] & &\Rightarrow_M^* u'[\bar{\sigma}(p_1(\bar{t}_1), \dots, p_l(\bar{t}_l))] \\
 &\Rightarrow_M u[q(\bar{r}[\bar{t}_1, \dots, \bar{t}_l])] & &\Rightarrow_M u'[q(\bar{r}[\bar{t}_1, \dots, \bar{t}_l])] && \text{(rule } \bar{\mu}) \\
 &\Rightarrow_M^* q_0(v[\bar{r}[\bar{t}_1, \dots, \bar{t}_l]]) & &\Rightarrow_M^* q_0(v'[\bar{r}[\bar{t}_1, \dots, \bar{t}_l]]) \\
 &= q_0(\bar{t}), & &= q_0(\bar{t}'),
 \end{aligned}$$

where $\bar{s}, \bar{s}_1, \dots, \bar{s}_l, \bar{s}' \in T_\Sigma$, $\bar{t}, \bar{t}_1, \dots, \bar{t}_l, \bar{t}' \in T_\Delta$ and u, u', v, v' are the same as in derivations (1) and (2). Since M is shape preserving, we have that $\bar{s} \approx \bar{t}$ and $\bar{s}' \approx \bar{t}'$. Let $i \in inf(\bar{\mu})$. Then τ_{M,p_i} is infinite, and by Lemma 3.10, $ran(\tau_{M,p_i})$ is also infinite. Moreover, it follows from Corollary 3.15, that the variable x_i occurs in the tree \bar{r} . Let $\hat{t}_i \in ran(\tau_{M,p_i})$ such that $\hat{t}_i \not\approx \bar{t}_i$. Then clearly $\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l] \not\approx \bar{r}[\bar{t}_1, \dots, \bar{t}_l]$. Let $\hat{s}_i \in dom(\tau_{M,p_i})$ such that $\hat{s}_i \Rightarrow_M^* p_i(\hat{t}_i)$. Let us form the input tree $\hat{s} = u[\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l)]$. Then there is a derivation of M of the following form.

$$\begin{aligned}
\hat{s} &= u[\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l)] \\
&\Rightarrow_M^* u[\bar{\sigma}(p_1(\bar{t}_1), \dots, p_i(\hat{t}_i), \dots, p_l(\bar{t}_l))] \\
&\Rightarrow_M u[q(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l])] & (\text{rule } \bar{\mu}) \\
&\Rightarrow_M^* q_0(v[\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l]]) \\
&= q_0(\hat{t}),
\end{aligned}$$

where $\hat{t} \in T_\Delta$. Since M is shape preserving, $\hat{s} \approx \hat{t}$. Moreover, since γ and γ' are incomparable, we obtain that

$$\begin{aligned}
stree(\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l), \gamma') &= stree(\hat{s}, \alpha\gamma') \approx stree(\hat{t}, \alpha\gamma') \\
&= stree(\bar{t}, \alpha\gamma') \approx stree(\bar{s}, \alpha\gamma') = stree(\bar{\sigma}(\bar{s}_1, \dots, \bar{s}_l), \gamma').
\end{aligned}$$

Now let $\hat{s}' = u'[\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l)]$. Then there is a derivation

$$\begin{aligned}
\hat{s}' &= u'[\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l)] \\
&\Rightarrow_M^* u'[\bar{\sigma}(p_1(\bar{t}_1), \dots, p_i(\hat{t}_i), \dots, p_l(\bar{t}_l))] \\
&\Rightarrow_M u'[q(\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l])] & (\text{rule } \bar{\mu}) \\
&\Rightarrow_M^* q_0(v'[\bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l]]) \\
&= q_0(\hat{t}'),
\end{aligned}$$

of M , where $\hat{t}' \in T_\Delta$. Since M is shape preserving, $\hat{s}' \approx \hat{t}'$. Then

$$\begin{aligned}
stree(\bar{\sigma}(\bar{s}_1, \dots, \hat{s}_i, \dots, \bar{s}_l), \gamma') &= stree(\hat{s}', \alpha'\gamma') \approx stree(\hat{t}', \alpha'\gamma') = \bar{r}[\bar{t}_1, \dots, \hat{t}_i, \dots, \bar{t}_l] \\
&\not\approx \bar{r}[\bar{t}_1, \dots, \bar{t}_l] = stree(\bar{t}', \alpha'\gamma') \approx stree(\bar{s}', \alpha'\gamma') = stree(\bar{\sigma}(\bar{s}_1, \dots, \bar{s}_l), \gamma').
\end{aligned}$$

This is a contradiction, which proves our lemma. ■

Now, we can state the following.

Corollary 3.21 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\tau_{M,q}$ is infinite and $\|inf(\mu)\| \neq 1$. Then $\|mp(\mu)\| = 1$.

Proof. If $\|inf(\mu)\| = 0$, then the statement holds by Lemmas 3.19 and 3.20. Otherwise, the statement holds by Case 2 of Definition 3.16. ■

In the rest of the Thesis, if, for a rule μ of a bottom-up tree transducer M , the condition $\|mp(\mu)\| = 1$ holds, we write $\gamma = mp(\mu)$ instead of $\gamma \in mp(\mu)$.

Now let us consider a rule μ of a shape preserving bottom-up tree transducer such that $\|inf(\mu)\| > 1$. In the next two lemmas we give certain properties of μ which we will use later. First we show an important role of the matching path of μ .

Lemma 3.22 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\|inf(\mu)\| > 1$. Let $\gamma = mp(\mu)$ and $\delta \in \Delta$ be the root of the tree $stree(r, \gamma)$. (Note, that, by Corollary 3.15, $stree(r, \gamma)$ contains at least one output symbol, i.e., $stree(r, \gamma) \notin X_k$.) Assume that there is a derivation (1) of M as in Lemma 3.18. Then the symbol σ in s and the symbol δ in t match each other. More formally, the condition $\alpha = \beta\gamma$ holds, where $\alpha = occ(u, x_1)$ and $\beta = occ(v, x_1)$.

Proof. We prove the lemma by contradiction. Let us suppose that $\alpha \neq \beta\gamma$. It follows from Corollary 3.15 that, for every $i \in inf(\mu)$, the variable x_i occurs exactly once in the tree r . Let, for every $i \in inf(\mu)$, $\gamma_i = occ(r, x_i)$. Then, by Definition 3.16, for every $i \in inf(\mu)$, γ is a prefix of γ_i . Moreover, τ_{M, q_i} is infinite which, by Lemma 3.11, implies that αi and $\beta\gamma_i$ are comparable. Now we distinguish the following three cases.

Case 1: The occurrences α and $\beta\gamma$ are incomparable. Let $i \in inf(\mu)$. Since αi and $\beta\gamma_i$ are comparable and γ is a prefix of γ_i , it follows that α and $\beta\gamma$ are comparable, which is a contradiction.

Case 2: The occurrence α is a proper prefix of the occurrence $\beta\gamma$. Let $\beta\gamma = \alpha\alpha'$, for some $\alpha' \in \mathbb{N}^+$, and let $j = \alpha'(1)$. Since $\|inf(\mu)\| > 1$, there is an index $i \in inf(\mu)$ such that $i \neq j$. Then clearly αi and $\beta\gamma$ are incomparable. On the other hand, the occurrences αi and $\beta\gamma_i$ are comparable. This is again a contradiction, since γ is a prefix of γ_i .

Case 3: The occurrence $\beta\gamma$ is a proper prefix of the occurrence α . Let $i, j \in inf(\mu)$ such that γ_i and γ_j are not comparable, and γ is the longest occurrence in \mathbb{N}^* such that γ is a prefix of γ_i and γ_j . Since $\beta\gamma$ is a proper prefix of α , it follows that either (i) α and $\beta\gamma_i$ are incomparable or (ii) α and $\beta\gamma_j$ are incomparable. In both cases we have a contradiction to the fact that, for every $k \in inf(\mu)$, α and $\beta\gamma_k$ are comparable. ■

Next we show that the following also holds. For every $i \in inf(\mu)$, the variable x_i occurs in the subtree $stree(r, \gamma i)$, where $\gamma = mp(\mu)$ and r is the tree which occurs on the right-hand side of μ .

Lemma 3.23 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer, $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ a rule in R such that $\|inf(\mu)\| > 1$, and $\gamma = mp(\mu)$. Then, for every $i \in inf(\mu)$, γi is a prefix of γ_i , where $\gamma_i = occ(r, x_i)$ (by Corollary 3.15, for every $i \in inf(\mu)$, x_i occurs exactly once in r).

Proof. Let $i \in inf(\mu)$. Since μ is useful, there is a derivation (1) of M as in Lemma 3.18. Note that, by Lemma 3.13, we can assume that $v \in \widehat{T}_\Delta(X_1)$. Let $\alpha = occ(u, x_1)$ and $\beta = occ(v, x_1)$. By Lemma 3.22, $\alpha = \beta\gamma$. We show that γi is a prefix of γ_i , by contradiction. Clearly γ_i is not a proper prefix of γi (cf. Case 2 of Definition 3.16), so let us suppose that γ_i and γi are incomparable. Since τ_{M, q_i} is infinite, it follows from

Lemma 3.10, that $\text{ran}(\tau_{M,q_i})$ is also infinite. Then there is a tree $\bar{t}_i \in \text{ran}(\tau_{M,q_i})$ such that $t_i \not\approx \bar{t}_i$. Let $\bar{s}_i \in \text{dom}(\tau_{M,q_i})$ such that $\bar{s}_i \Rightarrow_M^* q_i(\bar{t}_i)$. Let us form the input tree $\bar{s} = u[\sigma(s_1, \dots, \bar{s}_i, \dots, s_k)]$. Then there is a derivation of M of the following form.

$$\begin{aligned}
 \bar{s} &= u[\sigma(s_1, \dots, \bar{s}_i, \dots, s_k)] \\
 &\Rightarrow_M^* u[\sigma(q_1(t_1), \dots, q_i(\bar{t}_i), \dots, q_k(t_k))] \\
 &\Rightarrow_M u[q(r[t_1, \dots, \bar{t}_i, \dots, t_k])] && (\text{rule } \mu) \\
 &\Rightarrow_M^* q_0(v[r[t_1, \dots, \bar{t}_i, \dots, t_k]]) \\
 &= q_0(\bar{t}),
 \end{aligned}$$

where $\bar{t} \in T_\Delta$. Since M is shape preserving, $\bar{s} \approx \bar{t}$. Moreover, since γ_i and γ_i are incomparable and $\alpha_i = \beta\gamma_i$, we have that α_i and $\beta\gamma_i$ are incomparable. Then we get that $\bar{t}_i = \text{stree}(\bar{t}, \beta\gamma_i) \approx \text{stree}(\bar{s}, \beta\gamma_i) = \text{stree}(s, \beta\gamma_i) \approx \text{stree}(t, \beta\gamma_i) = t_i$, which contradicts that $t_i \not\approx \bar{t}_i$. Therefore we have proved that γ_i is a prefix of γ_i . ■

We close this subsection with summarizing our results concerning the matching path of certain rules of a shape preserving bottom-up tree transducer. Therefore let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer, and let us consider a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R such that $\|\text{inf}(\mu)\| \neq 1$ and $\tau_{M,q}$ is infinite. Moreover, let us given a derivation

$$\begin{aligned}
 s = u[\sigma(s_1, \dots, s_k)] &\Rightarrow^* u[\sigma(q_1(t_1), \dots, q_k(t_k))] \Rightarrow \\
 &u[q(r[t_1, \dots, t_k])] \Rightarrow^* q_0(v[r[t_1, \dots, t_k]]) = t
 \end{aligned}$$

of M , where $s \in T_\Sigma$, $u \in \widehat{T}_\Sigma(X_1)$, $v \in \widehat{T}_\Delta(X_1)$, $s_1, \dots, s_k \in T_\Sigma$, $t_1, \dots, t_k \in T_\Delta$ and $t \in T_\Delta$ (cf. Figure 3.3). Since M is shape preserving, $s \approx t$. Let $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$. Now, if $\gamma = \text{mp}(\mu)$, then, roughly speaking, the following holds. According to the cardinality of $\|\text{inf}(\mu)\|$ there are the following two cases.

Case 1: $\|\text{inf}(\mu)\| > 1$. Then, by Lemma 3.22, $\alpha = \beta\gamma$ and thus the trees $\text{stree}(s, \alpha) = \sigma(s_1, \dots, s_k)$ and $\text{stree}(t, \beta\gamma) = \text{stree}(r[t_1, \dots, t_k], \gamma)$ have the same shape and occur at the same occurrence in the trees s and t , respectively. Moreover, by Corollary 3.15 and Lemma 3.23, for every $i \in \text{inf}(\mu)$, the variable x_i occurs once in r , namely in the i th subtree of the root of $\text{stree}(r, \gamma)$.

Case 2: $\|\text{inf}(\mu)\| = 0$. Then by Lemma 3.18, (i) either γ is a right matching path of μ , or (ii) γ is a left matching path of μ . Moreover, by Case 1 of Definition 3.16 and by Lemmas 3.19 and 3.20 the following holds. In Case (i) $\alpha = \beta\gamma$, while in Case (ii) $\alpha\gamma = \beta$. Thus in Case (i) the trees $\text{stree}(s, \alpha) = \sigma(s_1, \dots, s_k)$ and $\text{stree}(t, \beta\gamma) = \text{stree}(r[t_1, \dots, t_k], \gamma)$, while in Case (ii) the trees $\text{stree}(s, \alpha\gamma) = \text{stree}(\sigma(s_1, \dots, s_k), \gamma)$ and $\text{stree}(t, \beta) = r[t_1, \dots, t_k]$ have the same shape and occur at the same occurrence in s and t , respectively.

3.2 Transforming Tree Transducers into Quasirelabelings

Here we transform tree transducers into quasirelabelings. With these quasirelabelings our work will be easier since shape preserving quasirelabelings have the following nice property. Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving top-down quasirelabeling tree transducer and consider a rule $\mu = q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1q_1(x_1), \dots, \gamma_kq_k(x_k))$ in R such that $k \neq 1$ and a derivation $q_0(s) \Rightarrow_M^* t$, where $s \in T_\Sigma$ and $t \in T_\Delta$, in which the rule μ is applied. If $\alpha \in \text{occ}(s)$ and $\beta \in \text{occ}(t)$ are the occurrences of the non-unary symbols σ and δ scanned and written out by μ , respectively, then $\alpha = \beta$. Of course, a similar discussion is also true when M is a shape preserving bottom-up quasirelabeling tree transducer.

First we consider top-down tree transducers.

3.2.1 Transforming Shape Preserving Top-Down Tree Transducers into Equivalent Quasirelabelings

In this subsection we develop a procedure which eliminates the permutations from the rules of a shape preserving top-down tree transducer $M = (Q, \Sigma, \Delta, q_0, R)$. As we have seen in Subsection 3.1.1, M is a permutation quasirelabeling. To construct an equivalent quasirelabeling to M , we need the following preparation.

Definition 3.24 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a permutation quasirelabeling and let

$$\mu = q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1q_1(x_{\pi(1)}), \dots, \gamma_kq_k(x_{\pi(k)}))$$

be a rule in R , where $k > 1$, $\delta \in \Delta^{(k)}$, $\gamma, \gamma_1, \dots, \gamma_k \in (\Delta^{(1)})^*$ and $\pi : [k] \rightarrow [k]$ is a permutation. The *permutation degree* of μ is the number of indexes $1 \leq i \leq k$ for which $\pi(i) \neq i$. A rule with permutation degree greater than one is called a *permutation rule*. Moreover, a state $q \in Q$ is a *permutation state* if there is a permutation rule of the above form. The permutation degree of M is the sum of the permutation degrees of its rules of the above form. (Note that the permutation degree of a quasirelabeling and thus of a relabeling is zero.)

Finally we define the binary relation \prec over Q in the following way: for every $p, q \in Q$, let $q \prec p$ if and only if there exist $u, u' \in \hat{T}_\Sigma(X_1)$ and $v, v' \in \hat{T}_\Delta(X_1)$, such that the following conditions hold:

- $q_0(u[u']) \Rightarrow_M^* v[p(u')] \Rightarrow_M^+ v[v'[q(x_1)]]$,
- $\text{occ}(u, x_1)$ and $\text{occ}(v, x_1)$ are comparable,
- $\text{occ}(u[u'], x_1)$ and $\text{occ}(v[v'], x_1)$ are incomparable.

Note that u may be x_1 , however u' cannot be x_1 in this definition of \prec . \square

Intuitively, for two states $p, q \in Q$, the condition $q \prec p$ holds if there are $u, u' \in \widehat{T}_\Sigma(X_1)$ and $v, v' \in \widehat{T}_\Delta(X_1)$ and a derivation $q_0(u) \Rightarrow_M^* v[p(x_1)]$ such that no piece of the path $\alpha = occ(u, x_1)$ takes part in a permutation during that derivation, moreover, there is a derivation $p(u') \Rightarrow_M^+ v'[q(x_1)]$ such that a piece of the path $\beta = occ(u', x_1)$ takes part in a permutation during that derivation. (Note that in $q_0(u) \Rightarrow_M^* v[p(x_1)]$ a permutation rule might be applied on α , which however did not move the involved piece of α . Moreover, a permutation rule was applied in $p(u') \Rightarrow_M^+ v'[q(x_1)]$ on the path β which did move the involved piece of β .)

Let us see now how the relation \prec looks like in the case of the top-down tree transducer M_t of Example 2.3.

Example 3.25 Let M_t be the top-down tree transducer appearing in Example 2.3. As we have seen M_t is a permutation quasirelabeling. For instance $q_3 \prec q_1$ because with $u = \gamma(x_1)$, $u' = \sigma(x_1, \gamma\alpha_2, \alpha_3)$, $v = x_1$, and $v' = \omega\delta(\omega\beta_1, x_1, \beta_3)$ we get the derivation

$$\begin{aligned} q_0(u[u']) &= q_0(\gamma(\sigma(x_1, \gamma\alpha_2, \alpha_3))) \\ &\Rightarrow_M^* q_1(\sigma(x_1, \gamma\alpha_2, \alpha_3)) \\ &\Rightarrow_M^+ \omega\delta(\omega\beta_1, q_3(x_1), \beta_3) \\ &= v[v'[q_3(x_1)]]. \end{aligned}$$

It is not hard to see that the full Hasse diagram of the relation \prec on Q is the one in Figure 3.6. The \triangleleft indicates that q_1 is a permutation state.

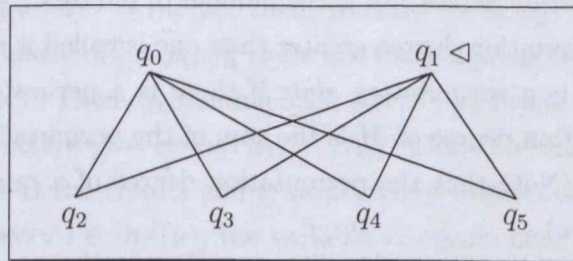


Figure 3.6: The Hasse diagram of \prec for M_t in Example 3.25.

Now, we define periodic top-down tree transducers. Later we will see that \prec is computable for periodic permutation quasirelabelings. Moreover, we will show that a shape preserving top-down tree transducer is periodic. \blacksquare

Definition 3.26 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a top-down tree transducer. We say that M is *periodic* if, for every derivation $q(u) \Rightarrow_M^* v[q(x_1)]$, where q in Q , $u \in \widehat{T}_\Sigma(X_1)$ such that $\text{length}(\text{occ}(u, x_1)) \leq ||Q||$, and $v \in \widehat{T}_\Delta(X_1)$, we have that $\text{length}(\text{occ}(u, x_1)) = \text{length}(\text{occ}(v, x_1))$.

Now we show that \prec is computable for every periodic permutation quasirelabeling.

Lemma 3.27 For every periodic permutation quasirelabeling $M = (Q, \Sigma, \Delta, q_0, R)$, the relation \prec is computable.

Proof. Let $n = ||Q||$ and $p, q \in Q$. In order to verify whether there are $u \in \widehat{T}_\Sigma(X_1)$ and $v \in \widehat{T}_\Delta(X_1)$ such that $q_0(u) \Rightarrow_M^* v[p(x_1)]$ and that no permutation happens on the path $\alpha = \text{occ}(u, x_1)$, it is sufficient to consider trees u of height at most $2n$. In fact, we may assume without loss of generality that if there is such an u , then the length of α is at most n , otherwise we can apply standard pumping arguments (cf. Lemma 10.1 in Chapter II. of [GS84], also Proposition 5.2 in [GS97]) and use the fact that M is a periodic permutation quasirelabeling. Moreover, we can assume that the length of any path which leads from a node being in α to an arbitrary leaf of u is at most n , otherwise we can apply again standard pumping arguments. Hence the height of u is at most $2n$.

In order to verify whether there are $u' \in \widehat{T}_\Sigma(X_1)$ and $v' \in \widehat{T}_\Delta(X_1)$ such that $p(u') \Rightarrow_M^* v'[q(x_1)]$ and that a permutation happens on the path $\beta = \text{occ}(u', x_1)$, it is sufficient to consider trees u' of height at most $3n$. Indeed, we may assume without loss of generality that if there is such an u' , then the length of the path from the root of u' to the node where the permutation rule was applied is at most n , otherwise we can apply again standard pumping arguments and use the fact that M is a periodic permutation quasirelabeling. Similarly, we can assume also that the length of the path from the node where the permutation rule was applied to x_1 is again at most n and that the length of any path which leads from a node being in β to an arbitrary leaf of u' is at most n . Hence the height of u' is at most $3n$.

Thus it is decidable if $q \prec p$ holds. ■

Next we show that shape preserving top-down tree transducers are periodic.

Lemma 3.28 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving top-down tree transducer. Then M is a periodic top-down tree transducer.

Proof. We prove the lemma by contradiction. Therefore assume that M is not periodic. Then, by Definition 3.26, there is a state q in Q and a derivation $q(u) \Rightarrow_M^* v[q(x_1)]$, where $u \in \widehat{T}_\Sigma(X_1)$ with $\text{length}(\text{occ}(u, x_1)) \leq ||Q||$ and $v \in \widehat{T}_\Delta(X_1)$, such that $\text{length}(\alpha) \neq \text{length}(\beta)$, where $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$.

Since q is useful, there are trees $u_1 \in \widehat{T}_\Sigma(X_1)$, $u_2 \in T_\Sigma$ such that $q_0(u_1) \Rightarrow_M^* v_1[q(x_1)]$ and $q(u_2) \Rightarrow_M^* v_2$, where $v_1 \in \widehat{T}_\Delta(X_1)$, $v_2 \in T_\Delta$. Moreover, since $\text{length}(\alpha) \neq \text{length}(\beta)$, it is easy to see that there is a number $n \in \mathbb{N}$ such that $\text{height}(s) \neq \text{height}(t)$, where $s = u_1[u^n[u_2]]$ and $t = v_1[v^n[v_2]]$. Then there is a derivation of M of the following form.

$$\begin{aligned} q_0(s) &= q_0(u_1[u^n[u_2]]) \\ &\Rightarrow_M^* v_1[q(u^n[u_2])] \\ &\Rightarrow_M^* v_1[v^n[q(u_2)]] \\ &\Rightarrow_M^* v_1[v^n[v_2]] \\ &= q_0(t). \end{aligned}$$

Since M is shape preserving, $s \approx t$, which clearly implies that $\text{height}(s) = \text{height}(t)$. However, this is a contradiction which proves the lemma. \blacksquare

By Lemma 3.28, a shape preserving permutation quasirelabeling M is periodic. Then, by Lemma 3.27, the relation \prec is computable for M . Next we prove that in this case \prec is a strict partial order.

Lemma 3.29 The relation \prec is a strict partial order for any shape preserving permutation quasirelabeling $M = (Q, \Sigma, \Delta, q_0, R)$.

Proof. We show that \prec is irreflexive and transitive. In fact, the transitivity can be proved easily by using standard arguments, hence we leave this part of the proof.

We prove the irreflexivity by contradiction. Let us suppose there is a $p \in Q$ such that $p \prec p$ holds. Then there exist $u, u' \in \widehat{T}_\Sigma(X_1)$ and $v, v' \in \widehat{T}_\Delta(X_1)$ such that

$$q_0(u[u']) \Rightarrow_M^* v[p(u')] \Rightarrow_M^* v[v'[p(x_1)]]$$

and, moreover, $\text{occ}(u, x_1)$ and $\text{occ}(v, x_1)$ are comparable but $\text{occ}(u[u'], x_1)$ and $\text{occ}(v[v'], x_1)$ are incomparable. Since $p(u') \Rightarrow^* v'[p(x_1)]$ and $u' \neq x_1$, $\text{dom}(\tau_{M,p})$ is infinite, hence not uniform. On the other hand, since clearly $\text{occ}(u[u'], x_1)$ and $\text{occ}(v[v'], x_1)$ are incomparable, by Lemma 3.6, $\text{dom}(\tau_{M,p})$ is uniform, which is a contradiction. \blacksquare

As the last preparation step we state the following rather obvious fact concerning finite tree transformations.

Lemma 3.30 Let $\rho = \{(s_1, t_1), \dots, (s_n, t_n)\} \subseteq T_\Sigma \times T_\Delta$ be finite relation and $\gamma \in (\Delta^{(1)})^*$, where Σ and Δ are ranked alphabets such that the set $\{s_1, \gamma t_1, \dots, s_n, \gamma t_n\}$ is uniform. Then there is a quasirelabeling $M = (Q, \Sigma, \Delta, q_0, R)$ such that $\tau_M = \rho$.

Proof. In case $\gamma = \varepsilon$ the statement is clear because we can construct M as the disjoint union of the relabelings M_j which compute the relations $\{(s_j, t_j)\}$. Hence, in this particular case M is a relabeling.

Now let us assume that $\gamma \in (\Delta^{(1)})^+$ with $\text{length}(\gamma) = m$. Then, for every $j \in [n]$, there are $\gamma_j \in (\Sigma^{(1)})^+$ and $u_j \in T_\Sigma$ such that $\text{length}(\gamma_j) = m$ and $\gamma_j u_j = s_j$. Obviously $\{u_1, t_1, \dots, u_n, t_n\}$ is uniform, so, by the discussion of the case $\gamma = \varepsilon$, there is a relabeling $M' = (Q', \Sigma, \Delta, q'_0, R')$ such that $\tau_{M'} = \{(u_1, t_1), \dots, (u_n, t_n)\}$.

Let q_0 and, for every $j \in [n]$, $p_{j1}, \dots, p_{j(m-1)}$ be new states. Moreover, construct the rules

$$\begin{aligned} q_0(\gamma_j(1)(x_1)) &\rightarrow p_{j1}(x_1), \\ p_{j1}(\gamma_j(2)(x_1)) &\rightarrow p_{j2}(x_1), \\ &\dots \\ p_{j(m-1)}(\gamma_j(m)(x_1)) &\rightarrow q'_0(x_1). \end{aligned}$$

(Recall that $\gamma_j(i)$ is the i th letter of γ_j . In case $m = 1$, we have the only rule $q_0(\gamma_j(1)(x_1)) \rightarrow q'_0(x_1)$.)

Now, let $M = (Q, \Sigma, \Delta, q_0, R)$, where $Q = Q' \cup \{q_0\} \cup \{p_{ji} \mid j \in [n], i \in [m-1]\}$ and let R be the set of the rules constructed above and of the rules in R' . It should be clear that M is a quasirelabeling and $\tau_M = \rho$. ■

Now we can show that every shape preserving permutation quasirelabeling is effectively equivalent to a quasirelabeling.

Lemma 3.31 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving permutation quasirelabeling. Then a quasirelabeling $M' = (Q', \Sigma, \Delta, q'_0, R')$ can be constructed such that $\tau_M = \tau_{M'}$.

Proof. If the permutation degree of M is zero, then M is a quasirelabeling thus we are ready. Otherwise, it is sufficient to show that a permutation quasirelabeling $N = (\bar{Q}, \Sigma, \Delta, \bar{q}_0, \bar{R})$ can be constructed such that $\tau_M = \tau_N$ and the permutation degree of N is less than that of M .

To see this, let us take a permutation rule

$$\mu = q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1 q_1(x_{\pi(1)}), \dots, \gamma_k q_k(x_{\pi(k)}))$$

in R , where $k > 1$, $q, q_1, \dots, q_k \in Q$, $\sigma \in \Sigma^{(k)}$, $\delta \in \Delta^{(k)}$, $\gamma, \gamma_1, \dots, \gamma_k \in (\Delta^{(1)})^*$ and $\pi : [k] \rightarrow [k]$ is a permutation, such that q is maximal among the permutation states, i.e., there is no permutation state p with $q \prec p$. (Note that by Lemmas 3.27 and 3.28 \prec is computable, and by Lemma 3.29, \prec is a strict partial order.)

Since μ is a permutation rule, there exist $n > 1$, a sequence $1 \leq i_1, \dots, i_n \leq k$ of different indexes such that $i_2 = \pi(i_1), \dots, i_n = \pi(i_{n-1})$ and $i_1 = \pi(i_n)$. Moreover,

since q is maximal among the permutation states and M is shape preserving, there is a derivation such that the occurrence of δ which is added to the output by the application of μ matches the occurrence of σ to which μ was applied. More formally, M has a derivation (\dagger) as in Lemma 3.7 such that $\text{occ}(u, x_1) = \text{occ}(v, x_1)1^m$, where $m = \text{length}(\gamma)$. Then the following statement holds.

(*) For every $j \in [n]$, a tree in $\text{dom}(\tau_{M, q_{\pi^{-1}(i_j)}})$ has the same shape as a tree in $\gamma_{i_j} \text{ran}(\tau_{M, q_{i_j}})$, see Figure 3.7. Hence $\text{dom}(\tau_{M, q_{\pi^{-1}(i_j)}}) \cup \gamma_{i_j} \text{ran}(\tau_{M, q_{i_j}})$ is uniform.

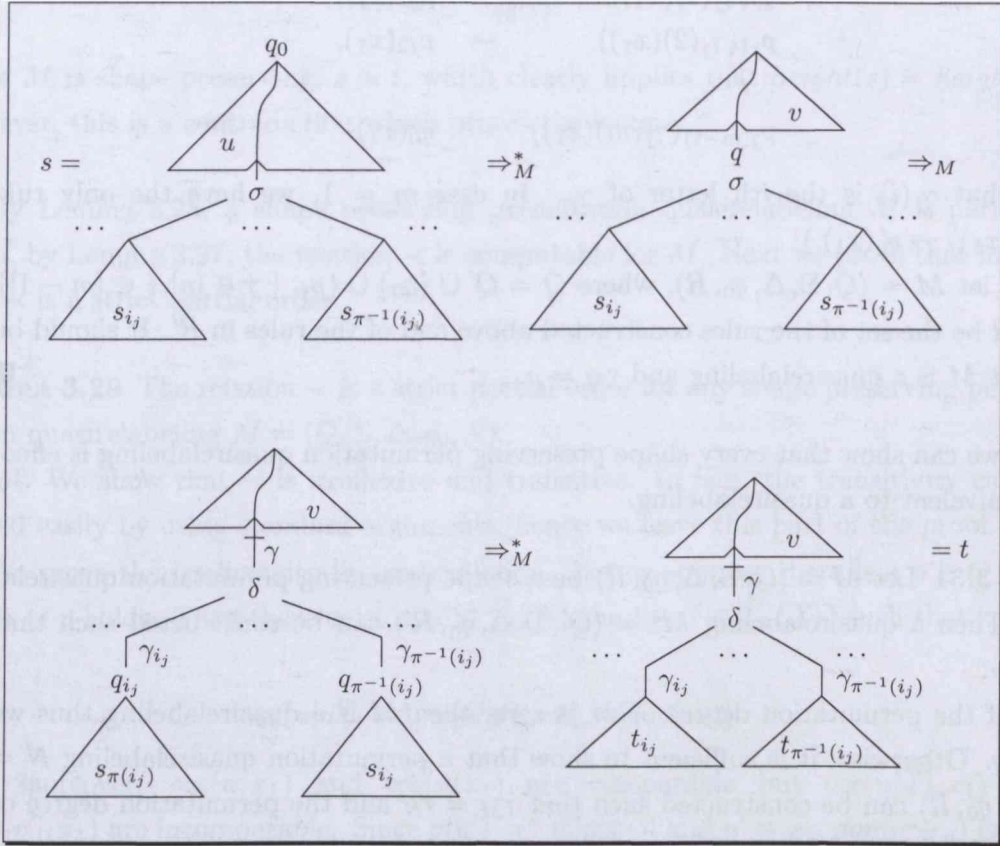


Figure 3.7: $s_{i_j} \approx \gamma_{i_j} t_{i_j}$, hence the set $\text{dom}(\tau_{M, q_{\pi^{-1}(i_j)}}) \cup \gamma_{i_j} \text{ran}(\tau_{M, q_{i_j}})$ is uniform.

Now let, for every $j \in [n]$, $M^{(j)} = (Q^{(j)}, \Sigma, \Delta, p^{(j)}, R^{(j)})$ be the quasirelabeling which computes the relation $\text{dom}(\tau_{M, q_{\pi^{-1}(i_j)}}) \times \text{ran}(\tau_{M, q_{i_j}})$. Such a quasirelabeling exists by Lemmas 3.7 and 3.30. Assume that the state sets $Q^{(j)}$ are disjoint.

Moreover, let $\bar{\mu}$ be the rule obtained from μ as follows. For every $j \in [n]$, we substitute the construct $q_{i_j}(x_{\pi(i_j)})$ by $p^{(j)}(x_{i_j})$ in the right-hand side of μ . Then the permutation degree of $\bar{\mu}$ is less than that of μ .

Now construct the top-down tree transducer $(\tilde{Q}, \Sigma, \Delta, \tilde{q}_0, \tilde{R})$, where

- $\tilde{q}_0 = q_0$,
- $\tilde{Q} = Q \cup Q^{(1)} \cup \dots \cup Q^{(n)}$,
- $\tilde{R} = (R - \{\mu\}) \cup \{\bar{\mu}\} \cup R^{(1)} \cup \dots \cup R^{(n)}$.

Then eliminate the useless rules of the top-down tree transducer $(\tilde{Q}, \Sigma, \Delta, \tilde{q}_0, \tilde{R})$ and let the resulting top-down tree transducer be $N = (\bar{Q}, \Sigma, \Delta, \bar{q}_0, \bar{R})$. Then it should be clear that the permutation degree of N is less than that of M .

Now we show that $\tau_M = \tau_N$. Let us denote by $\Rightarrow_{M \cup N}$ the derivation relation over $T_{Q \cup \bar{Q} \cup \Sigma \cup \Delta}$ in which both the rules in R and in \bar{R} can be applied.

First we show $\tau_M \subseteq \tau_N$. To see this, it is sufficient to show the following (we freely reuse the notations of the trees we have already used up to this point because this will not cause any confusion). Let $s \in T_\Sigma$ and $t \in T_\Delta$ with $q_0(s) \Rightarrow_{M \cup N}^* t$ and let a derivation sequence of t from $q_0(s)$ be given such that the rule μ is applied $K \geq 1$ times in the sequence. Then we can construct another derivation sequence from $q_0(s)$ to t such that the rule μ is applied $K - 1$ times in the steps of the second derivation sequence.

Indeed, if $q_0(s) \Rightarrow_M^* t$ such that the rule μ is applied K times in the steps of that derivation, then applying K times the construction we obtain that $q_0(s) \Rightarrow_N^* t$.

Let us take a derivation $q_0(s) \Rightarrow_{M \cup N}^* t$, in which the rule μ is applied K times. Let $\rho : [n] \rightarrow [n]$ be a permutation such that $i_{\rho(1)} < \dots < i_{\rho(n)}$. Then $q_0(s) \Rightarrow_{M \cup N}^* t$ can be written as

$$\begin{aligned}
 q_0(s) &= q_0(u[\sigma(s_1, \dots, s_k)]) \\
 &\Rightarrow_{M \cup N}^* v[q(\sigma(s_1, \dots, s_k))] \\
 &\Rightarrow_M v[\gamma\delta(\dots, \gamma_{i_{\rho(1)}} q_{i_{\rho(1)}}(s_{\pi(i_{\rho(1)})}), \dots, \gamma_{i_{\rho(n)}} q_{i_{\rho(n)}}(s_{\pi(i_{\rho(n)})}), \dots)] \\
 &\quad (\text{rule } \mu) \\
 &\Rightarrow_{M \cup N}^* v[\gamma\delta(\dots, \gamma_{i_{\rho(1)}} t_1, \dots, \gamma_{i_{\rho(n)}} t_n, \dots)] \\
 &= t,
 \end{aligned}$$

where $u \in \hat{T}_\Sigma(X_1)$, $s_1, \dots, s_k \in T_\Sigma$, $v \in \hat{T}_\Delta(X_1)$ and $t_1, \dots, t_n \in T_\Delta$. Then we have that

$$\begin{aligned}
 q_0(s) &= q_0(u[\sigma(s_1, \dots, s_k)]) \\
 &\Rightarrow_{M \cup N}^* v[q(\sigma(s_1, \dots, s_k))] \\
 &\Rightarrow_N v[\gamma\delta(\dots, \gamma_{i_{\rho(1)}} p^{(\rho(1))}(s_{i_{\rho(1)}}), \dots, \gamma_{i_{\rho(n)}} p^{(\rho(n))}(s_{i_{\rho(n)}}), \dots)] \\
 &\quad (\text{rule } \bar{\mu}) \\
 &\Rightarrow_{M \cup N}^* v[\gamma\delta(\dots, \gamma_{i_{\rho(1)}} t_1, \dots, \gamma_{i_{\rho(n)}} t_n, \dots)] \\
 &= t
 \end{aligned}$$

because, for every $j \in [n]$, $t_j \in \text{ran}(\tau_{M, q_{i_{\rho(j)}}})$ and $s_{i_{\rho(j)}} \in \text{dom}(\tau_{M, q_{\pi^{-1}(i_{\rho(j)})}})$ and

$p^{(\rho(j))}$ is the initial state of the quasirelabeling which computes the tree transformation $\text{dom}(\tau_{M, q_{\pi^{-1}(i_{\rho(j)})}}) \times \text{ran}(\tau_{M, q_{i_{\rho(j)}}})$.

The above argumentation is clearly reversible, so $\tau_N \subseteq \tau_M$ also holds. Thus, we have shown that a quasirelabeling M' can be constructed such that M' is equivalent to M . ■

3.2.2 Constructing Frame Transducers of Transformable Bottom-Up Tree Transducers

In this subsection we define the concept of transformable bottom-up tree transducers and show that every shape preserving bottom-up tree transducer M is transformable, provided that τ_M is infinite. Moreover we define the frame transducer $fr(M)$ of a transformable bottom-up tree transducer M . The transducer $fr(M)$ will work on trees over special ranked alphabets, called tree alphabets. These tree alphabets contain certain parts of trees from the sets $\text{dom}(\tau_M)$ and $\text{ran}(\tau_M)$. Formally, a part of a tree can be defined as follows. Let Σ be a ranked alphabet, $s \in T_\Sigma$ and $u \in \widehat{T}_\Sigma(X_k)$ for some $k \in \mathbb{N}$. Then we say that u is a part of s if there are trees $u' \in \widehat{T}_\Sigma(X_1)$ and $u_1, \dots, u_k \in T_\Sigma$ such that $s = u'[u_1, \dots, u_k]$. Of course in the tree alphabets these parts are considered as input and output symbols of the frame transducer. The benefit of constructing the transducer $fr(M)$ is that it computes a tree transformation which is closely related to τ_M in the sense that we can get back τ_M easily from $\tau_{fr(M)}$ with the help of certain tree homomorphisms. Moreover, $fr(M)$ has certain nice properties. For example, $fr(M)$ is a bottom-up quasirelabeling tree transducer, and if M is shape preserving, then $fr(M)$ is also shape preserving. These properties will make the proof of that every shape preserving bottom-up tree transducer is equivalent to a relabeling much easier.

First we define formally the tree alphabet mentioned above and a tree homomorphism.

Definition 3.32 Let Σ be a ranked alphabet and S be a finite subset of $\bigcup_{l=0}^k \widehat{T}_\Sigma(X_l)$ for some $k \geq 0$, such that the following conditions hold:

- $x_1 \notin S$,
- for every $s \in S$ with $s \in \widehat{T}_\Sigma(X_l)$, if $s = \sigma(s_1, \dots, s_m)$ for some $m \geq 0$, then, for every $i \in [m]$, either $s_i \in X_l$ or $s_i \in T_\Sigma$.

(Note that the above conditions assure that, for every $l \in [k]$, $x_l \notin S$ and that, for every $s \in S$ with $s \in \widehat{T}_\Sigma(X_l)$ and $s = \sigma(s_1, \dots, s_m)$, $m \geq l$ holds.)

We define the ranked alphabet, called *tree alphabet* and denoted also by S , such that, for every $l \in [k]$, $S^{(l)} = S \cap \widehat{T}_\Sigma(X_l)$. We also define the mapping $\bar{h}_S : S \cup X \rightarrow T_\Sigma(X_k)$

such that, for every $s \in S \cup X$, $\bar{h}_S(s) = s$. Then \bar{h}_S extends to a tree homomorphism $h_S : T_S(X) \rightarrow T_\Sigma(X)$ in a usual way. \square

Let us consider now a ranked alphabet Σ and a number $l \geq 1$. Next we define the decomposition of an element of $\hat{T}_\Sigma(X_l)$ satisfying certain conditions into subtrees such that the set of these subtrees forms a tree alphabet. This decomposition will be used when we define the rules and the sets of input and output symbols of $fr(M)$.

Definition 3.33 Let Σ be a ranked alphabet and let $s \in \hat{T}_\Sigma(X_l)$, where $l \geq 1$. Let γ be the longest path in $occ(s)$ such that $stree(s, \gamma) \in \hat{T}_\Sigma(X_l)$.

Assume that if $l > 1$, then the following additional condition holds. For every $i, j \in [l]$ with $i \neq j$, the longest common prefix of $occ(stree(s, \gamma), x_i)$ and $occ(stree(s, \gamma), x_j)$ is ε .

Then the *decomposition* of s , denoted by $dec(s)$, is defined, by case distinction according to l , as follows.

Case 1: $l = 1$. (Note that in this case $\gamma = occ(s, x_1)$.) If $s = x_1$, then let $dec(s) = \emptyset$. Otherwise, let $n = length(\gamma)$. Then there are trees $u_1, \dots, u_n \in \hat{T}_\Sigma(X_1)$ such that, for every $1 \leq m \leq n$, $occ(u_m, x_1) = \gamma(m)$ and $s = u_1[\dots[u_n]]$. Then let $dec(s) = \{u_1, \dots, u_n\}$.

Case 2: $l > 1$. Let $stree(s, \gamma) = \sigma(r_1, \dots, r_k)$. (Note that $k \geq l$ holds.)

Let $s = u^{(1)}[\sigma(r_1, \dots, r_k)]$ for some $u^{(1)} \in \hat{T}_\Sigma(X_1)$ such that $stree(u^{(1)}, \gamma) = x_1$. Let, for every $j \in [l]$, i_j be the index in $[k]$ such that x_j occurs in r_{i_j} .

Let $u^{(2)} = \sigma(\bar{r}_1, \dots, \bar{r}_k)$, where for every $i \in [k]$, $\bar{r}_i = x_j$ if $i = i_j$ (i.e. x_j occurs in r_i) for some $j \in [l]$, and $\bar{r}_i = r_i$ otherwise. Then let

$$dec(s) = dec(u^{(1)}) \cup dec(r_{i_1}) \cup dec(r_{i_2}[x_2 \leftarrow x_1]) \cup \dots \cup dec(r_{i_l}[x_l \leftarrow x_1]) \cup \{u^{(2)}\}.$$

\square

Next we give an example to demonstrate how this decomposition works.

Example 3.34 Let $\Sigma = \{\alpha_1^{(0)}, \alpha_2^{(0)}, \gamma_1^{(1)}, \gamma_2^{(1)}, \delta^{(2)}, \sigma^{(3)}\}$. The decomposition of the trees $s_1, s_2 \in \hat{T}_\Sigma(X_1)$ and $s_3, s_4 \in \hat{T}_\Sigma(X_2)$ can be seen in Figure 3.8. \square

In the following we define transformable bottom-up tree transducers. As we have already mentioned, if a bottom-up tree transducer M is transformable, then we can associate another bottom-up tree transducer to M which is called the frame transducer of M .

Definition 3.35 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer. We say that M is *transformable*, if τ_M is infinite and for every rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R such that $\tau_{M,q}$ is infinite and $||inf(\mu)|| \neq 1$, the following conditions hold.

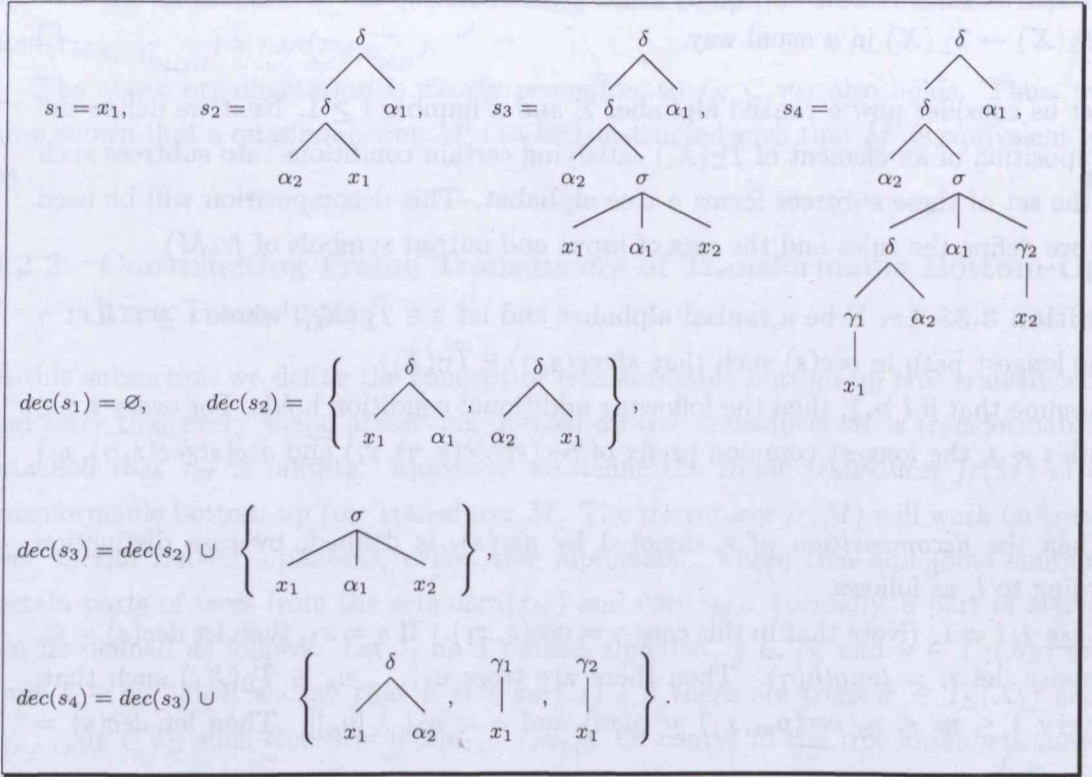


Figure 3.8: Decomposition of trees.

- (i) For every $i \in \text{inf}(\mu)$, the variable x_i occurs exactly once in r .
- (ii) $\|mp(\mu)\| = 1$.
- (iii) If $\|\text{inf}(\mu)\| = 0$, then the following holds. Let $\gamma = mp(\mu)$. Then either γ is a left matching path of μ and, for every $s \in \{\sigma(s_1, \dots, s_k) \mid \forall i \in [k] : s_i \in \text{dom}(\tau_{M,q_i})\}$, we have that $\gamma \in \text{occ}(s)$, or γ is a right matching path of μ and, for every $t \in \{r[t_1, \dots, t_k] \mid \forall i \in [k] : t_i \in \text{ran}(\tau_{M,q_i})\}$, the condition $\gamma \in \text{occ}(t)$ holds.
- (iv) If $\|\text{inf}(\mu)\| > 1$, then the following holds. Let $\gamma = mp(\mu)$. Then, for every $i, j \in \text{inf}(\mu)$ with $i < j$, we have that $\text{occ}(\text{stree}(r, \gamma), x_i)(1) < \text{occ}(\text{stree}(r, \gamma), x_j)(1)$. (Note that, by Case 2 of Definition 3.16, for every $i \in \text{inf}(\mu)$, $\text{length}(\gamma) < \text{length}(\text{occ}(r, x_i))$.)

□

Now, using the results of Subsection 3.1.2, we can easily show that every shape preserving bottom-up tree transducer M is transformable, provided that τ_M is infinite.

Lemma 3.36 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer. If τ_M is infinite, then M is transformable.

Proof. Let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\tau_{M,q}$ is infinite and $||inf(\mu)|| \neq 1$. We show that the conditions (i)-(iv) of Definition 3.35 are fulfilled as follows.

Conditions (i) and (ii) hold by Corollaries 3.15 and 3.21, respectively.

We prove that Condition (iii) also holds by contradiction. Assume that $||inf(\mu)|| = 0$ and Condition (iii) does not hold. Let $\gamma = mp(\mu)$. By Lemma 3.18, γ is either left matching path or right matching path of μ . Then either (1) γ is a left matching path of μ , and there is a tree $s \in \{\sigma(s_1, \dots, s_k) \mid \forall i \in [k] : s_i \in dom(\tau_{M,q_i})\}$ such that $\gamma \notin occ(s)$, or (2) γ is a right matching path of μ and there is a tree $t \in \{r[t_1, \dots, t_k] \mid \forall i \in [k] : t_i \in ran(\tau_{M,q_i})\}$ such that $\gamma \notin occ(t)$. Without loss of generality, assume that Case (1) holds (the other case can be handled similarly). Then $s = \sigma(s_1, \dots, s_k)$ for some $s_1, \dots, s_k \in T_\Sigma$. Furthermore, there is a derivation of M of the following form.

$$\begin{aligned}
 \bar{s} &= u[\sigma(s_1, \dots, s_k)] \\
 &\Rightarrow_M^* u[\sigma(q_1(t_1), \dots, q_k(t_k))] \\
 &\Rightarrow_M u[q(r[t_1, \dots, t_k])] && \text{(rule } \mu) \\
 &\Rightarrow_M^* q_0(v[r[t_1, \dots, t_k]]) \\
 &= q_0(\bar{t}),
 \end{aligned}$$

where $\bar{s} \in T_\Sigma$, $u \in \widehat{T}_\Sigma(X_1)$, $v \in T_\Delta(X_1)$ and $\bar{t}, t_1, \dots, t_k \in T_\Delta$. By Corollary 3.13, $v \in \widehat{T}_\Delta(X_1)$. Moreover, since γ is a left matching path, it follows from Definition 3.16 and Lemma 3.20, that $\beta = \alpha\gamma$, where $\alpha = occ(u, x_1)$ and $\beta = occ(v, x_1)$. Since M is shape preserving, $\bar{s} \approx \bar{t}$, which implies that $\beta = \alpha\gamma \in occ(\bar{s})$. Then, clearly $\gamma \in occ(s)$, a contradiction.

Finally, that μ satisfies Condition (iv) can be seen easily with the help of Lemma 3.23. ■

Before we define formally how to construct the frame transducer of a transformable bottom-up tree transducer, we give some intuition for that construction.

Therefore let $M = (Q, \Sigma, \Delta, q_0, R)$ be a transformable bottom-up tree transducer. The frame transducer $fr(M)$ of M is a bottom-up tree transducer such that its rules scan and write out parts of those trees which are in the domain and range of τ_M , respectively. Of course these parts are considered by $fr(M)$ as input and output symbols from certain tree alphabets. This means that $\tau_{fr(M)} \subseteq T_{\Sigma_M} \times T_{\Delta_M}$, where Σ_M and Δ_M are tree alphabets, which contain certain parts of trees from $dom(\tau_M)$ and $ran(\tau_M)$, respectively, i.e. for every $\tilde{\sigma} \in \Sigma_M$ and $\tilde{\delta} \in \Delta_M$, $h_{\Sigma_M}(\tilde{\sigma}) \in T_\Sigma(X)$ and $h_{\Delta_M}(\tilde{\delta}) \in T_\Delta(X)$. During the construction of $fr(M)$, for every rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R

such that $\tau_{M,q}$ is infinite, we will define a set of rules of $fr(M)$. Let $inf(\mu) = \{i_1, \dots, i_l\}$, where $0 \leq l \leq k$ and $i_1 < i_2 < \dots < i_l$. Moreover let us consider a derivation

$$s = \sigma(s_1, \dots, s_k) \Rightarrow_M^* \sigma(q_1(t_1), \dots, q_k(t_k)) \Rightarrow_M q(r[t_1, \dots, t_k]) = q(t)$$

of M , where $s, s_1, \dots, s_k \in T_\Sigma$ and $t, t_1, \dots, t_k \in T_\Delta$ (note that at the last derivation step M applies the rule μ .) Instead of performing the derivations $s_i \Rightarrow_M^* q_i(t_i)$ of M , where $i \notin inf(\mu)$, and the application of μ (i.e. scanning σ and writing out r), $fr(M)$ will do the following. It will process a tree $\tilde{s} \in \widehat{T}_{\Sigma_M}(X_l)$ for which $h_{\Sigma_M}(\tilde{s}) = \sigma(\bar{s}_1, \dots, \bar{s}_k)$, where if $i \notin inf(\mu)$, then $\bar{s}_i = s_i$ else \bar{s}_i is an appropriate variable such that $\sigma(\bar{s}_1, \dots, \bar{s}_k) \in \widehat{T}_\Sigma(X_l)$. Moreover, $fr(M)$ will write out a tree $\tilde{t} \in \widehat{T}_{\Delta_M}(X_l)$ such that $h_{\Delta_M}(\tilde{t}) = r[\bar{t}_1, \dots, \bar{t}_k]$, where if $i \notin inf(\mu)$, then $\bar{t}_i = t_i$ else \bar{t}_i is a suitable variable such that $r[\bar{t}_1, \dots, \bar{t}_k] \in \widehat{T}_\Delta(X_l)$ (note that since M is transformable, by Condition (i) of Definition 3.35, for every $i \in inf(\mu)$, x_i occurs exactly once in the tree r). Clearly, $h_{\Sigma_M}(\tilde{s})$ and $h_{\Delta_M}(\tilde{t})$ are parts of the input and output trees s and t , respectively.

Now we give the intuition behind the definitions of $\sigma(\bar{s}_1, \dots, \bar{s}_k)$ and $r[\bar{t}_1, \dots, \bar{t}_k]$, i.e. the intuition of that the condition $\bar{s}_i = s_i$ and $\bar{t}_i = t_i$ holds, only if $i \notin inf(\mu)$. Therefore let S_μ be the set of all trees of the form $\sigma(u_1, \dots, u_k)$, where if $i \notin inf(\mu)$, then $u_i \in dom(\tau_{M,q_i})$ else u_i is an appropriate variable such that $S_\mu \subseteq \widehat{T}_\Sigma(X_l)$. Furthermore, for every $u = \sigma(u_1, \dots, u_k) \in S_\mu$, $T_{\mu,u}$ is defined to be the set of all trees of the form $r[v_1, \dots, v_k]$, where if $i \notin inf(\mu)$, then $v_i \in ran(\tau_{M,q_i}(u_i))$ else v_i is a suitable variable such that $T_{\mu,u} \subseteq \widehat{T}_\Delta(X_l)$. Assume that $||inf(\mu)|| \geq 1$. For every pair of trees (u, v) , where $u \in S_\mu$ and $v \in T_{\mu,u}$, $fr(M)$ will have a rule $\tilde{\mu} = \tilde{u}(q_{i_1}(x_1), \dots, q_{i_l}(x_l)) \rightarrow q(\tilde{v})$, where $\tilde{u} \in \Sigma_M^{(l)}$ and $\tilde{v} \in \widehat{T}_{\Delta_M}(X_l)$, such that $h_{\Sigma_M}(\tilde{u}) = u$ and $h_{\Delta_M}(\tilde{v}) = v$. Clearly the set of rules of $fr(M)$ must be finite, thus S_μ and $T_\mu = \sum_{u \in S_\mu} T_{\mu,u}$ must also be finite. On the other hand, if we defined in the definition of S_μ and $T_{\mu,u}$, for an index $i \in inf(\mu)$, that $u_i \in dom(\tau_{M,q_i})$ and $v_i \in ran(\tau_{M,q_i}(u_i))$, then S_μ and T_μ would be infinite (note that by Lemma 3.10, if for a state $q \in Q$, $\tau_{M,q}$ is infinite, then $ran(\tau_{M,q})$ is also infinite).

In this way, trees from $dom(\tau_{M,q_i})$ and $ran(\tau_{M,q_i})$, where $i \notin inf(\mu)$, are, so to speak, coded in the symbols of Σ_M and Δ_M , respectively. This is the reason of that when we will construct $fr(M)$, we will consider a rule μ in R only if $\tau_{M,q}$ is infinite, where q is the state occurring on the right-hand side of μ . (Note that since M is transformable, τ_{M,q_0} is infinite and since every state in M is useful, there is a rule μ in R such that q_0 occurs on the right-hand side of μ .)

In the following we give some examples to demonstrate the discussion above and to help the better understanding of the complex definition of the frame transducer appearing in Definition 3.38.

Example 3.37 Let us consider the shape preserving bottom-up tree transducer M_b of Example 2.4. By Lemma 3.36, M_b is transformable. Now we construct some rules of the frame transducer $fr(M_b)$ of M_b . According to the discussion given above, $fr(M_b)$ will scan and write out symbols from some tree alphabets Σ_{M_b} and Δ_{M_b} , respectively. This means that, for every $\tilde{\sigma} \in \Sigma_{M_b}$ and $\tilde{\delta} \in \Delta_{M_b}$, there are trees $s \in \text{dom}(\tau_{M_b})$ and $t \in \text{ran}(\tau_{M_b})$ such that $h_{\Sigma_{M_b}}(\tilde{\sigma})$ and $h_{\Delta_{M_b}}(\tilde{\delta})$ are parts of the trees s and t , respectively. The formal construction of $fr(M_b)$ will be given in Example 3.39. In the following we consider the rules μ_1, μ_3 and μ_6 of M_b (note that, for every state q from the right-hand sides of these rules, $\tau_{M_b,q}$ is infinite). Now we have the following three cases.

Case 1. Let us consider the derivation (\dagger) in Example 3.17 and the application of μ_3 in (\dagger) . While M_b performs the derivations $\alpha \Rightarrow_{M_b} q_\alpha(\beta_1)$ and $\alpha \Rightarrow_{M_b} q_\alpha(\beta_2)$ and scans the input symbol σ_1 , the frame transducer $fr(M_b)$ will do the following. It will process a tree $\tilde{\sigma}_{\mu_3,1}\tilde{\sigma}_{\mu_3,2}$, where $\tilde{\sigma}_{\mu_3,1} \in \Sigma_{M_b}^{(1)}$ and $\tilde{\sigma}_{\mu_3,2} \in \Sigma_{M_b}^{(0)}$ such that $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_3,1}) = \sigma_1(\alpha, x_1)$ and $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_3,2}) = \alpha$. Moreover it will write out a symbol $\tilde{\delta}_{\mu_3} \in \Delta_M^{(0)}$ for which $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_3}) = \beta_2$. According to this, $fr(M_b)$ will have the rules $\tilde{\sigma}_{\mu_3,1}(p_1(x_1)) \rightarrow q_1(x_1)$ and $\tilde{\sigma}_{\mu_3,2} \rightarrow p_1(\tilde{\delta}_{\mu_3})$, where p_1 is a new state not in Q . Clearly, with these rules $fr(M_b)$ can perform the derivation $\tilde{\sigma}_{\mu_3,1}\tilde{\sigma}_{\mu_3,2} \Rightarrow_{fr(M_b)}^* q_1(\tilde{\delta}_{\mu_3})$.

It can be seen easily that the trees $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_3,1}\tilde{\sigma}_{\mu_3,2}) = h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_3,1})[h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_3,2})] = \sigma_1(\alpha, x_1)[\alpha] = \sigma_1(\alpha, \alpha)$ and $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_3}) = \beta_2$ are parts of the input and output trees s and t , respectively. The trees $\sigma_1(\alpha, \alpha)$ and β_2 were resulted by the substitutions $\sigma_1(x_1, x_2)[\alpha, \alpha]$ and $x_1[\beta_2]$, respectively (note that $\text{inf}(\mu_3) = \emptyset$ and cf. the discussion before this example.)

Now we use the matching path of μ_3 to explain how the trees $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_3,1}) = \sigma_1(\alpha, x_1)$ and $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_3,2}) = \alpha$ can be determined from the tree $\sigma_1(\alpha, \alpha)$. By Example 3.17 and Corollary 3.21, $mp(\mu_3)=2$ and 2 is a left matching path of μ_3 . Therefore we split the tree $\sigma_1(\alpha, \alpha)$ into trees $\sigma_1(\alpha, x_1)$ and α as follows. $\alpha = \text{stree}(\sigma_1(\alpha, \alpha), 2)$ and $\sigma_1(\alpha, x_1)$ is the tree which we get by replacing in $\sigma_1(\alpha, \alpha)$ the subtree at the occurrence 2 with the variable x_1 . This splitting of the tree $\sigma_1(\alpha, \alpha)$ is necessary because we want $fr(M_b)$ to be a shape preserving bottom-up tree transducer.

Case 2. Let us consider now the derivation (\ddagger) in Example 2.4 and the application of μ_6 in (\ddagger) . Instead of the subderivation $\alpha \Rightarrow_{M_b} q_\alpha(\beta_1)$ and the application of μ_6 , which scans the input symbol γ_1 and writes out the tree $\omega\omega(x_1)$, the frame transducer $fr(M_b)$ will do the following. It will scan a symbol $\tilde{\sigma}_{\mu_6} \in \Sigma_{M_b}^{(0)}$ for which $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_6}) = \gamma_1\alpha$ and write out a tree $\tilde{\delta}_{\mu_6,1}\tilde{\delta}_{\mu_6,2}$, where $\tilde{\delta}_{\mu_6,1} \in \Delta_{M_b}^{(1)}$ and $\tilde{\delta}_{\mu_6,2} \in \Delta_{M_b}^{(0)}$ such that $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_6,1}) = \omega(x_1)$ and $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_6,2}) = \omega\beta_1$. According to this, $fr(M_b)$ will have the rule $\tilde{\sigma}_{\mu_6} \rightarrow q_3(\tilde{\delta}_{\mu_6,1}\tilde{\delta}_{\mu_6,2})$.

Clearly the trees $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_6}) = \gamma_1\alpha$ and $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_6,1}\tilde{\delta}_{\mu_6,2}) = \omega(x_1)[\omega\beta_1] = \omega\omega\beta_1$ are parts of the input and output trees s and t respectively. The trees $\gamma_1\alpha$ and $\omega\omega\beta_1$

were resulted by the substitutions $\gamma_1(x_1)[\alpha]$ and $\omega\omega(x_1)[\beta_1]$, respectively (note that $\inf(\mu_6) = \emptyset$).

Now we use the matching path of μ_6 to explain how the trees $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_6,1}) = \omega(x_1)$ and $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_6,2}) = \omega\beta_1$ can be determined from the tree $\omega\omega\beta_1$. By Example 3.17 and Corollary 3.21, $mp(\mu_6) = 1$, moreover 1 is a right matching path of μ_6 . Therefore we split the tree $\omega\omega\beta_1$ into trees $\omega(x_1)$ and $\omega\beta_1$ as follows. $\omega\beta_1 = stree(\omega\omega\beta_1, 1)$ and $\omega(x_1)$ is the tree which we get by replacing in $\omega\omega\beta_1$ the subtree at the occurrence 1 with the variable x_1 . Again, this splitting of the tree $\omega\omega\beta_1$ is necessary in order to ensure $fr(M_b)$ to be a shape preserving bottom-up tree transducer.

Case 3. Finally, let us consider the derivation (\dagger) in Example 2.4 and the application of μ_1 in (\dagger) . Instead of the subderivation $\alpha \Rightarrow_{M_b} q_\alpha(\beta_2)$ of (\dagger) and the application of the rule μ_1 , which scans the symbol σ and writes out the tree $\delta(\delta_1(\beta, x_1), x_2, x_3)$, $fr(M_b)$ will do the following. It will scan a symbol $\tilde{\sigma}_{\mu_1} \in \Sigma_{M_b}^{(2)}$ for which $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_1}) = \sigma(x_1, \alpha, x_2)$, and write out a tree $\tilde{\delta}_{\mu_1,1}(\tilde{\delta}_{\mu_1,2}(x_1), x_2)$, where $\tilde{\delta}_{\mu_1,1} \in \Delta_{M_b}^{(2)}$ and $\tilde{\delta}_{\mu_1,2} \in \Delta_{M_b}^{(1)}$ such that $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_1,1}) = \delta(x_1, \beta_2, x_2)$ and $h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_1,2}) = \delta_1(\beta, x_1)$. Consequently $fr(M_b)$ will have a rule $\tilde{\sigma}_{\mu_1}(q_1(x_1), q_2(x_2)) \rightarrow q_0(\tilde{\delta}_{\mu_1,1}(\tilde{\delta}_{\mu_1,2}(x_1), x_2))$.

One can easily see that the trees $h_{\Sigma_{M_b}}(\tilde{\sigma}_{\mu_1}) = \sigma(x_1, \alpha, x_2)$ and

$$\begin{aligned} h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_1,1}(\tilde{\delta}_{\mu_1,2}(x_1), x_2)) &= h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_1,1})[h_{\Delta_{M_b}}(\tilde{\delta}_{\mu_1,2}), x_2] = \\ &\delta(x_1, \beta_2, x_2)[\delta_1(\beta, x_1), x_2] = \delta(\delta_1(\beta, x_1), \beta_2, x_2) \end{aligned}$$

are parts of the input and output trees s and t , respectively. Let us denote the tree $\delta(\delta_1(\beta, x_1), \beta_2, x_2)$ by t_{μ_1} . The trees $\sigma(x_1, \alpha, x_2)$ and t_{μ_1} were resulted by the substitutions $\sigma(x_1, x_2, x_3)[x_1, \alpha, x_2] = \sigma(x_1, \alpha, x_2)$ and $\delta(\delta_1(\beta, x_1), x_2, x_3)[x_1, \beta_2, x_2] = t_{\mu_1}$ (note that $\inf(\mu_1) = \{1, 3\}$).

The trees $\delta(x_1, \beta_2, x_2)$ and $\delta_1(\beta, x_1)$ were resulted by the decomposition of t_{μ_1} . This decomposition is necessary in order to ensure $fr(M_b)$ to be a shape preserving bottom-up tree transducer. \square

Next we define formally the frame transducer of a transformable bottom-up tree transducer.

Definition 3.38 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a transformable bottom-up tree transducer. We define the bottom-up tree transducer $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$, called the *frame transducer* of M in the following way.

First, for every rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R such that $\tau_{M,q}$ is infinite, we define the tree alphabets Σ_μ , Δ_μ , the set of rules R_μ and the set of states Q_μ as follows. Let $\inf(\mu) = \{i_1, \dots, i_l\}$, where $0 \leq l \leq k$ and $i_1 < i_2 < \dots < i_l$. Let

$$S_\mu = \{\sigma(s_1, \dots, s_k) \mid \forall j \in [l] : s_{i_j} = x_j \text{ and } \forall i \in [k], i \notin \inf(\mu) : s_i \in \text{dom}(\tau_{M,q_i})\}$$

and, for every $s = \sigma(s_1, \dots, s_k) \in S_\mu$, let

$$T_{\mu,s} = \{r[t_1, \dots, t_k] \mid \forall j \in [l] : t_{i_j} = x_j \text{ and } \forall i \in [k], i \notin \text{inf}(\mu) : t_i \in \tau_{M,q_i}(s_i)\}.$$

Moreover let $T_\mu = \bigcup_{s \in S_\mu} T_{\mu,s}$. Clearly S_μ and T_μ are finite. According to the cardinality of the set $\text{inf}(\mu)$, we distinguish the following two cases.

Case 1: $\|\text{inf}(\mu)\| = 0$. Let $\gamma = mp(\mu)$. Then we distinguish the following two sub cases.

Case 1.a: γ is a left matching path of μ . Let $n = \text{length}(\gamma)$. Since M is transformable, for every $s \in S_\mu$, we have $\gamma \in \text{occ}(s)$. Let $\Delta_\mu = T_\mu$ and

$$\Sigma_\mu = \bigcup_{s \in S_\mu} (\{\text{stree}(s, \gamma)\} \cup \text{dec}(\bar{s})),$$

where $\bar{s} \in \widehat{T}_\Sigma(X_1)$ such that $s = \bar{s}[\text{stree}(s, \gamma)]$. Moreover, let

$$R_{\mu,s} = \{\tilde{\sigma}_1(p_1(x_1)) \rightarrow p_0(x_1), \dots, \tilde{\sigma}_n(p_n(x_1)) \rightarrow p_{n-1}(x_1)\} \cup \{\tilde{\sigma} \rightarrow p_n(\tilde{\delta}) \mid \tilde{\delta} \in \Delta_\mu, h_{\Delta_\mu}(\tilde{\delta}) \in T_{\mu,s}\},$$

where $p_0 = q$, p_1, \dots, p_n are new states and $\tilde{\sigma}_1, \dots, \tilde{\sigma}_n \in \Sigma_\mu^{(1)}$, $\tilde{\sigma} \in \Sigma_\mu^{(0)}$ such that $h_{\Sigma_\mu}(\tilde{\sigma}_1 \dots \tilde{\sigma}_n \tilde{\sigma}) = s$. Furthermore, let $Q_{\mu,s} = \{p_0, \dots, p_n\}$, $R_\mu = \bigcup_{s \in S_\mu} R_{\mu,s}$ and $Q_\mu = \bigcup_{s \in S_\mu} Q_{\mu,s}$.

For example, if we consider Case 1 in Example 3.37, then $\tilde{\sigma}_{\mu_3,1}, \tilde{\sigma}_{\mu_3,2} \in \Sigma_{\mu_3}$, $\tilde{\delta}_{\mu_3} \in \Delta_{\mu_3}$ and $\{\tilde{\sigma}_{\mu_3,1}(p_1(x_1)) \rightarrow q_1(x_1), \tilde{\sigma}_{\mu_3,2} \rightarrow p_1(\tilde{\delta}_{\mu_3,1})\} \subseteq R_{\mu_3, \sigma_1(\alpha, \alpha)}$.

Case 1.b: γ is a right matching path of μ . Since M is transformable, for every $t \in T_\mu$, we have $\gamma \in \text{occ}(t)$. Let $\Sigma_\mu = S_\mu$ and $\Delta_\mu = \bigcup_{t \in T_\mu} (\{\text{stree}(t, \gamma)\} \cup \text{dec}(\bar{t}))$, where $\bar{t} \in T_\Delta(X_1)$ such that $t = \bar{t}[\text{stree}(t, \gamma)]$.

Let, for every $s \in S_\mu$, $\tilde{s} \in \Sigma_\mu^{(0)}$ such that $h_{\Sigma_\mu}(\tilde{s}) = s$ and let, for every $t \in T_\mu$, $\tilde{t} = \tilde{\gamma}\tilde{\delta}$, where $\tilde{\gamma} \in (\Delta_\mu^{(1)})^*$ and $\tilde{\delta} \in \Delta_\mu^{(0)}$, such that $h_{\Delta_\mu}(\tilde{t}) = t$ and $h_{\Delta_\mu}(\tilde{\delta}) = \text{stree}(t, \gamma)$. Let $R_\mu = \bigcup_{s \in S_\mu, t \in T_\mu, s} \{\tilde{s} \rightarrow q(\tilde{t})\}$ and let $Q_\mu = \{q\}$.

For instance, if we consider Case 2 in Example 3.37, then $\tilde{\sigma}_{\mu_6} \in \Sigma_{\mu_6}$, $\tilde{\delta}_{\mu_6,1}, \tilde{\delta}_{\mu_6,2} \in \Delta_{\mu_6}$ and $\tilde{\sigma}_{\mu_6} \rightarrow q_3(\tilde{\delta}_{\mu_6,1}\tilde{\delta}_{\mu_6,2}) \in R_{\mu_6}$.

Case 2: $\|\text{inf}(\mu)\| \geq 1$. Let Σ_μ be the tree alphabet S_μ . Since M is transformable, $T_\mu \subseteq \widehat{T}_\Delta(X_1)$ and, if $l > 1$, then, for every $t \in T_\mu$, the conditions required by Definition 3.33 hold. If $T_\mu \neq \{x_1\}$, then let $\Delta_\mu = \bigcup_{t \in T_\mu} \text{dec}(t)$. Moreover, let, for every $s \in S_\mu$, $\tilde{s} \in \Sigma_\mu^{(l)}$ such that $h_{\Sigma_\mu}(\tilde{s}) = s$. Furthermore, if $\|\text{inf}(\mu)\| = 1$, then let, for every $t \in T_\mu$, $\tilde{t} \in (\Delta_\mu^{(1)})^*$, such that $h_{\Delta_\mu}(\tilde{t}) = t$. If $\|\text{inf}(\mu)\| > 1$, then let $\gamma = mp(\mu)$ and let, for every $t \in T_\mu$, $\bar{t} \in \widehat{T}_\Delta(X_1)$ such that $t = \bar{t}[\text{stree}(t, \gamma)]$ and $\tilde{t} = \tilde{\gamma}\tilde{\delta}(\tilde{\gamma}_1 x_1, \dots, \tilde{\gamma}_l x_l)$, where $\tilde{\gamma}, \tilde{\gamma}_1, \dots, \tilde{\gamma}_l \in (\Delta_\mu^{(1)})^*$ and $\tilde{\delta} \in \Delta_\mu^{(l)}$, such that $h_{\Delta_\mu}(\tilde{t}) = t$ and $h_{\Delta_\mu}(\tilde{\gamma}) = \bar{t}$.

Now, let $R_\mu = \bigcup_{s \in S_\mu, t \in T_\mu, s} \{\tilde{s}(q_{i_1}(x_1), \dots, q_{i_l}(x_l)) \rightarrow q(\tilde{t})\}$ and $Q_\mu = \{q, q_{i_1}, \dots, q_{i_l}\}$.

As in the previous cases, we give an example. If we consider Case 3 in Example 3.37, then $\tilde{\sigma}_{\mu_1} \in \Sigma_{\mu_1}$, $\tilde{\delta}_{\mu_1,1}, \tilde{\delta}_{\mu_1,2} \in \Delta_{\mu_1}$ and $\tilde{\sigma}_{\mu_1}(q_1(x_1), q_2(x_2)) \rightarrow q_0(\tilde{\delta}_{\mu_1,1}(\tilde{\delta}_{\mu_1,2}(x_1), x_2)) \in R_{\mu_6}$.

Finally, we define the sets Q_M, Σ_M, Δ_M and R_M as follows. Let R' be the smallest subset of R such that, for every rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R , if $\tau_{M,q}$ is infinite, then $\mu \in R'$. Moreover, let $Q_M = \bigcup_{\mu \in R'} Q_\mu$, $\Sigma_M = \bigcup_{\mu \in R'} \Sigma_\mu$, $\Delta_M = \bigcup_{\mu \in R'} \Delta_\mu$ and $R_M = \bigcup_{\mu \in R'} R_\mu$. Clearly these sets are finite, $q_0 \in Q_M$ and the sets Σ_M and Δ_M are tree alphabets. \square

Using Definition 3.38, we can construct the frame transducer of the bottom-up tree transducer M_b appearing in Example 2.4.

Example 3.39 Let M_b be the bottom-up tree transducer appearing in Example 2.4. Since M_b is shape preserving and τ_{M_b} is infinite, it follows from Lemma 3.36 that M_b is transformable. Now we give the frame transducer $fr(M_b)$ of M_b .

Let μ be a rule in R . We gave the set $inf(\mu)$ in Example 3.17. Moreover, if $||inf(\mu)|| \neq 1$ and $\tau_{M_b,q}$ is infinite, where q is the state occurring on the right-hand side of μ , then a matching path $\gamma \in mp(\mu)$ was also given, which by Corollary 3.21 implies that $mp(\mu) = \gamma$.

In Table 3.1 we show the sets which are necessary to define $fr(M_b)$ (cf. also Example 3.37). Let $fr(M_b) = (Q_{M_b}, \Sigma_{M_b}, \Delta_{M_b}, q_0, R_{M_b})$, where

- $Q_{M_b} = \{q_0, q_1, q_2, q_3, p_1\}$

- $\Sigma_{M_b} = \{\tilde{\sigma}_1, \dots, \tilde{\sigma}_5\}$, where

$$\begin{aligned} \tilde{\sigma}_1 &= \sigma(x_1, \alpha, x_2), & \tilde{\sigma}_3 &= \alpha, & \tilde{\sigma}_5 &= \gamma_1(\alpha). \\ \tilde{\sigma}_2 &= \sigma_1(\alpha, x_1), & \tilde{\sigma}_4 &= \gamma(x_1), \end{aligned}$$

- $\Delta_{M_b} = \{\tilde{\delta}_1, \dots, \tilde{\delta}_8\}$, where

$$\begin{aligned} \tilde{\delta}_1 &= \delta(x_1, \beta_1, x_2), & \tilde{\delta}_4 &= \beta_1, & \tilde{\delta}_7 &= \omega(\beta_1), \\ \tilde{\delta}_2 &= \delta(x_1, \beta_2, x_2), & \tilde{\delta}_5 &= \beta_2, & \tilde{\delta}_8 &= \omega(\beta_2). \\ \tilde{\delta}_3 &= \delta_1(\beta, x_1), & \tilde{\delta}_6 &= \omega(x_1), \end{aligned}$$

- $R_{M_b} = \{\tilde{\mu}_1, \dots, \tilde{\mu}_{10}\}$, where

$$\begin{aligned} \tilde{\mu}_1 &= \tilde{\sigma}_1(q_1(x_1), q_2(x_2)) \rightarrow q_0(\tilde{\delta}_1(\tilde{\delta}_3(x_1), x_2)), \\ \tilde{\mu}_2 &= \tilde{\sigma}_1(q_1(x_1), q_2(x_2)) \rightarrow q_0(\tilde{\delta}_2(\tilde{\delta}_3(x_1), x_2)), \\ \tilde{\mu}_3 &= \tilde{\sigma}_2(q_0(x_1)) \rightarrow q_1(x_1), \\ \tilde{\mu}_4 &= \tilde{\sigma}_2(p_1(x_1)) \rightarrow q_1(x_1), \\ \tilde{\mu}_5 &= \tilde{\sigma}_3 \rightarrow p_1(\tilde{\delta}_4), \end{aligned}$$

μ	$\inf(\mu)$	$mp(\mu)$	S_μ	T_μ	Σ_μ	Δ_μ
μ_1	1, 3	ε	$\sigma(x_1, \alpha, x_2)$	$\delta(\delta_1(\beta, x_1), \beta_1, x_2)$ $\delta(\delta_1(\beta, x_1), \beta_2, x_2)$	$\sigma(x_1, \alpha, x_2)$	$\delta(x_1, \beta_1, x_2)$ $\delta(x_1, \beta_2, x_2)$ $\delta_1(\beta, x_1)$
μ_2	2	not defined	$\sigma_1(\alpha, x_1)$	x_1	$\sigma_1(\alpha, x_1)$	not defined
μ_3	\emptyset	2 (left m.p.)	$\sigma_1(\alpha, \alpha)$	β_1 β_2	$\sigma_1(\alpha, x_1)$ α	β_1 β_2
μ_4	1	not defined	$\gamma(x_1)$	x_1	$\gamma(x_1)$	not defined
μ_5	1	not defined	$\gamma(x_1)$	$\omega(x_1)$	$\gamma(x_1)$	$\omega(x_1)$
μ_6	\emptyset	1 (right m.p.)	$\gamma_1(\alpha)$	$\omega(\omega(\beta_1))$ $\omega(\omega(\beta_2))$	$\gamma_1(\alpha)$	$\omega(x_1)$ $\omega(\beta_1)$ $\omega(\beta_2)$

Table 3.1: The sets which are necessary to define the frame transducer in Example 3.39.

$$\begin{aligned}
\tilde{\mu}_6 &= \tilde{\sigma}_3 \rightarrow p_1(\tilde{\delta}_5), \\
\tilde{\mu}_7 &= \tilde{\sigma}_4(q_3(x_1)) \rightarrow q_2(x_1), \\
\tilde{\mu}_8 &= \tilde{\sigma}_4(q_3(x_1)) \rightarrow q_3(\tilde{\delta}_6(x_1)), \\
\tilde{\mu}_9 &= \tilde{\sigma}_5 \rightarrow q_3(\tilde{\delta}_6(\tilde{\delta}_7)), \\
\tilde{\mu}_{10} &= \tilde{\sigma}_5 \rightarrow q_3(\tilde{\delta}_6(\tilde{\delta}_8)).
\end{aligned}$$

Note that in Example 3.37, the symbols $\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3$ and $\tilde{\sigma}_5$ were denoted by $\tilde{\sigma}_{\mu_1}, \tilde{\sigma}_{\mu_3,1}, \tilde{\sigma}_{\mu_3,2}$ and $\tilde{\sigma}_{\mu_6}$, respectively. Moreover, $\tilde{\delta}_2, \tilde{\delta}_3, \tilde{\delta}_5, \tilde{\delta}_6$ and $\tilde{\delta}_7$ were denoted by $\tilde{\delta}_{\mu_1,1}, \tilde{\delta}_{\mu_1,2}, \tilde{\delta}_{\mu_3}, \tilde{\delta}_{\mu_6,1}$ and $\tilde{\delta}_{\mu_6,2}$, respectively. \square

It turns out easily from Definition 3.38 that the frame transducer $fr(M)$ of a transformable bottom-up tree transducer M is a bottom-up quasirelabeling tree transducer. We will show in Subsection 3.2.3 that if M is shape preserving, then $fr(M)$ is also shape preserving.

We will need the following observation in what follows.

Observation 3.40 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a transformable bottom-up tree transducer, and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Then the following statements hold.

- (i) $fr(M)$ is a bottom-up quasirelabeling tree transducer.
- (ii) Let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R_M such that $k > 1$. Then $q, q_1, \dots, q_k \in Q$.
- (iii) Let $\mu = \sigma(p(x_1)) \rightarrow q(\gamma(x_1))$ be a rule in R_M , where $\gamma \in (\Delta_M^{(1)})^*$. If $p \in Q$, then also $q \in Q$.
- (iv) For every state $q \in Q$ the condition $q \in Q_M$ holds if and only if the set $\tau_{M,q}$ is infinite.
- (v) Let $u \in \hat{T}_{\Sigma_M}(X_1)$, $v \in T_{\Delta_M}(X_1)$ and $p, q \in Q_M$ such that there is a derivation $u[p(x_1)] \Rightarrow_{fr(M)}^* q(v)$. Then $v \in \hat{T}_{\Delta_M}(X_1)$.

Proof. The statements (i)-(iv) easily follow from Definition 3.38, while Statement (v) follows from the fact that, also by Definition 3.38, $fr(M)$ is linear and nondeleting. ■

We will need also the following observation.

Observation 3.41 Let Σ and Δ be ranked alphabets, let h be a linear and nondeleting tree homomorphism from $T_\Sigma(X)$ to $T_\Delta(X)$ and let $u \in \hat{T}_\Sigma(X_1)$. Then $h(u) \in \hat{T}_\Delta(X_1)$.

Proof. The proof is straightforward and thus it is left to the reader. ■

In the rest of the Thesis, if we consider a tree u in $\hat{T}_\Sigma(X_1)$ and a linear and nondeleting tree homomorphism $h : T_\Sigma(X) \rightarrow T_\Delta(X)$, where Σ and Δ are ranked alphabets, then we will assume that $h(u) \in \hat{T}_\Delta(X_1)$ without referring to Observation 3.41. Note that the tree homomorphisms h_{Σ_M} and h_{Δ_M} , appearing in Definition 3.38, are linear and nondeleting.

In the following we show that the frame transducer of a transformable bottom-up tree transducer M , broadly speaking, preserves the tree transformation computed by M . The reason of this, as we have already seen in Example 3.37 and in the discussion before it, is that the frame transducer does nothing else but performs more derivation steps of M in one derivation step. To demonstrate this we recall the derivations of the transducer M_b appearing in Examples 2.4 and 3.17, and give the corresponding derivations of $fr(M_b)$.

Example 3.42 Let us consider again the shape preserving bottom-up transducer M_b of Example 2.4, the derivations (\dagger) and (\ddagger) appearing in the same example, moreover

the derivation (\dagger) in Example 3.17. Let $fr(M_b)$ be the frame transducer of M_b defined in Example 3.39. In the present example, for every of these derivations of M_b , we associate a derivation of $fr(M_b)$ which, roughly speaking, simulates the corresponding one of M_b . According to the derivations of M_b we consider the following three cases.

Case 1: The derivation (\dagger) of M_b in Example 3.17. We can associate to this derivation the following one of $fr(M_b)$.

$$\begin{aligned}
 \tilde{s} &= \tilde{\sigma}_1(\tilde{\sigma}_2\tilde{\sigma}_3, \tilde{\sigma}_4\tilde{\sigma}_5) \\
 &\Rightarrow_{fr(M_b)} \tilde{\sigma}_1(\tilde{\sigma}_2p_1(\tilde{\delta}_5), \tilde{\sigma}_4\tilde{\sigma}_5) && (\text{rule } \tilde{\mu}_6) \\
 &\Rightarrow_{fr(M_b)} \tilde{\sigma}_1(q_1(\tilde{\delta}_5), \tilde{\sigma}_4\tilde{\sigma}_5) && (\text{rule } \tilde{\mu}_4) \\
 &\Rightarrow_{fr(M_b)}^* q_0(\tilde{\delta}_1(\tilde{\delta}_3\tilde{\delta}_5, \tilde{\delta}_6\tilde{\delta}_8)) \\
 &= q_0(\tilde{t}).
 \end{aligned}$$

It is easy to see that if s and t are the trees defined in Example 3.17, then $h_{\Sigma_{M_b}}(\tilde{s}) = s$ and $h_{\Delta_{M_b}}(\tilde{t}) = t$. We note that $\tilde{s} \approx \tilde{t}$.

Case 2: The derivation (\ddagger) of M_b in Example 2.4. Now, we give the corresponding derivation of $fr(M_b)$.

$$\begin{aligned}
 \tilde{s} &= \tilde{u}[\tilde{\sigma}_4\tilde{\sigma}_5] \\
 &\Rightarrow_{fr(M_b)} \tilde{u}[\tilde{\sigma}_4q_3(\tilde{\delta}_6\tilde{\delta}_7)] && (\text{rule } \tilde{\mu}_9) \\
 &\Rightarrow_{fr(M_b)} \tilde{u}[q_2(\tilde{\delta}_6\tilde{\delta}_7)] && (\text{rule } \tilde{\mu}_7) \\
 &\Rightarrow_{fr(M_b)}^* q_0(\tilde{v}[\tilde{\delta}_6\tilde{\delta}_7]) \\
 &= q_0(\tilde{t}),
 \end{aligned}$$

where $\tilde{u} = \tilde{\sigma}_1(\tilde{\sigma}_2(\tilde{s}_1), x_1)$, $\tilde{v} = \tilde{\delta}_2(\tilde{\delta}_3(\tilde{t}_1), x_1)$, $\tilde{s}_1 = \tilde{\sigma}_1(\tilde{\sigma}_2\tilde{\sigma}_3, \tilde{s}_2)$, $\tilde{t}_1 = \tilde{\delta}_2(\tilde{\delta}_3\tilde{\delta}_4, \tilde{t}_2)$, $\tilde{s}_2 = \tilde{\sigma}_4\tilde{\sigma}_5$ and $\tilde{t}_2 = \tilde{\delta}_6\tilde{\delta}_7$. It can be seen that $\tilde{u}[q_2(x_1)] \Rightarrow_{fr(M_b)}^* q_0(\tilde{v})$ and thus the above derivation is a valid derivation of $fr(M_b)$. It is also easy to see that if s and t are the trees defined in Example 2.4, then $h_{\Sigma_{M_b}}(\tilde{s}) = s$ and $h_{\Delta_{M_b}}(\tilde{t}) = t$. We note that $\tilde{s}_2 \approx \tilde{t}_2$, which implies that $\tilde{s}_1 \approx \tilde{t}_1$, and thus $\tilde{s} \approx \tilde{t}$.

Case 3: The derivation (\dagger) of M_b in Example 2.4. The corresponding derivation of $fr(M_b)$ is the following one.

$$\begin{aligned}
 \tilde{s} &= \tilde{\sigma}_1(\tilde{\sigma}_2(\tilde{s}_1), \tilde{s}_2) \\
 &\Rightarrow_{fr(M_b)}^* \tilde{\sigma}_1(\tilde{\sigma}_2(q_0(\tilde{t}_1)), \tilde{s}_2) \\
 &\Rightarrow_{fr(M_b)} \tilde{\sigma}_1(q_1(\tilde{t}_1), \tilde{s}_2) && (\text{rule } \tilde{\mu}_3) \\
 &\Rightarrow_{fr(M_b)}^* \tilde{\sigma}_1(q_1(\tilde{t}_1), q_2(\tilde{t}_2)) \\
 &\Rightarrow_{fr(M_b)} q_0(\tilde{\delta}_2(\tilde{\delta}_3(\tilde{t}_1), \tilde{t}_2)) && (\text{rule } \tilde{\mu}_2) \\
 &= q_0(\tilde{t}),
 \end{aligned}$$

where the trees \tilde{s} , \tilde{t} , \tilde{s}_1 , \tilde{t}_1 , \tilde{s}_2 and \tilde{t}_2 are the same as in the previous case. \square

Lemma 3.43 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a transformable bottom-up tree transducer, and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Let $q \in Q \cap Q_M$, then $\tau_{M,q} = h_{\Sigma_M}^{-1} \circ \tau_{fr(M),q} \circ h_{\Delta_M}$.

Proof. First we show that $\tau_{M,q} \subseteq h_{\Sigma_M}^{-1} \circ \tau_{fr(M),q} \circ h_{\Delta_M}$. To see this, it is enough to show the following. Let $s \in T_\Sigma$ and $t \in T_\Delta$ such that $s \Rightarrow_M^* q(t)$. Then there are trees $\tilde{s} \in T_{\Sigma_M}$ and $\tilde{t} \in T_{\Delta_M}$, such that $s = h_{\Sigma_M}(\tilde{s})$, $t = h_{\Delta_M}(\tilde{t})$ and $\tilde{s} \Rightarrow_{fr(M)}^* q(\tilde{t})$. We prove this by induction on the length n of the derivation $s \Rightarrow_M^* q(t)$. If $n = 1$, then $s \rightarrow q(t) \in R$. Then, clearly, $s = \sigma$, for some $\sigma \in \Sigma^{(0)}$ and $||inf(s \rightarrow q(t))|| = 0$. Let $\gamma = mp(\mu)$. By Definition 3.16 and Item (iii) of Definition 3.35, it follows that γ can be only a right matching path of μ . Then, by Case 1.b of Definition 3.38, there are trees $\tilde{s} \in T_{\Sigma_M}$ and $\tilde{t} \in T_{\Delta_M}$ such that $h_{\Sigma_M}(\tilde{s}) = s$ and $h_{\Delta_M}(\tilde{t}) = t$, moreover $\tilde{s} \rightarrow q(\tilde{t}) \in R_M$. This implies that $\tilde{s} \Rightarrow_{fr(M)}^* q(\tilde{t})$.

Now let us suppose that the statement holds for $n > 1$. For the induction step from n to $n + 1$, let us assume that there is a derivation $s \Rightarrow_M^{n+1} q(t)$. Let $s = \sigma(s_1, \dots, s_k)$, where $k \geq 1$, $\sigma \in \Sigma^{(k)}$ and $s_1, \dots, s_k \in T_\Sigma$. Then there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R such that the derivation $s \Rightarrow_M^{n+1} q(t)$ can be written in the following form.

$$\begin{aligned}
 s &= \sigma(s_1, \dots, s_k) \\
 &\Rightarrow_M^n \sigma(q_1(t_1), \dots, q_k(t_k)) \\
 &\Rightarrow_M q(r[t_1, \dots, t_k]) && \text{(rule } \mu) \\
 &= q(t),
 \end{aligned}$$

where $t_1, \dots, t_k \in T_\Delta$. According to the cardinality of $inf(\mu)$, we distinguish the following two cases.

Case 1: $||inf(\mu)|| = 0$. Let $\gamma = mp(\mu)$. Now, we have the following two sub cases.

Case 1.a: γ is a left matching path of μ (for example as in the case of μ_3 applied in the derivation (\dagger) of Example 3.17). Let $n = length(\gamma)$. Then, by Case 1.a of Definition 3.38, there are rules $\tilde{\sigma}_1(p_1(x_1)) \rightarrow p_0(x_1), \dots, \tilde{\sigma}_n(p_n(x_1)) \rightarrow p_{n-1}(x_1), \tilde{\sigma} \rightarrow p_n(\tilde{\delta})$ in R_M , where $p_1, \dots, p_n \in Q_M$, $p_0 = q$, $\tilde{\sigma}_1, \dots, \tilde{\sigma}_n \in \Sigma_M^{(1)}$, $\tilde{\sigma} \in \Sigma_M^{(0)}$ and $\tilde{\delta} \in \Delta_M^{(0)}$, such that $h_{\Sigma_M}(\tilde{\sigma}_1 \dots \tilde{\sigma}_n \tilde{\sigma}) = s$ and $h_{\Delta_M}(\tilde{\delta}) = t$. Now let $\tilde{s} = \tilde{\sigma}_1 \dots \tilde{\sigma}_n \tilde{\sigma}$ and $\tilde{t} = \tilde{\delta}$. Clearly $\tilde{s} \Rightarrow_{fr(M)}^* q(\tilde{t})$ (cf. the derivation appearing in Case 1 of Example 3.42, and the rules $\tilde{\mu}_4, \tilde{\mu}_6$ applied in this derivation).

Case 1.b: γ is a right matching path of μ (as in the case of μ_6 applied in the derivation (\ddagger) in Example 2.4). Then, by Case 1.b of Definition 3.38, there is a rule $\tilde{\sigma} \rightarrow q(\tilde{\gamma}\tilde{\delta})$ in R_M , where $\tilde{\sigma} \in \Sigma_M^{(0)}$, $\tilde{\gamma} \in (\Delta_M^{(1)})^*$ and $\tilde{\delta} \in \Delta_M^{(0)}$ such that $h_{\Sigma_M}(\tilde{\sigma}) = s$ and $h_{\Delta_M}(\tilde{\gamma}\tilde{\delta}) = t$. Let $\tilde{s} = \tilde{\sigma}$ and $\tilde{t} = \tilde{\gamma}\tilde{\delta}$, clearly $\tilde{s} \Rightarrow_{fr(M)}^* q(\tilde{t})$ (cf. the derivation appearing in Case 2 of Example 3.42, and the rule $\tilde{\mu}_9$ applied in it).

Case 2: $||inf(\mu)|| \geq 1$ (for example as in the case of μ_1 applied in the derivation (\dagger) in Example 2.4). Let $inf(\mu) = \{i_1, \dots, i_l\}$, where $l \geq 1$ and $i_1 < i_2 < \dots < i_l$.

For every $i \in \text{inf}(\mu)$, the set τ_{M,q_i} is infinite, which, by Statement (iv) of Observation 3.40, implies that $q_i \in Q \cap Q_M$. Then, by induction hypothesis, for every $i \in \text{inf}(\mu)$, there are trees $\tilde{s}_i \in \text{dom}(\tau_{fr(M),q_i})$ and $\tilde{t}_i \in \text{ran}(\tau_{fr(M),q_i})$, such that $s_i = h_{\Sigma_M}(\tilde{s}_i)$ and $t_i = h_{\Delta_M}(\tilde{t}_i)$.

Moreover, by Case 2 of Definition 3.38, there is a rule $\tilde{\mu} = \tilde{\sigma}(q_{i_1}(x_1), \dots, q_{i_l}(x_l)) \rightarrow q(\tilde{r})$ in R_M , such that $h_{\Sigma_M}(\tilde{\sigma})[s_{i_1}, \dots, s_{i_l}] = s$ and $h_{\Delta_M}(\tilde{r})[t_{i_1}, \dots, t_{i_l}] = t$. Let $\tilde{s} = \tilde{\sigma}(\tilde{s}_{i_1}, \dots, \tilde{s}_{i_l})$ and $\tilde{t} = \tilde{r}[\tilde{t}_{i_1}, \dots, \tilde{t}_{i_l}]$. Using that h_{Σ_M} and h_{Δ_M} are tree homomorphisms, it can be seen that $s = h_{\Sigma_M}(\tilde{s})$, $t = h_{\Delta_M}(\tilde{t})$. Furthermore there is a derivation of $fr(M)$ of the following form (cf. the derivation in Case 3 of Example 3.42, and the rule $\tilde{\mu}_2$ applied in it).

$$\begin{aligned}
 \tilde{s} &= \tilde{\sigma}(\tilde{s}_{i_1}, \dots, \tilde{s}_{i_l}) \\
 &\Rightarrow_{fr(M)}^* \tilde{\sigma}(q_{i_1}(\tilde{t}_{i_1}), \dots, q_{i_l}(\tilde{t}_{i_l})) \\
 &\Rightarrow_{fr(M)} q(\tilde{r}[\tilde{t}_{i_1}, \dots, \tilde{t}_{i_l}]) && \text{(rule } \tilde{\mu}) \\
 &= q(\tilde{t}).
 \end{aligned}$$

Now we prove the other inclusion, namely that $h_{\Sigma_M}^{-1} \circ \tau_{fr(M),q} \circ h_{\Delta_M} \subseteq \tau_{M,q}$. To see this it is enough to show the following. Let $\tilde{s} \in T_{\Sigma_M}$ and $\tilde{t} \in T_{\Delta_M}$ such that $\tilde{s} \Rightarrow_{fr(M)}^* q(\tilde{t})$. Then there are trees $s \in T_{\Sigma}$ and $t \in T_{\Delta}$, such that $s = h_{\Sigma_M}(\tilde{s})$, $t = h_{\Delta_M}(\tilde{t})$ and $s \Rightarrow_M^* q(t)$. Again, we prove this by induction on the length n of the derivation $\tilde{s} \Rightarrow_{fr(M)}^* q(\tilde{t})$. If $n = 1$, then $\tilde{s} \rightarrow q(\tilde{t}) \in R_M$ (as an example of this case see the application of $\tilde{\mu}_9$ in the derivation in Case 2 of Example 3.42). Then, it follows from Definition 3.38, that there is a rule in $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r) \in R$ with $|\text{inf}(\mu)| = 0$ and, for every $i \in [k]$, there are trees $s_i \in \text{dom}(\tau_{M,q_i})$ and $t_i \in \text{ran}(\tau_{M,q_i})$ such that $\sigma(s_1, \dots, s_k) = h_{\Sigma_M}(\tilde{s})$ and $r[t_1, \dots, t_k] = h_{\Delta_M}(\tilde{t})$. Let $s = \sigma(s_1, \dots, s_k)$ and $t = r[t_1, \dots, t_k]$. Clearly $s \Rightarrow_M^* q(t)$ (cf. the subderivation $\gamma_1 \alpha \Rightarrow_{M_b}^* q_3(\omega \omega \beta_1)$ of (\dagger) in Example 2.4 and the application of μ_6 in it.)

Now let us suppose that the statement holds for $n > 1$. For the induction step from n to $n + 1$, let us assume that there is a derivation $\tilde{s} \Rightarrow_{fr(M)}^{n+1} q(\tilde{t})$. Let $\tilde{s} = \tilde{\sigma}(\tilde{s}_1, \dots, \tilde{s}_l)$, where $l \geq 1$, $\tilde{\sigma} \in \Sigma_M^{(l)}$ and $\tilde{s}_1, \dots, \tilde{s}_l \in T_{\Sigma_M}$. Then there is a rule $\tilde{\mu} = \tilde{\sigma}(\tilde{q}_1(x_1), \dots, \tilde{q}_l(x_l)) \rightarrow q(\tilde{r})$ in R_M such that the above derivation can be written in the following form.

$$\begin{aligned}
 \tilde{s} &= \tilde{\sigma}(\tilde{s}_1, \dots, \tilde{s}_l) \\
 &\Rightarrow_{fr(M)}^n \tilde{\sigma}(\tilde{q}_1(\tilde{t}_1), \dots, \tilde{q}_l(\tilde{t}_l)) \\
 &\Rightarrow_{fr(M)} q(\tilde{r}[\tilde{t}_1, \dots, \tilde{t}_l]) && \text{(rule } \tilde{\mu}) \\
 &= q(\tilde{t}),
 \end{aligned}$$

where $\tilde{t}_1, \dots, \tilde{t}_l \in T_{\Delta_M}$. Now we distinguish the following two cases.

Case 1: $l > 1$ or ($l = 1$ and $\tilde{q}_1 \in Q \cap Q_M$) (for instance the case of rule $\tilde{\mu}_2$ applied in the derivation appearing in Case 3 of Example 3.42, since for this rule $l = 2$). Then by Definition 3.38, there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R , where $k \geq l$, such that the following holds. Let $\text{inf}(\mu) = \{i_1, \dots, i_l\}$ such that $i_1 < i_2 < \dots < i_l$. Then, for every $j \in [l]$, $q_{i_j} = \tilde{q}_j$ and, for every $i \in [k]$, $i \notin \text{inf}(\mu)$, there are trees $s_i \in \text{dom}(\tau_{M,q_i})$ and $t_i \in \text{ran}(\tau_{M,q_i})$ such that $\sigma(\bar{s}_1, \dots, \bar{s}_k) = h_{\Sigma_M}(\tilde{\sigma})$ and $r[\bar{t}_1, \dots, \bar{t}_k] = h_{\Delta_M}(\tilde{r})$, where $\bar{s}_1, \dots, \bar{s}_k$ and $\bar{t}_1, \dots, \bar{t}_k$ are defined as follows. For every $j \in [l]$, $\bar{s}_{i_j} = \bar{t}_{i_j} = x_j$ and for every $i \in [k]$, $i \notin \text{inf}(\mu)$, $\bar{s}_i = s_i$ and $\bar{t}_i = t_i$.

Moreover, if $l > 1$, then it follows from Statement (ii) of Observation 3.40 that, for every $i \in [l]$, $\tilde{q}_i \in Q \cap Q_M$. Then, by induction hypothesis, for every $i \in [l]$, there is a derivation $h_{\Sigma_M}(\tilde{s}_i) \Rightarrow_M^* \tilde{q}_i(h_{\Delta_M}(\tilde{t}_i))$. Now let

$$s = \sigma(\bar{s}_1, \dots, \bar{s}_k)[h_{\Sigma_M}(\tilde{s}_1), \dots, h_{\Sigma_M}(\tilde{s}_l)] \text{ and } t = r[\bar{t}_1, \dots, \bar{t}_k][h_{\Delta_M}(\tilde{t}_1), \dots, h_{\Delta_M}(\tilde{t}_l)].$$

Clearly $s = h_{\Sigma_M}(\tilde{s})$, $t = h_{\Delta_M}(\tilde{t})$ and $s \Rightarrow_M^* q(t)$ (see the derivation (\dagger) in Example 2.4 and the rule μ_1 applied in it).

Case 2: $l = 1$ and $\tilde{q}_1 \notin Q \cap Q_M$ (for example as in the case of rule $\tilde{\mu}_6$ in the derivation in Case 1 of Example 3.42). Then, by Case 1.a of Definition 3.38, there are rules $\tilde{\sigma}_1(p_1(x_1)) \rightarrow p_0(x_1), \dots, \tilde{\sigma}_n(p_n(x_1)) \rightarrow p_{n-1}(x_1), \tilde{\sigma} \rightarrow p_n(\tilde{\delta})$ in R_M , where $p_0 = q$ and $p_1 = \tilde{q}_1$, such that $\tilde{s} = \tilde{\sigma}_1 \dots \tilde{\sigma}_n \tilde{\sigma}$ and $\tilde{t} = \tilde{\delta}$. Moreover, it follows from Definition 3.38, that there is a rule $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r) \in R$ such that $|\text{inf}(\mu)| = 0$ and, for every $i \in [k]$, there are trees $s_i \in \text{dom}(\tau_{M,q_i})$, $t_i \in \text{ran}(\tau_{M,q_i})$ such that $\sigma(s_1, \dots, s_k) = h_{\Sigma_M}(\tilde{s})$ and $r[t_1, \dots, t_k] = h_{\Delta_M}(\tilde{t})$. Let $s = \sigma(s_1, \dots, s_k)$ and $t = r[t_1, \dots, t_k]$. Then clearly $s \Rightarrow_M^* q(t)$. (For example see the subderivation $\sigma_1(\alpha, \alpha) \Rightarrow_{M_b}^* q_1(\beta_2)$ of (\dagger) of Example 3.17 and the rule μ_3 applied in it). ■

Since by Definition 3.38, a transformable tree transducer and its frame transducer have the same initial state, from Lemma 3.43 we immediately get the following corollary.

Corollary 3.44 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a transformable bottom-up tree transducer, and let $\text{fr}(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Then $\tau_M = h_{\Sigma_M}^{-1} \circ \tau_{\text{fr}(M)} \circ h_{\Delta_M}$. ■

Now we reconsider again the tree transducer of Example 2.4.

Example 3.45 Let M_b be the bottom-up tree transducer appearing in Example 2.4 and let, for every $n > 0$, a_n, b_n, c_n and d_n be the trees defined in that example. Let moreover $\text{fr}(M_b)$ be the frame transducer of M_b defined in Example 3.39. We leave the complete proof of that $\tau_{M_b} = h_{\Sigma_{M_b}}^{-1} \circ \tau_{\text{fr}(M_b)} \circ h_{\Delta_{M_b}}$ to the reader, however we provide some help for that proof. We have already shown in Example 2.4 that, for

every $n > 0$, $(a_n, b_n) \in \tau_{M_b, q_0}$ and $(c_n, d_n) \in \tau_{M_b, q_2}$. Let $\tilde{a} = \tilde{\sigma}_1(\tilde{\sigma}_2(x_1), \tilde{\sigma}_4\tilde{\sigma}_5)$ and $\tilde{b} = \tilde{\delta}_2(\tilde{\delta}_3(x_1), \tilde{\delta}_6\tilde{\delta}_7)$. Moreover, for every $n > 0$, let $\tilde{a}_n = \tilde{a}^n[\tilde{\sigma}_3]$, $\tilde{b}_n = \tilde{b}^n[\tilde{\delta}_4]$, $\tilde{c}_n = \tilde{\sigma}_4^n\tilde{\sigma}_5$ and $\tilde{d}_n = \tilde{\delta}_6^n\tilde{\delta}_8$. It is easy to see that, for every $n > 0$, $h_{\Sigma_{M_b}}(\tilde{a}_n) = a_n$, $h_{\Sigma_{M_b}}(\tilde{c}_n) = c_n$, $h_{\Delta_{M_b}}(\tilde{b}_n) = b_n$ and $h_{\Delta_{M_b}}(\tilde{d}_n) = d_n$. Moreover, $(\tilde{a}_n, \tilde{b}_n) \in \tau_{fr(M_b), q_0}$ and $(\tilde{c}_n, \tilde{d}_n) \in \tau_{fr(M_b), q_2}$. We note that, for every $n > 0$, not only $a_n \approx b_n$ and $c_n \approx d_n$ hold, but also $\tilde{a}_n \approx \tilde{b}_n$ and $\tilde{c}_n \approx \tilde{d}_n$. \square

The following lemma also shows a close relationship between a transformable bottom-up tree transducer $M = (Q, \Sigma, \Delta, q_0, R)$ and its frame transducer $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$. Let us consider two states $p, q \in Q \cap Q_M$ and trees $s \in T_\Sigma, t \in T_\Delta, u \in \hat{T}_\Sigma(X_1), v \in \hat{T}_\Delta(X_1)$ as well as trees $\tilde{s} \in T_{\Sigma_M}, \tilde{t} \in T_{\Delta_M}, \tilde{u} \in \hat{T}_{\Sigma_M}(X_1), \tilde{v} \in \hat{T}_{\Delta_M}(X_1)$ such that $h_{\Sigma_M}(\tilde{s}) = s, h_{\Sigma_M}(\tilde{u}) = u, h_{\Delta_M}(\tilde{t}) = t$ and $h_{\Delta_M}(\tilde{v}) = v$. While by Lemma 3.43 $s \Rightarrow_M^* p(t)$ if and only if $\tilde{s} \Rightarrow_{fr(M)}^* p(\tilde{t})$, the following lemma states that $\tilde{u}[p(x_1)] \Rightarrow_{fr(M)}^* q(\tilde{v})$ implies that $u[p(x_1)] \Rightarrow_M^* q(v)$.

Lemma 3.46 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a transformable bottom-up tree transducer, and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Let $\tilde{u} \in \hat{T}_{\Sigma_M}(X_1), \tilde{v} \in T_{\Delta_M}(X_1)$ and $p, q \in Q \cap Q_M$ such that $\tilde{u}[p(x_1)] \Rightarrow_{fr(M)}^* q(\tilde{v})$. Then $h_{\Sigma_M}(\tilde{u})[p(x_1)] \Rightarrow_M^* q(h_{\Delta_M}(\tilde{v}))$.

Proof. We prove the lemma by induction on the length n of the occurrence $occ(\tilde{u}, x_1)$. If $n = 0$, then the statement is trivial. Now let us suppose that the statement holds for $n > 0$. For the induction step from n to $n + 1$, let us assume that $length(occ(\tilde{u}, x_1)) = n + 1$. Then there are numbers $l \geq 1$ and $j \in [l]$ such that $\tilde{u} = \tilde{\sigma}(\tilde{s}_1, \dots, \tilde{s}_l)$, where $\tilde{\sigma} \in \Sigma_M^{(l)}$, for every $i \in [l], i \neq j, \tilde{s}_i \in T_{\Sigma_M}$, and $\tilde{s}_j \in \hat{T}_{\Sigma_M}(X_1)$. Moreover, there is a rule $\tilde{\mu} = \tilde{\sigma}(\tilde{q}_1(x_1), \dots, \tilde{q}_l(x_l)) \rightarrow q(\tilde{r})$ in R_M such that the following conditions hold. For every $i \in [l], i \neq j, \tilde{s}_i \Rightarrow_{fr(M)}^* \tilde{q}_i(\tilde{t}_i)$, where $\tilde{t}_i \in T_{\Delta_M}, \tilde{s}_j[p(x_1)] \Rightarrow_{fr(M)}^* \tilde{q}_j(\tilde{t}_j)$, where $\tilde{t}_j \in T_{\Delta_M}(X_1)$ and $\tilde{v} = \tilde{r}[\tilde{t}_1, \dots, \tilde{t}_l]$.

First we show that, for every $i \in [l]$ the state \tilde{q}_i is in Q . If $l > 1$, then this is true by Statement (ii) of Observation 3.40. If $l = 1$, then a straightforward computation, using Statements (ii) and (iii) of Observation 3.40 and that $p \in Q$, shows that $\tilde{q}_1 \in Q$.

Then, since $length(occ(\tilde{s}_j, x_1)) = n$, by induction hypothesis, $h_{\Sigma_M}(\tilde{s}_j)[p(x_1)] \Rightarrow_M^* \tilde{q}_j(h_{\Delta_M}(\tilde{t}_j))$. Moreover, by Lemma 3.43, for every $m \in [l], m \neq j$, there is a derivation $h_{\Sigma_M}(\tilde{s}_m) \Rightarrow_M^* \tilde{q}_m(h_{\Delta_M}(\tilde{t}_m))$. Furthermore, it follows from Definition 3.38, that there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R , where $k \geq l$, such that the following holds. Let $inf(\mu) = \{i_1, \dots, i_l\}$, where $i_1 < i_2 < \dots < i_l$. Then, $q_{i_1} = \tilde{q}_1, \dots, q_{i_l} = \tilde{q}_l$ and, for every $i \in [k], i \notin inf(\mu)$, there are trees $s_i \in dom(\tau_{M, q_i})$ and $t_i \in ran(\tau_{M, q_i})$ such that $\sigma(\tilde{s}_1, \dots, \tilde{s}_k) = h_{\Sigma_M}(\tilde{\sigma})$ and $r[\tilde{t}_1, \dots, \tilde{t}_k] = h_{\Delta_M}(\tilde{r})$, where $\tilde{s}_1, \dots, \tilde{s}_k$ and $\tilde{t}_1, \dots, \tilde{t}_k$ are defined as follows. For every $m \in [l], \tilde{s}_{i_m} = \tilde{t}_{i_m} = x_m$ and for every $i \in [k], i \notin inf(\mu)$, $\tilde{s}_i = s_i$ and $\tilde{t}_i = t_i$.

Now let, for every $m \in [l]$, $s_{i_m} = h_{\Sigma_M}(\tilde{s}_m)$ and $t_{i_m} = h_{\Delta_M}(\tilde{t}_m)$. Using that h_{Σ_M} and h_{Δ_M} are tree homomorphisms, we get that

$$\begin{aligned} \sigma(s_1, \dots, s_k) &= \sigma(\tilde{s}_1, \dots, \tilde{s}_k)[s_{i_1}, \dots, s_{i_l}] = \\ &h_{\Sigma_M}(\tilde{\sigma})[h_{\Sigma_M}(\tilde{s}_1), \dots, h_{\Sigma_M}(\tilde{s}_l)] = h_{\Sigma_M}(\tilde{\sigma}(\tilde{s}_1, \dots, \tilde{s}_l)) = h_{\Sigma_M}(\tilde{u}) \end{aligned}$$

and

$$\begin{aligned} r[t_1, \dots, t_k] &= r[\tilde{t}_1, \dots, \tilde{t}_k][t_{i_1}, \dots, t_{i_l}] = \\ &h_{\Delta_M}(\tilde{r})[h_{\Delta_M}(\tilde{t}_1), \dots, h_{\Delta_M}(\tilde{t}_l)] = h_{\Delta_M}(\tilde{r}[\tilde{t}_1, \dots, \tilde{t}_l]) = h_{\Delta_M}(\tilde{v}). \end{aligned}$$

Clearly $\sigma(s_1, \dots, s_k)[p(x_1)] \Rightarrow_M^* q(r[t_1, \dots, t_k])$ which proves our lemma. \blacksquare

3.2.3 Properties of the Frame Transducer of a Shape Preserving Bottom-Up Tree Transducer

From now on, whenever we consider a shape preserving bottom-up tree transducer M such that τ_M is infinite, by Lemma 3.36, we will assume that the frame transducer $fr(M)$ of M exists.

Let us consider a shape preserving bottom-up tree transducer $M = (Q, \Sigma, \Delta, q_0, R)$ and its frame transducer $fr(M)$. In this subsection we will show some important properties of $fr(M)$. For example we will show that $fr(M)$ is also shape preserving and if $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(\gamma\delta(\gamma_1x_1, \dots, \gamma_kx_k))$ is an arbitrary rule of $fr(M)$, then $h_{\Sigma_M}(\sigma) \approx h_{\Delta_M}(\delta)$.

First we show that $fr(M)$ is shape preserving, however for this we need some preparations. In fact we will show the following. Let $(\tilde{s}, \tilde{t}) \in \tau_{fr(M)}$. Since, by Statement (i) of Observation 3.40, $fr(M)$ is a quasirelabeling, it follows that if $fr(M)$ scans an input symbol $\tilde{\sigma}$, with rank different to one, during the derivation $\tilde{s} \Rightarrow_{fr(M)} q_0(\tilde{t})$, then it writes out exactly one output symbol $\tilde{\delta}$ with rank different to one. In the following two lemmas we show that the trees $h_{\Sigma_M}(\tilde{\sigma})$ and $h_{\Delta_M}(\tilde{\delta})$ are, roughly speaking, at the same occurrence in the trees $h_{\Sigma_M}(\tilde{s})$ and $h_{\Delta_M}(\tilde{t})$, respectively. The key for the proofs of these lemmas is that in Definition 3.38, for a rule μ of M such that $||inf(\mu)|| \neq 1$, we constructed the rules of the frame transducer using the matching path of μ .

Lemma 3.47 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Let $\tilde{\mu} = \tilde{\sigma}(\tilde{q}_1(x_1), \dots, \tilde{q}_l(x_l)) \rightarrow \tilde{q}(\tilde{\gamma}\tilde{\delta}(\tilde{\gamma}_1x_1, \dots, \tilde{\gamma}_lx_l))$ be a rule in R_M , where $l \neq 1$, $\tilde{q}_1, \dots, \tilde{q}_l, \tilde{q} \in Q \cap Q_M$, $\tilde{\delta} \in \Delta_M^{(l)}$ and $\tilde{\gamma}, \tilde{\gamma}_1, \dots, \tilde{\gamma}_l \in (\Delta_M^{(1)})^*$. Let

$$\begin{aligned}
(\dagger) \\
\tilde{s} &= \tilde{u}[\tilde{\sigma}(\tilde{s}_1, \dots, \tilde{s}_l)] \\
&\Rightarrow_{fr(M)}^* \tilde{u}[\tilde{\sigma}(\tilde{q}_1(\tilde{t}_1), \dots, \tilde{q}_l(\tilde{t}_l))] \\
&\Rightarrow_{fr(M)} \tilde{u}[\tilde{q}(\tilde{\gamma}\tilde{\delta}(\tilde{\gamma}_1\tilde{t}_1, \dots, \tilde{\gamma}_l\tilde{t}_l))] & (\text{rule } \tilde{\mu}) \\
&\Rightarrow_{fr(M)}^* q_0(\tilde{v}[\tilde{\gamma}\tilde{\delta}(\tilde{\gamma}_1\tilde{t}_1, \dots, \tilde{\gamma}_l\tilde{t}_l)]) \\
&= q_0(\tilde{t}),
\end{aligned}$$

be a derivation of $fr(M)$, where $\tilde{s} \in T_{\Sigma_M}$, $\tilde{u} \in \hat{T}_{\Sigma_M}(X_1)$, $\tilde{s}_1, \dots, \tilde{s}_l \in T_{\Sigma_M}$, $\tilde{t}_1, \dots, \tilde{t}_l \in T_{\Delta_M}$, $\tilde{v} \in \hat{T}_{\Delta_M}(X_1)$ and $\tilde{t} \in T_{\Delta_M}$. Then $\alpha = \beta\gamma$, where $\alpha = occ(h_{\Sigma_M}(\tilde{u}), x_1)$, $\beta = occ(h_{\Delta_M}(\tilde{v}), x_1)$ and $\gamma = occ(h_{\Delta_M}(\tilde{\gamma}), x_1)$.

Proof. We prove the statement by contradiction. Therefore, let us suppose that $\alpha \neq \beta\gamma$. Let $u = h_{\Sigma_M}(\tilde{u})$ and $v = h_{\Delta_M}(\tilde{v})$. Since $q_0, \tilde{q} \in Q$, by Lemma 3.46, there is a derivation $u[\tilde{q}(x_1)] \Rightarrow_M^* q_0(v)$. Now we distinguish the following two cases.

Case 1: $l = 0$. Then there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow \tilde{q}(r)$ in R such that $||inf(\mu)|| = 0$, $mp(\mu) = \gamma$ and γ is right matching path of μ (cf. Case 1.b of Definition 3.38). Clearly, there is a derivation of M of the following form.

$$\begin{aligned}
(\ddagger) \\
s &= u[\sigma(s_1, \dots, s_k)] \\
&\Rightarrow_M^* u[\sigma(q_1(t_1), \dots, q_k(t_k))] \\
&\Rightarrow_M u[\tilde{q}(r[t_1, \dots, t_k])] & (\text{rule } \mu) \\
&\Rightarrow_M^* q_0(v[r[t_1, \dots, t_k]]) \\
&= q_0(t),
\end{aligned}$$

where $s, s_1, \dots, s_k \in T_{\Sigma}$, and $t, t_1, \dots, t_k \in T_{\Delta}$. Since M is shape preserving and γ is a right matching path of μ , it follows from Case 1 of Definition 3.16 and Lemma 3.19 that $\alpha = \beta\gamma$, a contradiction.

Case 2: $l > 1$. Then there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow \tilde{q}(r)$ in R , where $k \geq l$, such that $||inf(\mu)|| = l$ and $\gamma = mp(\mu)$ (cf. Case 2 of Definition 3.38). Again, there is a derivation (\ddagger) of M as in Case 1. Since M is shape preserving, it follows from Lemma 3.22 that $\alpha = \beta\gamma$, which is again a contradiction. \blacksquare

Lemma 3.48 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be its frame transducer. Let $\tilde{\sigma}_1(p_1(x_1)) \rightarrow p_0(x_1), \dots, \tilde{\sigma}_n(p_n(x_1)) \rightarrow p_{n-1}(x_1)$ and $\tilde{\sigma} \rightarrow p_n(\tilde{\delta})$ be rules in R_M , where $n \geq 1$, $\tilde{\sigma}_1, \dots, \tilde{\sigma}_n \in \Sigma_M^{(1)}$, $\tilde{\sigma} \in \Sigma_M^{(0)}$, $\tilde{\delta} \in \Delta_M^{(0)}$, $p_1, \dots, p_n \in Q_M \setminus Q$ and $p_0 \in Q \cap Q_M$. Let

$$\begin{aligned}
\tilde{s} &= \tilde{u}[\tilde{\sigma}_1 \dots \tilde{\sigma}_n \tilde{\sigma}] \\
&\Rightarrow_{fr(M)}^{n+1} \tilde{u}[p_0(\tilde{\delta})] \\
&\Rightarrow_{fr(M)}^* q_0(\tilde{v}[\tilde{\delta}]) \\
&= q_0(\tilde{t}).
\end{aligned}$$

be a derivation of $fr(M)$, where $\tilde{s} \in T_{\Sigma_M}$, $\tilde{u} \in \hat{T}_{\Sigma_M}(X_1)$, $\tilde{v} \in \hat{T}_{\Delta_M}(X_1)$ and $\tilde{t} \in T_{\Delta_M}$. Then $\beta = \alpha\gamma$, where $\alpha = occ(h_{\Sigma_M}(\tilde{u}), x_1)$, $\beta = occ(h_{\Delta_M}(\tilde{v}), x_1)$ and $\gamma = occ(h_{\Sigma_M}(\tilde{\sigma}_1 \dots \tilde{\sigma}_n), x_1)$.

Proof. We prove the lemma by contradiction, so assume that $\beta \neq \alpha\gamma$. Let $u = h_{\Sigma_M}(\tilde{u})$ and $v = h_{\Delta_M}(\tilde{v})$. Since $q_0, p_0 \in Q$, by Lemma 3.46, there is a derivation $u[p_0(x_1)] \Rightarrow_M^* q_0(v)$. Moreover, there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow p_0(r)$ in R such that $\|inf(\mu)\| = 0$, $mp(\mu) = \gamma$ and γ is left matching path of μ (cf. Case 1.a of Definition 3.38). Clearly, there is a derivation of M of the following form.

$$\begin{aligned}
s &= u[\sigma(s_1, \dots, s_k)] \\
&\Rightarrow_M^* u[\sigma(q_1(t_1), \dots, q_k(t_k))] \\
&\Rightarrow_M u[p_0(r[t_1, \dots, t_k])] && (\text{rule } \mu) \\
&\Rightarrow_M^* q_0(v[r[t_1, \dots, t_k]]) \\
&= q_0(t),
\end{aligned}$$

where $s, s_1, \dots, s_l \in T_{\Sigma}$, and $t, t_1, \dots, t_l \in T_{\Delta}$. Since M is shape preserving and γ is a left matching path of μ , it follows from Case 1 of Definition 3.16 and Lemma 3.20 that $\beta = \alpha\gamma$. This is a contradiction, which proves our lemma. ■

Next we show that if $fr(M)$ scans an input symbol $\tilde{\sigma}$ and writes out an output symbol $\tilde{\delta}$ such that these symbols have rank greater than one, then the trees $h_{\Sigma_M}(\tilde{\sigma})$ and $h_{\Delta_M}(\tilde{\delta})$ have the variables at their leaves at the same occurrences.

Lemma 3.49 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be its frame transducer. Let $\tilde{\mu} = \tilde{\sigma}(\tilde{q}_1(x_1), \dots, \tilde{q}_l(x_l)) \rightarrow \tilde{q}(\tilde{\gamma}\tilde{\delta}(\tilde{\gamma}_1x_1, \dots, \tilde{\gamma}_lx_l))$ be a rule in R_M , where $l > 1$, $\tilde{\delta} \in \Delta_M^{(l)}$ and $\tilde{\gamma}, \tilde{\gamma}_1, \dots, \tilde{\gamma}_l \in (\Delta_M^{(1)})^*$, and let $u = h_{\Sigma_M}(\tilde{\sigma})$ and $v = h_{\Delta_M}(\tilde{\delta})$. Then, for every $j \in [l]$, $occ(u, x_j) = occ(v, x_j)$.

Proof. We prove the lemma by contradiction. Assume that there is an index $j \in [l]$, such that $occ(u, x_j) \neq occ(v, x_j)$. It follows from Definition 3.38 that there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R , where $k \geq l$ such that $\|inf(\mu)\| = l$ and the following holds. Let $inf(\mu) = \{i_1, \dots, i_l\}$, where $i_1 < i_2 < \dots < i_l$. Then there is no occurrence $\omega \in occ(stree(r, \gamma_{i_j}))$ such that $stree(r, \gamma_{i_j}\omega) = x_{i_j}$, where $\gamma = mp(\mu)$. However, by Corollary 3.15, there is exactly one occurrence $\bar{\omega} \in occ(r)$ such that $stree(r, \bar{\omega}) = x_{i_j}$. Furthermore, by Lemma 3.23, γ_{i_j} is a prefix of $\bar{\omega}$, which is clearly a contradiction. ■

Next we define the set of segments of a tree, which consists of those parts of the tree which are built up from unary symbols.

Definition 3.50 Let Σ be a ranked alphabet and $v \in T_\Sigma$. Let

$$F(v) = \{\gamma \in (\Sigma^{(1)})^+ \mid \text{there exist } s \in \widehat{T}_\Sigma(X_1) \text{ and } u \in T_\Sigma \text{ such that } s[\gamma u] = v\}.$$

A $\gamma \in F(v)$ is *maximal* if there is an $s \in \widehat{T}_\Sigma(X_1)$ and a $u \in T_\Sigma$ such that $s[\gamma u] = v$ and, for every $s' \in \widehat{T}_\Sigma(X_1)$, $\gamma' \in (\Sigma^{(1)})^*$ and $u' \in T_\Sigma$, if $\text{occ}(s', x_1)$ is a prefix of $\text{occ}(s, x_1)$ and u' is a subtree of u , and $s'[\gamma' u'] = v$, then $s = s'$, $\gamma = \gamma'$ and $u = u'$. \square

Now using our results, we can show the following. Let us consider two maximal segments $\tilde{\gamma}_s \in F(\tilde{s})$ and $\tilde{\gamma}_t \in F(\tilde{t})$ such that the tree transformed by $fr(M)$ from $\tilde{\gamma}_s$ is a part of the tree $\tilde{\gamma}_t$. Then $h_{\Sigma_M}(\tilde{\gamma}_s)$ and $h_{\Delta_M}(\tilde{\gamma}_t)$ are at the same occurrence in $h_\Sigma(\tilde{s})$ and $h_\Delta(\tilde{t})$, respectively. Moreover, the variable x_1 is at the same occurrence in $h_{\Sigma_M}(\tilde{\gamma}_s)$ and $h_{\Delta_M}(\tilde{\gamma}_t)$.

Let us consider for example the frame transducer $fr(M_b)$ defined in Example 3.39 and its derivation

$$\begin{aligned} & (\dagger) \\ \tilde{s} &= \tilde{u}[\tilde{\sigma}_4 \tilde{\sigma}_4 \tilde{\sigma}_4 \tilde{\sigma}_5] \\ &\Rightarrow_{fr(M_b)} \tilde{u}[\tilde{\sigma}_4 \tilde{\sigma}_4 \tilde{\sigma}_4 q_3(\tilde{\delta}_6 \tilde{\delta}_8)] \\ &\Rightarrow_{fr(M_b)}^* \tilde{u}[q_2(\tilde{\delta}_6 \tilde{\delta}_6 \tilde{\delta}_6 \tilde{\delta}_8)] \\ &\Rightarrow_{fr(M_b)}^* q_0(\tilde{v}[\tilde{\delta}_6 \tilde{\delta}_6 \tilde{\delta}_6 \tilde{\delta}_8]) \\ &= q_0(\tilde{t}), \end{aligned}$$

where \tilde{u} and \tilde{v} are the same as in Case 2 of Example 3.42. Here, the trees $\tilde{\sigma}_4 \tilde{\sigma}_4 \tilde{\sigma}_4$ and $\tilde{\delta}_6 \tilde{\delta}_6 \tilde{\delta}_6$ are maximal segments in $F(\tilde{s})$ and $F(\tilde{t})$, respectively. One can easily check that $\text{occ}(h_{\Sigma_{M_b}}(\tilde{u}), x_1) = \text{occ}(h_{\Delta_{M_b}}(\tilde{v}), x_1)$ and that $\text{occ}(h_{\Sigma_{M_b}}(\tilde{u}[\tilde{\sigma}_4 \tilde{\sigma}_4 \tilde{\sigma}_4]), x_1) = \text{occ}(h_{\Delta_{M_b}}(\tilde{v}[\tilde{\delta}_6 \tilde{\delta}_6 \tilde{\delta}_6]), x_1)$.

We generalize this observation formally in the next lemma. Then the fact that the frame transducer of a shape preserving bottom-up tree transducer is also shape preserving will be an easy consequence of this lemma.

Lemma 3.51 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving tree transducer such that τ_M is infinite and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Let $\tilde{s} \in \text{dom}(\tau_{fr(M)})$, $\tilde{\gamma}_s$ be a maximal segment in $F(\tilde{s})$, $\tilde{u}_1 \in \widehat{T}_{\Sigma_M}(X_1)$ and $\tilde{u}_2 \in T_{\Sigma_M}$ such that $\tilde{s} = \tilde{u}_1[\tilde{\gamma}_s \tilde{u}_2]$. Let

$$\begin{aligned}
(\dagger) \\
\tilde{s} &= \tilde{u}_1[\tilde{\gamma}_s \tilde{u}_2] \\
&\Rightarrow_{fr(M)}^* \tilde{u}_1[\tilde{\gamma}_s p(\tilde{\gamma}_t^{(3)} \tilde{v}_2)] \\
&\Rightarrow_{fr(M)}^* \tilde{u}_1[p'(\tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)} \tilde{v}_2)] \\
&\Rightarrow_{fr(M)}^* q_0(\tilde{v}_1[\tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)} \tilde{v}_2]) \\
&= q_0(\tilde{t})
\end{aligned}$$

be a derivation of $fr(M)$, where $\tilde{v}_1 \in \hat{T}_{\Delta_M}(X_1)$, $\tilde{t}, \tilde{v}_2 \in T_{\Delta_M}$ and $\tilde{\gamma}_t^{(1)}, \tilde{\gamma}_t^{(2)}, \tilde{\gamma}_t^{(3)} \in (\Delta_M^{(1)})^*$, such that $\tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)}$ is a maximal segment in $F(\tilde{t})$. Let $\tilde{\gamma}_t = \tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)}$, $\alpha_1 = occ(h_{\Sigma_M}(\tilde{u}_1), x_1)$, $\beta_1 = occ(h_{\Delta_M}(\tilde{v}_1), x_1)$, $\alpha_2 = occ(h_{\Sigma_M}(\tilde{u}_1[\tilde{\gamma}_s]), x_1)$ and $\beta_2 = occ(h_{\Delta_M}(\tilde{v}_1[\tilde{\gamma}_t]), x_1)$. Then $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$.

Proof. Without loss of generality, we assume that $\tilde{u}_1 \neq x_1$. (The case when $\tilde{u}_1 = x_1$ can be considered similarly, so we left its proof to the reader.) Since $\tilde{\gamma}_s$ is maximal, $\tilde{u}_1 = \tilde{u}'_1[\tilde{\sigma}_1(\tilde{s}_{11}, \dots, \tilde{s}_{1,i-1}, x_1, \tilde{s}_{1,i+1}, \dots, \tilde{s}_{1k})]$, where $k > 1$, $i \in [k]$, $\tilde{\sigma}_1 \in \Sigma_M^{(k)}$, $\tilde{u}'_1 \in \hat{T}_{\Sigma_M}(X_1)$ and, for every $j \in [k]$, $j \neq i$, $\tilde{s}_{1j} \in T_{\Sigma_M}$. Then there is a rule $\tilde{\mu}_1 = \tilde{\sigma}_1(\tilde{q}_{11}(x_1), \dots, \tilde{q}_{1k}(x_k)) \rightarrow \tilde{q}(\tilde{\gamma}_1 \tilde{\delta}_1(\tilde{\gamma}_{11} x_1, \dots, \tilde{\gamma}_{1k} x_k))$ in R_M , where $\tilde{\delta}_1 \in \Delta_M^{(k)}$ and $\tilde{\gamma}_1, \tilde{\gamma}_{11}, \dots, \tilde{\gamma}_{1k} \in (\Delta_M^{(1)})^*$, such that, for every $j \in [k]$, $j \neq i$, $\tilde{s}_{1j} \in dom(\tau_{fr(M), \tilde{q}_{1j}})$, $\tilde{q}_{1i} = p'$ and $\tilde{\gamma}_{1i} = \tilde{\gamma}_t^{(1)}$. Moreover, for every $j \in [k]$, $j \neq i$, there is a tree $\tilde{t}_{1j} \in ran(\tau_{fr(M), \tilde{q}_{1j}})$ such that $\tilde{v}_1 = \tilde{v}'_1[\tilde{\gamma}_1 \tilde{\delta}_1(\tilde{\gamma}_{11} \tilde{t}_{11}, \dots, x_1, \dots, \tilde{\gamma}_{1k} \tilde{t}_{1k})]$, for some $\tilde{v}'_1 \in T_{\Delta_M}(X_1)$. Furthermore, $\tilde{u}'_1[\tilde{q}(x_1)] \Rightarrow q_0(\tilde{v}'_1)$. Then the derivation (\dagger) can be written in the following form.

$$\begin{aligned}
\tilde{s} &= \tilde{u}'_1[\tilde{\sigma}_1(\tilde{s}_{11}, \dots, \tilde{\gamma}_s \tilde{u}_2, \dots, \tilde{s}_{1k})] \\
&\Rightarrow_{fr(M)}^* \tilde{u}'_1[\tilde{\sigma}_1(\tilde{q}_{11}(\tilde{t}_{11}), \dots, p'(\tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)} \tilde{v}_2), \dots, \tilde{q}_{1k}(\tilde{t}_{1k}))] \\
&\Rightarrow_{fr(M)} \tilde{u}'_1[\tilde{q}(\tilde{\gamma}_1 \tilde{\delta}_1(\tilde{\gamma}_{11} \tilde{t}_{11}, \dots, \tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)} \tilde{v}_2, \dots, \tilde{\gamma}_{1k} \tilde{t}_{1k}))] & (\text{rule } \tilde{\mu}_1) \\
&\Rightarrow_{fr(M)}^* q_0(\tilde{v}'_1[\tilde{\gamma}_1 \tilde{\delta}_1(\tilde{\gamma}_{11} \tilde{t}_{11}, \dots, \tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)} \tilde{v}_2, \dots, \tilde{\gamma}_{1k} \tilde{t}_{1k})]) \\
&= q_0(\tilde{t}).
\end{aligned}$$

By Statement (v) of Observation 3.40, $\tilde{v}'_1 \in \hat{T}_{\Delta_M}(X_1)$. Let $\alpha'_1 = occ(h_{\Sigma_M}(\tilde{u}'_1), x_1)$ and $\beta'_1 = occ(h_{\Delta_M}(\tilde{v}'_1), x_1) occ(h_{\Delta_M}(\tilde{\gamma}_1), x_1)$. Since $k > 1$, by Statement (ii) of Observation 3.40, the states $\tilde{q}, \tilde{q}_{11}, \dots, \tilde{q}_{1k}$ are in Q . Then by Lemma 3.47, $\alpha'_1 = \beta'_1$. Now, let $\omega \in occ(h_{\Sigma_M}(\tilde{\sigma}_1))$ such that $stree(h_{\Sigma_M}(\tilde{\sigma}_1), \omega) = x_i$. Then, by Lemma 3.49, $stree(h_{\Delta_M}(\tilde{\delta}_1), \omega) = x_i$ which implies that $\alpha_1 = \alpha'_1 \omega = \beta'_1 \omega = \beta_1$. Now, it remained to prove, that $\alpha_2 = \beta_2$. We distinguish the following two cases.

Case 1: $\tilde{u}_2 \notin \Sigma_M^{(0)}$. Then $\tilde{u}_2 = \tilde{\sigma}_2(\tilde{s}_{21}, \dots, \tilde{s}_{2l})$, where $l > 1$, $\tilde{\sigma}_2 \in \Sigma_M^{(l)}$ and $\tilde{s}_{21}, \dots, \tilde{s}_{2l} \in T_{\Sigma_M}$ (note that $\tilde{\gamma}_s$ is maximal segment). Moreover, there is a rule $\tilde{\mu}_2 = \tilde{\sigma}_2(\tilde{q}_{21}(x_1), \dots, \tilde{q}_{2l}(x_l)) \rightarrow p(\tilde{\gamma}_2 \tilde{\delta}_2(\tilde{\gamma}_{21} x_1, \dots, \tilde{\gamma}_{2l} x_l))$ in R_M , where $\tilde{\delta}_2 \in \Delta_M^{(l)}$ and $\tilde{\gamma}_2, \tilde{\gamma}_{21}, \dots, \tilde{\gamma}_{2l} \in (\Delta_M^{(1)})^*$, such that, for every $j \in [l]$, $\tilde{s}_{2j} \in dom(\tau_{fr(M), \tilde{q}_{2j}})$ and $\tilde{\gamma}_2 = \tilde{\gamma}_t^{(3)}$. Furthermore, for every $j \in [l]$, there is a tree $\tilde{t}_{2j} \in ran(\tau_{fr(M), \tilde{q}_{2j}})$ such that $\tilde{v}_2 = \tilde{\delta}_2(\tilde{\gamma}_{21} \tilde{t}_{21}, \dots, \tilde{\gamma}_{2l} \tilde{t}_{2l})$. Then the derivation (\dagger) can be written in the form

$$\begin{aligned}
\tilde{s} &= \tilde{u}_1[\tilde{\gamma}_s \tilde{\sigma}_2(\tilde{s}_{21}, \dots, \tilde{s}_{2l})] \\
&\Rightarrow_{fr(M)}^* \tilde{u}_1[\tilde{\gamma}_s \tilde{\sigma}_2(\tilde{q}_{21}(\tilde{t}_{21}), \dots, \tilde{q}_{2l}(\tilde{t}_{2l}))] \\
&\Rightarrow_{fr(M)} \tilde{u}_1[\tilde{\gamma}_s p(\tilde{\gamma}_t^{(3)} \tilde{\delta}_2(\tilde{\gamma}_{21} \tilde{t}_{21}, \dots, \tilde{\gamma}_{2l} \tilde{t}_{2l}))] & (\text{rule } \tilde{\mu}_2) \\
&\Rightarrow_{fr(M)}^* q_0(\tilde{v}_1[\tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)} \tilde{\delta}_2(\tilde{\gamma}_{21} \tilde{t}_{21}, \dots, \tilde{\gamma}_{2l} \tilde{t}_{2l})]) \\
&= q_0(\tilde{t}).
\end{aligned}$$

Since $l > 1$, by Statement (ii) of Observation 3.40, the states $p, \tilde{q}_{21}, \dots, \tilde{q}_{2l}$ are in Q . Then, by Lemma 3.47, it follows that $\alpha_2 = occ(h_{\Delta_M}(\tilde{v}_1[\tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)}]), x_1) occ(h_{\Delta_M}(\tilde{\gamma}_t^{(3)}), x_1)$. Using that h_{Δ_M} is a tree homomorphism, we get the desired equation

$$\alpha_2 = occ(h_{\Delta_M}(\tilde{v}_1[\tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)} \tilde{\gamma}_t^{(3)}]), x_1) = \beta_2.$$

Case 2: $\tilde{u}_2 \in \Sigma_M^{(0)}$. Then $\tilde{u}_2 = \tilde{\sigma}_2$, for some $\tilde{\sigma}_2 \in \Sigma_M^{(0)}$ and there is a rule $\tilde{\mu}_2 = \tilde{\sigma}_2 \rightarrow p(\tilde{\gamma}_2 \tilde{\delta}_2)$ in R_M , where $\tilde{\gamma}_2 = \tilde{\gamma}_t^{(3)}$, such that $\tilde{v}_2 = \tilde{\delta}_2$. Now there are the following two sub cases.

Case 2.a: $p \in Q$. Then the proof can be continued similarly as in Case 1.

Case 2.b: $p \notin Q$. Then $\tilde{\mu}_2$ has the form $\tilde{\sigma}_2 \rightarrow p(\tilde{\delta}_2)$ and there are rules $\tilde{\sigma}'_1(p_1(x_1)) \rightarrow p_0(x_1), \dots, \tilde{\sigma}'_n(p_n(x_1)) \rightarrow p_{n-1}(x_1)$ in R_M , where $n \geq 1$, $\tilde{\sigma}'_1, \dots, \tilde{\sigma}'_n \in \Sigma_M^{(1)}$, $p_1, \dots, p_{n-1} \in Q_M \setminus Q$ and $p_0 \in Q$ such that $\tilde{\gamma}_s = \tilde{\gamma}'_s \tilde{\sigma}'_1 \dots \tilde{\sigma}'_n$, for some $\tilde{\gamma}'_s \in (\Sigma_M^{(1)})^*$. Then the derivation (\dagger) can be written in the form

$$\begin{aligned}
\tilde{s} &= \tilde{u}_1[\tilde{\gamma}'_s \tilde{\sigma}'_1 \dots \tilde{\sigma}'_n \tilde{\sigma}_2] \\
&\Rightarrow_{fr(M)}^* \tilde{u}_1[\tilde{\gamma}'_s \tilde{\sigma}'_1 \dots \tilde{\sigma}'_n p(\tilde{\delta}_2)] & (\text{rule } \tilde{\mu}_2) \\
&\Rightarrow_{fr(M)}^* \tilde{u}_1[\tilde{\gamma}'_s p_0(\tilde{\delta}_2)] \\
&\Rightarrow_{fr(M)}^* q_0(\tilde{v}_1[\tilde{\gamma}_t^{(1)} \tilde{\gamma}_t^{(2)} \tilde{\delta}_2]) \\
&= q_0(\tilde{t}).
\end{aligned}$$

It follows by Lemma 3.48, that $occ(h_{\Sigma_M}(\tilde{u}_1[\tilde{\gamma}'_s]), x_1) occ(h_{\Sigma_M}(\tilde{\sigma}_1 \dots \tilde{\sigma}_n), x_1) = \beta_2$. This, since h_{Σ_M} is a tree homomorphism, implies that $\alpha_2 = occ(h_{\Sigma_M}(\tilde{u}_1[\tilde{\gamma}'_s \tilde{\sigma}_1 \dots \tilde{\sigma}_n]), x_1) = \beta_2$, which completes the proof of the lemma. \blacksquare

Now, we can show that the frame transducer of a shape preserving bottom-up tree transducer is also shape preserving.

Lemma 3.52 For every shape preserving bottom-up tree transducer M , if τ_M is infinite, then the frame transducer $fr(M)$ is also shape preserving.

Proof. Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Now, we show that $fr(M)$ is shape preserving by contradiction. Assume that there are trees $(\tilde{s}, \tilde{t}) \in \tau_{fr(M)}$ such that $\tilde{s} \not\approx \tilde{t}$. By Statement (i) of Observation 3.40, $fr(M)$ is

a quasirelabeling. Then it follows that \tilde{s} and \tilde{t} can differ from each other only in parts which are built up from unary symbols.

More formally, there is a maximal segment $\tilde{\gamma}_s \in F(\tilde{s})$, there are trees $\tilde{u}_1 \in \hat{T}_{\Sigma_M}(X_1)$ and $\tilde{u}_2 \in T_{\Sigma_M}$ with $\tilde{s} = \tilde{u}_1[\tilde{\gamma}_s\tilde{u}_2]$ and there is a derivation of $fr(M)$ as (\dagger) in Lemma 3.51 such that $length(\tilde{\gamma}_t) \neq length(\tilde{\gamma}_s)$, where $\tilde{\gamma}_t = \tilde{\gamma}_t^{(1)}\tilde{\gamma}_t^{(2)}\tilde{\gamma}_t^{(3)}$. (Note that by Statement (v) of Observation 3.40, we can assume that $\tilde{v}_1 \in \hat{T}_{\Delta_M}(X_1)$.)

Now, consider the occurrences $\alpha_1 = occ(h_{\Sigma_M}(\tilde{u}_1), x_1)$, $\alpha' = occ(h_{\Sigma_M}(\tilde{\gamma}_s), x_1)$ and $\alpha_2 = occ(h_{\Sigma_M}(\tilde{u}_1[\tilde{\gamma}_s]), x_1)$. Moreover, let $\beta_1 = occ(h_{\Delta_M}(\tilde{v}_1), x_1)$, $\beta' = occ(h_{\Delta_M}(\tilde{\gamma}_t), x_1)$ and $\beta_2 = occ(h_{\Delta_M}(\tilde{v}_1[\tilde{\gamma}_t]), x_1)$. Using that h_{Σ_M} and h_{Δ_M} are tree homomorphisms, we have that $\alpha_2 = occ(h_{\Sigma_M}(\tilde{u}_1[\tilde{\gamma}_s]), x_1) = occ(h_{\Sigma_M}(\tilde{u}_1)[h_{\Sigma_M}(\tilde{\gamma}_s)], x_1) = \alpha_1\alpha'$ and $\beta_2 = occ(h_{\Delta_M}(\tilde{v}_1[\tilde{\gamma}_t]), x_1) = occ(h_{\Delta_M}(\tilde{v}_1)[h_{\Delta_M}(\tilde{\gamma}_t)], x_1) = \beta_1\beta'$. Moreover, by Lemma 3.51, $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$ which implies that $\alpha' = \beta'$. Clearly $length(\alpha') = length(\beta')$. On the other hand, it is easy to see by induction that $length(\alpha') = length(\tilde{\gamma}_s)$ and $length(\beta') = length(\tilde{\gamma}_t)$, which, using that $length(\tilde{\gamma}_t) \neq length(\tilde{\gamma}_s)$, implies that $length(\alpha') \neq length(\beta')$. However, this is a contradiction, which proves that $fr(M)$ is shape preserving. \blacksquare

In the rest of this subsection we show two important properties of the frame transducer $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ of a shape preserving bottom-up tree transducer $M = (Q, \Sigma, \Delta, q_0, R)$. First we show that if $fr(M)$ scans an input symbol $\tilde{\sigma}$ and writes out an output symbol $\tilde{\delta}$ such that these symbols have rank different to one, then the trees $h_{\Sigma_M}(\tilde{\sigma})$ and $h_{\Delta_M}(\tilde{\delta})$ have the same shape.

Lemma 3.53 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Let $\tilde{\mu} = \tilde{\sigma}(\tilde{q}_1(x_1), \dots, \tilde{q}_l(x_l)) \rightarrow \tilde{q}(\tilde{\gamma}\tilde{\delta}(\tilde{\gamma}_1x_1, \dots, \tilde{\gamma}_lx_l))$ be a rule in R_M , where $l \neq 1$, $\tilde{\delta} \in \Delta_M^{(l)}$ and $\tilde{\gamma}, \tilde{\gamma}_1, \dots, \tilde{\gamma}_l \in (\Delta_M^{(1)})^*$. Then $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$.

Proof. According to l we distinguish the following three cases.

Case 1: $l = 0$ and $\tilde{q} \in Q$ (cf. for example the rule $\tilde{\mu}_9 = \tilde{\sigma}_5 \rightarrow q_3(\tilde{\delta}_6\tilde{\delta}_7)$ of $fr(M_b)$ defined in Example 3.39, since q_3 is a state of M_b , and note that, for the symbols $\tilde{\sigma}_5$ and $\tilde{\delta}_7$, which both have rank zero, $h_{\Sigma_{M_b}}(\tilde{\sigma}_5) \approx h_{\Delta_{M_b}}(\tilde{\delta}_7)$). By Statement (iv) of Observation 3.40, $\tau_{M, \tilde{q}}$ is infinite. Let $\gamma = occ(h_{\Delta_M}(\tilde{\gamma}), x_1)$. Then by Definition 3.38, there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow \tilde{q}(r)$ in R such that $||inf(\mu)|| = 0$, γ is right matching path of μ and the following holds. There is a derivation of M as (1) in Lemma 3.18 (note that by Corollary 3.13 we can assume that $v \in \hat{T}_{\Delta}(X_1)$) such that $h_{\Sigma_M}(\tilde{\sigma}) = stree(s, \alpha)$ and $h_{\Delta_M}(\tilde{\delta}) = stree(t, \beta\gamma)$, where $\alpha = occ(u, x_1)$ and $\beta = occ(v, x_1)$. Since γ is a right matching path of μ , it follows from Definition 3.16 and Lemma 3.19 that $\alpha = \beta\gamma$. Moreover, since M is shape preserving, $s \approx t$. Then clearly $stree(s, \alpha) \approx stree(t, \beta\gamma)$ which implies that $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$.

Case 2: $l = 0$ and $\tilde{q} \notin Q$ (cf. for instance the rule $\tilde{\mu}_5 = \tilde{\sigma}_3 \rightarrow p_1(\tilde{\delta}_4)$ of $fr(M_b)$, since p_1 is not a state of M_b , and note that $h_{\Sigma_{M_b}}(\tilde{\sigma}_3) \approx h_{\Delta_{M_b}}(\tilde{\delta}_4)$). Then, by Definition 3.38, $\tilde{\mu} = \tilde{\sigma} \rightarrow \tilde{q}(\tilde{\delta})$ and there are rules $\tilde{\sigma}_1(p_1(x_1)) \rightarrow p_0(x_1), \dots, \tilde{\sigma}_n(p_n(x_1)) \rightarrow p_{n-1}(x_1)$ in R_M , where $n \geq 1$, $\tilde{\sigma}_1, \dots, \tilde{\sigma}_n \in \Sigma_M^{(1)}$, $p_0 \in Q$, $p_1, \dots, p_{n-1} \in Q_M \setminus Q$ and $p_n = \tilde{q}$. By Statement (iv) of Observation 3.40, τ_{M,p_0} is infinite. Let $\gamma = occ(h_{\Sigma_M}(\tilde{\sigma}_1 \dots \tilde{\sigma}_n), x_1)$. Then, by Definition 3.38, there is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R such that $q = p_0$, $\|inf(\mu)\| = 0$, γ is a left matching path of μ and there is a derivation of M as (1) in Lemma 3.18 (note that by Corollary 3.13 we can assume that $v \in \hat{T}_\Delta(X_1)$) such that $h_{\Sigma_M}(\tilde{\sigma}) = stree(s, \alpha\gamma)$ and $h_{\Delta_M}(\tilde{\delta}) = stree(t, \beta)$, where $\alpha = occ(u, x_1)$ and $\beta = occ(v, x_1)$. Since γ is a left matching path of μ , it follows from Definition 3.16 and Lemma 3.20 that $\alpha\gamma = \beta$. Furthermore, since M is shape preserving, $s \approx t$. Then clearly $stree(s, \alpha\gamma) \approx stree(t, \beta)$, which implies that $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$.

Case 3: $l > 1$ (cf. e.g. the rule $\tilde{\mu}_1 = \tilde{\sigma}_1(q_1(x_1), q_2(x_2)) \rightarrow q_0(\tilde{\delta}_1(\tilde{\delta}_3(x_1), x_2))$ of $fr(M_b)$, where the symbols $\tilde{\sigma}_1$ and $\tilde{\delta}_1$, which are the only symbols with rank different to one, have the property that $h_{\Sigma_{M_b}}(\tilde{\sigma}_1) \approx h_{\Delta_{M_b}}(\tilde{\delta}_1)$). In this case we prove the lemma by contradiction. Assume that $h_{\Sigma_M}(\tilde{\sigma}) \not\approx h_{\Delta_M}(\tilde{\delta})$. By Statement (ii) of Observation 3.40, we have that $\tilde{q}, \tilde{q}_1, \dots, \tilde{q}_l \in Q$. Let $\tilde{u} = h_{\Sigma_M}(\tilde{\sigma})$, $\tilde{v} = h_{\Delta_M}(\tilde{\delta})$. Since Σ_M and Δ_M are tree alphabets, $\tilde{u} \in \hat{T}_\Sigma(X_l)$ and $\tilde{v} \in \hat{T}_\Delta(X_l)$. Moreover, by Lemma 3.49, for every $i \in [l]$, $occ(\tilde{u}, x_i) = occ(\tilde{v}, x_i)$. Then it follows that either (i) there is an occurrence in $\omega \in occ(\tilde{u})$ such that $\omega \notin occ(\tilde{v})$ or (ii) there is an occurrence in $\omega \in occ(\tilde{v})$ such that $\omega \notin occ(\tilde{u})$. Without loss of generality, assume that Case (i) holds (the proof can be continued similarly if Case (ii) holds). Clearly, for every $i \in [l]$, $occ(\tilde{v}, x_i)$ is not a prefix of ω . Moreover, there is a derivation (\dagger) of $fr(M)$ as in Lemma 3.47 (note that by Statement (v) of Observation 3.40, $\tilde{v} \in \hat{T}_{\Delta_M}(X_1)$). Let $u_1 = h_{\Sigma_M}(\tilde{u})$, $v_1 = h_{\Delta_M}(\tilde{v})$, $u_2 = h_{\Sigma_M}(\tilde{\sigma}(\tilde{s}_1, \dots, \tilde{s}_l))$ and $v_2 = h_{\Delta_M}(\tilde{\gamma}\tilde{\delta}(\tilde{\gamma}_1\tilde{t}_1, \dots, \tilde{\gamma}_l\tilde{t}_l))$. Then, by Lemma 3.43 and Lemma 3.46, there is a derivation

$$\begin{aligned} s &= u_1[u_2] \\ &\Rightarrow_M^* u_1[\tilde{q}(v_2)] \\ &\Rightarrow_M^* q_0(v_1[v_2]) \\ &= q_0(t), \end{aligned}$$

of M , where $s \in T_\Sigma$ and $t \in T_\Delta$. Let $\alpha = occ(u_1, x_1)$, $\beta = occ(v_1, x_1)$ and $\gamma = occ(h_{\Delta_M}(\tilde{\gamma}), x_1)$. Since M is shape preserving $s \approx t$. We will show that $\alpha\omega \in occ(s)$ while $\alpha\omega \notin occ(t)$ which is clearly a contradiction since $occ(s) = occ(t)$.

It is easy to see that $\alpha\omega \in occ(u_1[u_2]) = occ(s)$. Moreover, by Lemma 3.47, $\alpha = \beta\gamma$. Using that h_{Δ_M} is a tree homomorphism, we have that $stree(v_2, \gamma) = stree(h_{\Delta_M}(\tilde{\gamma})[h_{\Delta_M}(\tilde{\delta}(\tilde{\gamma}_1\tilde{t}_1, \dots, \tilde{\gamma}_l\tilde{t}_l))], \gamma) = h_{\Delta_M}(\tilde{\delta}(\tilde{\gamma}_1\tilde{t}_1, \dots, \tilde{\gamma}_l\tilde{t}_l))$. Then it can be seen that $\omega \notin occ(stree(v_2, \gamma))$. Thus we have that $\alpha\omega = \beta\gamma\omega \notin occ(v_1[v_2]) = occ(t)$ which completes the proof of the lemma. \blacksquare

Let us consider now two trees $(\tilde{s}, \tilde{t}) \in \tau_{fr(M)}$. In the next lemma we show that for two maximal segments $\tilde{\gamma}_s \in F(\tilde{s})$ and $\tilde{\gamma}_t \in F(\tilde{t})$ such that these segments are at the same occurrence in \tilde{s} and \tilde{t} , respectively, the condition $h_{\Sigma_M}(\tilde{\gamma}_s) \approx h_{\Delta_M}(\tilde{\gamma}_t)$ holds. For example consider the trees \tilde{s} and \tilde{t} appearing in the derivation (\dagger) discussed before Lemma 3.51. Clearly $\tilde{\sigma}_4\tilde{\sigma}_4\tilde{\sigma}_4$ and $\tilde{\delta}_6\tilde{\delta}_6\tilde{\delta}_6$ are maximal segments in $F(\tilde{s})$ and $F(\tilde{t})$, moreover they occur at the same occurrence in \tilde{s} and \tilde{t} , respectively. It is also easy to see that $h_{\Sigma_{M_b}}(\tilde{\sigma}_4\tilde{\sigma}_4\tilde{\sigma}_4) \approx h_{\Delta_{M_b}}(\tilde{\delta}_6\tilde{\delta}_6\tilde{\delta}_6)$.

Lemma 3.54 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite and let $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ be the frame transducer of M . Let $(\tilde{s}, \tilde{t}) \in \tau_{fr(M)}$ and $\tilde{\gamma}_s$ be a maximal segment in $F(\tilde{s})$. Let $\tilde{u}_1 \in \hat{T}_{\Sigma_M}(X_1)$, $\tilde{u}_2 \in T_{\Sigma_M}$, such that $\tilde{u} = \tilde{u}_1[\tilde{\gamma}_s\tilde{u}_2]$. Moreover let $\tilde{\gamma}_t \in (\Delta_M^{(1)})^*$, $\tilde{v}_1 \in \hat{T}_{\Delta_M}(X_1)$ and $\tilde{v}_2 \in T_{\Delta_M}$ such that $\tilde{t} = \tilde{v}_1[\tilde{\gamma}_t\tilde{v}_2]$, $occ(\tilde{u}_1, x_1) = occ(\tilde{v}_1, x_1)$ and $occ(\tilde{u}_1[\tilde{\gamma}_s], x_1) = occ(\tilde{v}_1[\tilde{\gamma}_t], x_1)$. Then $h_{\Sigma_M}(\tilde{\gamma}_s) \approx h_{\Delta_M}(\tilde{\gamma}_t)$.

Proof. We prove the lemma by contradiction. For this, let us assume that the conditions of the lemma hold but $h_{\Sigma_M}(\tilde{\gamma}_s) \not\approx h_{\Delta_M}(\tilde{\gamma}_t)$.

It is easy to see that there is a derivation (\ddagger) of $fr(M)$ as in Lemma 3.51 such that $\tilde{\gamma}_t = \tilde{\gamma}_t^{(1)}\tilde{\gamma}_t^{(2)}\tilde{\gamma}_t^{(3)}$. By Lemma 3.52 $fr(M)$ is shape preserving, which implies that $\tilde{\gamma}_t$ is a maximal segment in $F(\tilde{t})$. Let $\alpha_1 = occ(h_{\Sigma_M}(\tilde{u}_1), x_1)$, $\alpha' = occ(h_{\Sigma_M}(\tilde{\gamma}_s), x_1)$, $\alpha_2 = occ(h_{\Sigma_M}(\tilde{u}_1[\tilde{\gamma}_s]), x_1)$, $\beta_1 = occ(h_{\Delta_M}(\tilde{v}_1), x_1)$, $\beta' = occ(h_{\Delta_M}(\tilde{\gamma}_t), x_1)$ and $\beta_2 = occ(h_{\Delta_M}(\tilde{v}_1[\tilde{\gamma}_t]), x_1)$. By Lemma 3.51, $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$. Using that h_{Σ_M} and h_{Δ_M} are tree homomorphisms, we have that

$$\alpha_2 = occ(h_{\Sigma_M}(\tilde{u}_1[\tilde{\gamma}_s]), x_1) = occ(h_{\Sigma_M}(\tilde{u}_1)[h_{\Sigma_M}(\tilde{\gamma}_s)], x_1) = \alpha_1\alpha'$$

and

$$\beta_2 = occ(h_{\Delta_M}(\tilde{v}_1[\tilde{\gamma}_t]), x_1) = occ(h_{\Delta_M}(\tilde{v}_1)[h_{\Delta_M}(\tilde{\gamma}_t)], x_1) = \beta_1\beta'.$$

Then it follows that $\alpha' = \beta'$.

Now it is easy to see that either (i) there is an occurrence $\omega \in occ(h_{\Sigma_M}(\tilde{\gamma}_s))$ such that $\omega \notin occ(h_{\Delta_M}(\tilde{\gamma}_t))$ or (ii) there is an occurrence $\omega \in occ(h_{\Delta_M}(\tilde{\gamma}_t))$ such that $\omega \notin occ(h_{\Sigma_M}(\tilde{\gamma}_s))$. Without loss of generality, assume that Case (i) holds. (If Case (ii) holds, then the proof can be continued similarly.) Then β' and ω are not comparable, otherwise $\omega \in occ(h_{\Delta_M}(\tilde{\gamma}_t))$, which would contradict to our assumption.

Let $s = h_{\Sigma_M}(\tilde{s})$ and $t = h_{\Delta_M}(\tilde{t})$. Using that h_{Σ_M} and h_{Δ_M} are tree homomorphisms, we get that $s = h_{\Sigma_M}(\tilde{u}_1)[h_{\Sigma_M}(\tilde{\gamma}_s)[h_{\Sigma_M}(\tilde{u}_2)]]$ and $t = h_{\Delta_M}(\tilde{v}_1)[h_{\Delta_M}(\tilde{\gamma}_t)[h_{\Delta_M}(\tilde{v}_2)]]$. It can be seen, that $\omega \in occ(h_{\Sigma_M}(\tilde{\gamma}_s)[h_{\Sigma_M}(\tilde{u}_2)])$ and since β' and ω are not comparable, $\omega \notin occ(h_{\Delta_M}(\tilde{\gamma}_t)[h_{\Delta_M}(\tilde{v}_2)])$. Thus, since $\alpha_1 = \beta_1$, we have that $\alpha_1\omega \in occ(s)$ and $\alpha_1\omega = \beta_1\omega \notin occ(t)$, which implies that $s \not\approx t$. However, by Corollary 3.44, $(s, t) \in \tau_M$, and since M is shape preserving, $s \approx t$, a contradiction. \blacksquare

3.3 Characterizing the Shape Preserving Property

Here we prove the main result of this chapter, i.e., we show that, every shape preserving tree transducer is equivalent to a relabeling. Using this result we can easily characterize shape preserving tree transducers by relabelings.

3.3.1 The Top-Down Case

To show that part of the main result which concern top-down tree transducers it is enough to show that every shape preserving quasirelabeling is equivalent to a relabeling. Then it easily follows from this fact and from the results of Subsections 3.1.1 and 3.2.1 that shape preserving top-down tree transducers are equivalent to relabelings.

Again, we start our work with some preparations.

Definition 3.55 Let Σ and Δ be ranked alphabets.

a) We denote by $\langle \Sigma, \Delta \rangle$ the ranked alphabet defined by $\langle \Sigma, \Delta \rangle^{(k)} = \Sigma^{(k)} \times \Delta^{(k)}$, for every $k \geq 0$.

b) We denote by Σ_\diamond and Δ_\diamond the ranked alphabets which are obtained from Σ and Δ , respectively, by adding a new unary symbol \diamond to both $\Sigma^{(1)}$ and $\Delta^{(1)}$.

c) Let $\bar{h}_\Sigma : T_{\langle \Sigma_\diamond, \Delta_\diamond \rangle} \rightarrow T_\Sigma(X)$ and $\bar{h}_\Delta : T_{\langle \Sigma_\diamond, \Delta_\diamond \rangle} \rightarrow T_\Delta(X)$ defined as follows. For every $k \geq 0$ and $\langle \sigma, \delta \rangle \in \langle \Sigma_\diamond, \Delta_\diamond \rangle^{(k)}$, let

$$\begin{aligned} \bar{h}_\Sigma(\langle \sigma, \delta \rangle) &= \begin{cases} \sigma(x_1, \dots, x_k) & \text{if } \sigma \neq \diamond \\ x_1 & \text{otherwise,} \end{cases} \\ \bar{h}_\Delta(\langle \sigma, \delta \rangle) &= \begin{cases} \delta(x_1, \dots, x_k) & \text{if } \delta \neq \diamond \\ x_1 & \text{otherwise.} \end{cases} \end{aligned}$$

We note that in this definition $\sigma = \diamond$ (resp. $\delta = \diamond$) implies $\delta \in \Delta_\diamond^{(1)}$ (resp. $\sigma \in \Sigma_\diamond^{(1)}$). Now let $h_\Sigma : T_{\langle \Sigma_\diamond, \Delta_\diamond \rangle} \rightarrow T_\Sigma$ and $h_\Delta : T_{\langle \Sigma_\diamond, \Delta_\diamond \rangle} \rightarrow T_\Delta$ be the tree homomorphisms induced by \bar{h}_Σ and \bar{h}_Δ , respectively. \square

Observation 3.56 For every $s \in T_{\langle \Sigma, \Delta \rangle}$, $h_\Sigma(s) \approx h_\Delta(s)$. Hence, for every $L \subseteq T_{\langle \Sigma, \Delta \rangle}$, the tree transformation $h_\Sigma^{-1} \circ Id(L) \circ h_\Delta$ is shape preserving, where $Id(L) = \{(s, s) \mid s \in L\}$. The same does not hold with the ranked alphabet $\langle \Sigma_\diamond, \Delta_\diamond \rangle$. \blacksquare

Lemma 3.57 For every quasirelabeling $M = (Q, \Sigma, \Delta, q_0, R)$ a top-down tree automaton $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, p_0, R_T)$ can be constructed such that $\tau_M = h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$.

Proof. For every rule

$$\mu = q(\sigma(x_1, \dots, x_k)) \rightarrow \gamma\delta(\gamma_1 q_1(x_1), \dots, \gamma_k q_k(x_k)) \in R,$$

where $k \neq 1$, $q, q_1, \dots, q_k \in Q$, $\sigma \in \Sigma^{(k)}$, $\delta \in \Delta^{(k)}$, and $\gamma, \gamma_1, \dots, \gamma_k \in (\Delta^{(1)})^*$ with $length(\gamma) = n$ and $length(\gamma_1) = n_1, \dots, length(\gamma_k) = n_k$, construct the rules

$$\begin{aligned}
q(\langle \diamond, \gamma(1) \rangle(x_1)) &\rightarrow \langle \diamond, \gamma(1) \rangle(p_1(x_1)), \\
&\dots \\
p_{n-1}(\langle \diamond, \gamma(n) \rangle(x_1)) &\rightarrow \langle \diamond, \gamma(n) \rangle(p_n(x_1)), \\
p_n(\langle \sigma, \delta \rangle(x_1, \dots, x_k)) &\rightarrow \langle \sigma, \delta \rangle(p_{11}(x_1), \dots, p_{k1}(x_k)), \tag{†}
\end{aligned}$$

as well as, for every $1 \leq j \leq k$, construct the rules

$$\begin{aligned}
p_{j1}(\langle \diamond, \gamma_j(1) \rangle(x_1)) &\rightarrow \langle \diamond, \gamma_j(1) \rangle(p_{j2}(x_1)), \\
&\dots \\
p_{jn_j}(\langle \diamond, \gamma_j(n_j) \rangle(x_1)) &\rightarrow \langle \diamond, \gamma_j(n_j) \rangle(q_j(x_1)),
\end{aligned}$$

where p_1, \dots, p_n and p_{j1}, \dots, p_{jn_j} are new states. (In case $n = 0$ we mean $p_n = q$ in the rule (†) and there are no rules above (†). In case $n_j = 1$, we have the only rule $p_{j1}(\langle \diamond, \gamma_j(1) \rangle(x_1)) \rightarrow \langle \diamond, \gamma_j(1) \rangle(q_j(x_1))$, while in case $n_j = 0$, p_{j1} is meant to be q_j in the rule (†) and we do not need further rules for the index j .)

Let R_μ be the set of all rules constructed from μ and Q_μ be the set of all states appearing in those rules.

Moreover, for every rule $\mu = q(\sigma(x_1)) \rightarrow \gamma p(x_1)$ in R , where $q, p \in Q$, $\sigma \in \Sigma^{(1)}$ and $\gamma \in (\Delta^{(1)})^*$ with $\text{length}(\gamma) = n$, construct the rules

$$\begin{aligned}
q(\langle \diamond, \gamma(1) \rangle(x_1)) &\rightarrow \langle \diamond, \gamma(1) \rangle(p_1(x_1)), \\
&\dots \\
p_{n-1}(\langle \sigma, \gamma(n) \rangle(x_1)) &\rightarrow \langle \sigma, \gamma(n) \rangle(p(x_1)),
\end{aligned}$$

where p_1, \dots, p_{n-1} are new states. (In case $n = 1$, we have the only rule $q(\langle \sigma, \gamma(1) \rangle(x_1)) \rightarrow \langle \sigma, \gamma(1) \rangle(p(x_1))$, while in case $n = 0$, we have the only rule $q(\langle \sigma, \diamond \rangle(x_1)) \rightarrow \langle \sigma, \diamond \rangle(p(x_1))$.) Again, let R_μ be the set of all rules constructed from μ and Q_μ be the set of all states appearing in those rules.

Now let $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, p_0, R_T)$ be the top-down tree automaton, where $P = \bigcup_{\mu \in R} Q_\mu$, $p_0 = q_0$ and $R_T = \bigcup_{\mu \in R} R_\mu$. It should be clear that $\tau_M = h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$. ■

We will need the following definition.

Definition 3.58 Let Σ and Δ be ranked alphabets. For a tree language $L \subseteq T_{\langle \Sigma_\diamond, \Delta_\diamond \rangle}$, we define $F(L) = \bigcup_{v \in L} F(v)$ (cf. Definition 3.50). The tree language L is k -bounded, for $k \geq 0$, if, for every $\gamma \in F(L)$, the approximation $|\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| \leq k$ holds. Moreover, L is bounded if it is k -bounded for some k . □

Now we prove a technical lemma. Its proof relies on the simple idea that some states of a finite tree automaton necessarily repeat on sufficiently long paths (and thus segments) of a tree.

Lemma 3.59 Let Σ and Δ be ranked alphabets and $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, p_0, R_T)$ be a top-down tree automaton such that the tree transformation $\tau = h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ is shape preserving. Then $L(T)$ is bounded.

Proof. We prove by contradiction that $L(T)$ is k -bounded, where $k = \|P\|$. Assume that $L(T)$ is not k -bounded, let

$$L_k = \{\gamma \in F(L(T)) \mid |\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| > k\}$$

and let γ be an element of L_k with minimal length. Then also $\text{length}(\gamma) > k$. Moreover, for every $0 \leq i \leq \text{length}(\gamma)$, let $\beta_i, \gamma_i \in (\langle \Sigma_\diamond, \Delta_\diamond \rangle^{(1)})^*$ be such that $\text{length}(\beta_i) = i$ and $\gamma = \beta_i \gamma_i$.

Since $\gamma \in F(L(T))$ and $\text{length}(\gamma) > k$, there are $s \in \widehat{T}_{\langle \Sigma_\diamond, \Delta_\diamond \rangle}(X_1)$, $u \in T_{\langle \Sigma_\diamond, \Delta_\diamond \rangle}$, $q, q' \in P$ and $0 \leq i < j \leq \text{length}(\gamma)$ such that $s[\gamma u] \in L(T)$ and

$$p_0(s[\gamma u]) \Rightarrow_T^* s[q(\gamma u)] \Rightarrow_T^* s[\beta_i q'(\gamma_i u)] \Rightarrow_T^* s[\beta_j q'(\gamma_j u)] \Rightarrow_T^* s[\gamma u].$$

Let $\beta_{ij} \in (\langle \Sigma_\diamond, \Delta_\diamond \rangle^{(1)})^+$ be such that $\beta_i \beta_{ij} = \beta_j$ and thus $\beta_i \beta_{ij} \gamma_j = \gamma$. Then, for every $l \geq 0$, $s[\beta_i \beta_{ij}^l \gamma_j u] \in L(T)$ hence $\beta_i \beta_{ij}^l \gamma_j \in F(L(T))$.

Now $\text{length}(h_\Sigma(\beta_{ij})) \neq \text{length}(h_\Delta(\beta_{ij}))$ because otherwise the tree $\gamma' = \beta_i \gamma_j \in F(L(T))$ also satisfies $|\text{length}(h_\Sigma(\gamma')) - \text{length}(h_\Delta(\gamma'))| > k$ thus also $\gamma' \in L_k$. This, however, is impossible because $\text{length}(\gamma') < \text{length}(\gamma)$ and γ is an element of L_k with minimal length.

Thus we can assume that, say, $\text{length}(h_\Sigma(\beta_{ij})) > \text{length}(h_\Delta(\beta_{ij}))$. Then, for a sufficiently big l ,

$$\text{height}(h_\Sigma(s[\beta_i \beta_{ij}^l \gamma_j u])) > \text{height}(h_\Delta(s[\beta_i \beta_{ij}^l \gamma_j u])),$$

which contradicts the fact, that τ is shape preserving. Hence $L(T)$ is k -bounded. \blacksquare

The following definition is a key to construct the relabeling equivalent to a shape preserving quasirelabeling. For a top-down tree automaton $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, q_0, R_T)$ such that $L(T)$ is k -bounded we give another top-down tree automaton $T' = (P', \langle \Sigma, \Delta \rangle, p'_0, R_{T'})$ such that $h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$. Moreover, if $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ is shape preserving, then $h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ also holds. This automaton T' will be given with the help of a top-down tree transducer $N = (P_N, \langle \Sigma_\diamond, \Delta_\diamond \rangle, \langle \Sigma, \Delta \rangle, p_0, R_N)$.

Intuitively, N does nothing else but throws away the \diamond symbols from the input trees as follows. The states of N work like a kind of puffer of symbols in $\Sigma^{(1)}$ and $\Delta^{(1)}$. In fact, every state p in P_N is a two component structure such that either the first component of p is ε and the second component is a word from $(\Delta^{(1)})^*$ or the first component is a word from $(\Sigma^{(1)})^*$ and the second component is ε , moreover the length of the words which are stored in the states is at most k . The initial state p_0 of N is

a state of which both components are ε . N can scan an input symbol $\langle \sigma, \delta \rangle \in \langle \Sigma, \Delta \rangle$ with rank different to one if and only if it is in the state p_0 and in this case it writes out $\langle \sigma, \delta \rangle$. Now assume that N is scanning a maximal segment γ of an input tree and a prefix γ' of γ is already processed by N . Assume also that N is in the state $p = (u, v)$, where $u \in (\Sigma^{(1)})^*$ and $v \in (\Delta^{(1)})^*$. If, for example, $\text{length}(h_\Sigma(\gamma')) < \text{length}(h_\Delta(\gamma'))$, then v is that suffix of $h_\Delta(\gamma')$ by which $h_\Delta(\gamma')$ is longer than $h_\Sigma(\gamma')$ and in this case $u = \varepsilon$. Now, if in this situation N scans an input symbol $\langle \sigma, \delta \rangle$, where $\sigma \in \Sigma^{(1)}$ and $\delta \in \Delta_\diamond^{(1)}$, then it writes out the symbol $\langle \sigma, \delta' \rangle$, where δ' is the first letter of v . The fact that $L(T)$ is k -bounded ensures that no overflowing of the words stored in the states happens during the work of N . The formal definitions of T' and N are as follows.

Definition 3.60 Let $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, q_0, R_T)$ be a top-down tree automaton such that $L(T)$ is k -bounded. The *shape preserving frame* of T is the top-down tree automaton $T' = (P', \langle \Sigma, \Delta \rangle, p'_0, R_{T'})$ constructed in the following way. First we construct a linear top-down tree transducer $N = (P_N, \langle \Sigma_\diamond, \Delta_\diamond \rangle, \langle \Sigma, \Delta \rangle, p_0, R_N)$ as follows. Let

- $P_N = (\Sigma^{(1)})^{*,k} \times \{\varepsilon\} \cup \{\varepsilon\} \times (\Delta^{(1)})^{*,k}$,
- $p_0 = [\varepsilon, \varepsilon]$, (We use the brackets $[$ and $]$ to denote elements of P_N .)
- R_N is the smallest set of rules satisfying the following conditions.
 - For every m with $m \neq 1$, $\sigma \in \Sigma^{(m)}$ and $\delta \in \Delta^{(m)}$, the rule $[\varepsilon, \varepsilon](\langle \sigma, \delta \rangle(x_1, \dots, x_m)) \rightarrow \langle \sigma, \delta \rangle([\varepsilon, \varepsilon](x_1), \dots, [\varepsilon, \varepsilon](x_m))$ is in R_N .
 - For every $u \in (\Sigma^{(1)})^{*,k-1}$ and $\beta \in \Sigma^{(1)}$, the rule $[u, \varepsilon](\langle \beta, \diamond \rangle(x_1)) \rightarrow [u\beta, \varepsilon](x_1)$ is in R_N .
 - For every $\beta, \beta' \in \Sigma^{(1)}$, $u \in (\Sigma^{(1)})^{*,k-1}$ and $\gamma \in \Delta^{(1)}$, the rule $[\beta'u, \varepsilon](\langle \beta, \gamma \rangle(x_1)) \rightarrow \langle \beta', \gamma \rangle[u\beta, \varepsilon](x_1)$ is in R_N .
 - For every $\beta \in \Sigma^{(1)}$, $u \in (\Sigma^{(1)})^{*,k-1}$ and $\gamma \in \Delta^{(1)}$, the rule $[\beta u, \varepsilon](\langle \diamond, \gamma \rangle(x_1)) \rightarrow \langle \beta, \gamma \rangle[u, \varepsilon](x_1)$ is in R_N .
 - For every $v \in (\Delta^{(1)})^{*,k-1}$ and $\gamma \in \Delta^{(1)}$, the rule $[\varepsilon, v](\langle \diamond, \gamma \rangle(x_1)) \rightarrow [\varepsilon, v\gamma](x_1)$ is in R_N .
 - For every $\gamma, \gamma' \in \Delta^{(1)}$, $v \in (\Delta^{(1)})^{*,k-1}$ and $\beta \in \Sigma^{(1)}$, the rule $[\varepsilon, \gamma'v](\langle \beta, \gamma \rangle(x_1)) \rightarrow \langle \beta, \gamma' \rangle[\varepsilon, v\gamma](x_1)$ is in R_N .
 - For every $\gamma' \in \Delta^{(1)}$, $v \in (\Delta^{(1)})^{*,k-1}$ and $\beta \in \Sigma^{(1)}$, the rule $[\varepsilon, \gamma'v](\langle \beta, \diamond \rangle(x_1)) \rightarrow \langle \beta, \gamma' \rangle[\varepsilon, v](x_1)$ is in R_N .

Now let $L = \text{ran}(\tau_T \circ \tau_N)$. Since linear top-down tree transducers preserve recognizability [Rou70] (cf. also Corollary IV.6.6 in [GS84]), L is recognizable, hence a top-down tree automaton $T' = (P', \langle \Sigma, \Delta \rangle, p'_0, R_{T'})$ can be constructed such that $L = \text{dom}(\tau_{T'})$.

Let T and T' be the tree transducers appearing in the discussion before Definition 3.60. In the next two lemmas we show that if $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ is shape preserving then $h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta = h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$.

Lemma 3.61 Let $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, q_0, R_T)$ be a top-down tree automaton such that $L(T)$ is k -bounded and $T' = (P', \langle \Sigma, \Delta \rangle, p'_0, R_{T'})$ is the shape preserving frame of T . Then $h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ is shape preserving and $h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$.

Proof. Since T' recognizes trees over the ranked alphabet $\langle \Sigma, \Delta \rangle$, by Observation 3.56, the tree transformation $h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ is shape preserving.

Since $L(T)$ is k -bounded, for every maximal $\gamma \in F(L(T))$ and prefix γ' of γ , the approximation $|\text{length}(h_\Sigma(\gamma')) - \text{length}(h_\Delta(\gamma'))| \leq k$ holds. Now the piece of string of length at most k with which $h_\Sigma(\gamma')$ is “ahead” of $h_\Delta(\gamma')$ (resp. $h_\Delta(\gamma')$ is ahead of $h_\Sigma(\gamma')$) is stored in the first (resp. second) component of the states of N .

Moreover, N is able to process an input symbol $\langle \sigma, \delta \rangle \in \langle \Sigma_\diamond, \Delta_\diamond \rangle^{(m)}$ with $m \neq 1$ only in state $[\varepsilon, \varepsilon]$. Hence it should be clear that N has the following property. For every $v \in \text{ran}(\tau_T)$, the inclusion $v \in \text{dom}(\tau_N)$ holds, if and only if, for every maximal $\gamma \in F(v)$, $|\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| = 0$. Moreover, if this is the case, then $h_\Sigma(v) = h_\Sigma(v')$ and $h_\Delta(v) = h_\Delta(v')$, where $v' = \tau_N(v)$.

Now we can show that $h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ in the following way. Let $(s, t) \in h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$. Then, there is a $v \in \text{ran}(\tau_T) \cap \text{dom}(\tau_N)$ such that, for $v' = \tau_N(v)$, we have $h_\Sigma(v') = s$ and $h_\Delta(v') = t$. Since $v \in \text{ran}(\tau_T) \cap \text{dom}(\tau_N)$, by the above note $h_\Sigma(v) = h_\Sigma(v')$ and $h_\Delta(v) = h_\Delta(v')$. Hence, also $(s, t) \in h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$. ■

Lemma 3.62 Let $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, q_0, R_T)$ be a top-down tree automaton such that $L(T)$ is k -bounded and $T' = (P', \langle \Sigma, \Delta \rangle, p'_0, R_{T'})$ is the shape preserving frame of T . If $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ is shape preserving, then $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$.

Proof. Let $(s, t) \in h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$, then there is a $v \in \text{ran}(\tau_T)$ such that $h_\Sigma(v) = s$ and $h_\Delta(v) = t$. Since $s \approx t$, for every maximal $\gamma \in F(v)$, $|\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| = 0$ holds. Then, by the proof of Lemma 3.61, $v \in \text{dom}(\tau_N)$ and $h_\Sigma(v) = h_\Sigma(v')$ and $h_\Delta(v) = h_\Delta(v')$, where $v' = \tau_N(v)$. This means $(s, t) \in h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$. ■

As the last step of the preparation we state an obvious fact.

Lemma 3.63 Let $T = (P, \langle \Sigma, \Delta \rangle, p_0, R_T)$ be a top-down tree automaton. Then there is a relabeling $M = (P, \Sigma, \Delta, p_0, R_M)$ such that $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta = \tau_M$.

Proof. R_M is constructed as follows. For every rule $q(\langle\sigma, \delta\rangle(x_1, \dots, x_k)) \rightarrow \langle\sigma, \delta\rangle(q_1(x_1), \dots, q_k(x_k))$ in R_T , let the rule $q(\sigma(x_1, \dots, x_k)) \rightarrow \delta(q_1(x_1), \dots, q_k(x_k))$ be in R_M . It should be clear that $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta = \tau_M$. ■

Now we can state one of the main results of this chapter.

Theorem 3.64 *Every shape preserving top-down tree transducer is equivalent to a top-down relabeling tree transducer.*

Proof. It follows from Lemmas 3.5, 3.31, 3.57, 3.59, 3.61, 3.62, and 3.63. ■

3.3.2 The Bottom-Up Case

In this subsection we show that every shape preserving bottom-up tree transducer M is equivalent to a relabeling. First we give the relabeling frame transducer $rfr(M)$ of M which is a relabeling equivalent to $fr(M)$. Then we show certain properties of $rfr(M)$ which we will use to finish our work.

The formal definition of relabeling frame transducers is as follows.

Definition 3.65 Let M be a transformable bottom-up tree transducer such that $fr(M)$, the frame transducer of M , is shape preserving. By Statement (i) of Observation 3.40, $fr(M)$ is a bottom-up quasirelabeling tree transducer. Then there is a top-down quasirelabeling M' equivalent to $fr(M)$ (cf. the discussion after the definition of quasirelabelings in Chapter 2). Moreover, by Theorem 3.64, there is a top-down relabeling tree transducer M'' equivalent to M' . Finally, there is a bottom-up relabeling tree transducer $rfr(M)$, such that $\tau_{M''} = \tau_{rfr(M)}$. Then, clearly, $\tau_{fr(M)} = \tau_{rfr(M)}$. This $rfr(M)$ is called the *relabeling frame transducer* of M . □

In the rest of the Thesis, if we consider a shape preserving bottom-up tree transducer M and its frame transducer $fr(M)$, then, since by Lemma 3.52 $fr(M)$ is shape preserving, we will assume that the relabeling frame transducer $rfr(M)$ exists.

Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer, $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ its frame transducer and $rfr(M) = (Q', \Sigma_M, \Delta_M, q'_0, R')$ its relabeling frame transducer. In the following two lemmas we show that if $rfr(M)$ scans an input symbol $\tilde{\sigma} \in \Sigma_M$ and writes out an output symbol $\tilde{\delta} \in \Delta_M$, then $h_{\Sigma_M}(\tilde{\sigma})$ and $h_{\Delta_M}(\tilde{\delta})$ have the same shape.

The next lemma concerns the case when $\tilde{\sigma}$ and $\tilde{\delta}$ have rank different to one. Let $\mu = \tilde{\sigma}(q_1(x_1), \dots, q_l(x_l)) \rightarrow q(\tilde{\delta}(x_1, \dots, x_l))$ be a rule in R' . In the proof of the lemma we will show that there is a rule $\tilde{\mu} = \tilde{\sigma}(\tilde{q}_1(x_1), \dots, \tilde{q}_l(x_l)) \rightarrow \tilde{q}(\tilde{\gamma}\tilde{\delta}(\tilde{\gamma}_1x_1, \dots, \tilde{\gamma}_lx_l)) \in R_M$ which, by Lemma 3.53, easily implies that $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$.

Lemma 3.66 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite, $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ its frame transducer, and $rfr(M) = (Q', \Sigma_M, \Delta_M, q'_0, R')$ the relabeling frame transducer of M . Let $\mu = \tilde{\sigma}(q_1(x_1), \dots, q_l(x_l)) \rightarrow q(\tilde{\delta}(x_1, \dots, x_l))$ be a rule in R' , where $l \neq 1$ and $\tilde{\delta} \in \Delta_M^{(l)}$. Then $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$.

Proof. Since every rule in R' is useful, there is a derivation

$$\begin{aligned}
 s &= u[\tilde{\sigma}(s_1, \dots, s_l)] \\
 &\Rightarrow_{rfr(M)}^* u[\tilde{\sigma}(q_1(t_1), \dots, q_l(t_l))] \\
 &\Rightarrow_{rfr(M)} u[q(\tilde{\delta}(t_1, \dots, t_l))] & (\text{rule } \mu) \\
 &\Rightarrow_{rfr(M)}^* q'_0(v[\tilde{\delta}(t_1, \dots, t_l)]) \\
 &= q'_0(t)
 \end{aligned}$$

of $rfr(M)$, where $s, s_1, \dots, s_l \in T_{\Sigma_M}$, $t, t_1, \dots, t_l \in T_{\Delta_M}$, $u \in \widehat{T}_{\Sigma_M}(X_1)$ and $v \in \widehat{T}_{\Delta_M}(X_1)$. Since $rfr(M)$ is a relabeling tree transducer, it is easy to see that $\alpha = \beta$, where $\alpha = occ(u, x_1)$ and $\beta = occ(v, x_1)$.

By Definition 3.65, $fr(M)$ and $rfr(M)$ are equivalent, which implies that $(s, t) \in \tau_{fr(M)}$. Then there is a rule $\tilde{\mu} = \tilde{\sigma}(\tilde{q}_1(x_1), \dots, \tilde{q}_l(x_l)) \rightarrow \tilde{q}(\tilde{\gamma}\tilde{\delta}'(\tilde{\gamma}_1x_1, \dots, \tilde{\gamma}_lx_l))$ in R_M , where $\tilde{\delta}' \in \Delta_M^{(l)}$ and $\tilde{\gamma}, \tilde{\gamma}_1, \dots, \tilde{\gamma}_l \in (\Delta_M^{(1)})^*$, such that there is a derivation of $fr(M)$ of the following form.

$$\begin{aligned}
 s &= u[\tilde{\sigma}(s_1, \dots, s_l)] \\
 &\Rightarrow_{fr(M)}^* u[\tilde{\sigma}(\tilde{q}_1(\tilde{t}_1), \dots, \tilde{q}_l(\tilde{t}_l))] \\
 &\Rightarrow_{fr(M)} u[\tilde{q}(\tilde{\gamma}\tilde{\delta}'(\tilde{\gamma}_1\tilde{t}_1, \dots, \tilde{\gamma}_l\tilde{t}_l))] & (\text{rule } \tilde{\mu}) \\
 &\Rightarrow_{fr(M)}^* q_0(\tilde{v}[\tilde{\gamma}\tilde{\delta}'(\tilde{\gamma}_1\tilde{t}_1, \dots, \tilde{\gamma}_l\tilde{t}_l)]) \\
 &= q_0(t),
 \end{aligned}$$

where $\tilde{t}_1, \dots, \tilde{t}_l \in T_{\Delta_M}$ and $\tilde{v} \in T_{\Delta_M}(X_1)$. By Statement (v) of Observation 3.40, $\tilde{v} \in \widehat{T}_{\Delta_M}(X_1)$. Let $\beta' = occ(\tilde{v}[\tilde{\gamma}'], x_1)$. Since $fr(M)$ is a shape preserving bottom-up quasirelabeling tree transducer, it is not difficult to see that $\alpha = \beta'$. Clearly the root of the tree $stree(t, \beta)$ is $\tilde{\delta}$ while the root of the tree $stree(t, \beta')$ is $\tilde{\delta}'$. Since $\beta = \alpha = \beta'$, this implies that $\tilde{\delta}' = \tilde{\delta}$. Then it follows from Lemma 3.53 that $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$. ■

The following lemma is a consequence of Lemma 3.54. We show that, for every rule $\mu = \tilde{\sigma}(q_1(x_1)) \rightarrow q(\tilde{\delta}(x_1))$ of $rfr(M)$, where $\tilde{\sigma} \in \Sigma_M^{(1)}$ and $\tilde{\delta} \in \Delta_M^{(1)}$, the condition $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$ holds.

Lemma 3.67 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite, $fr(M) = (Q_M, \Sigma_M, \Delta_M, q_0, R_M)$ its frame transducer, and $rfr(M) = (Q', \Sigma_M, \Delta_M, q'_0, R')$ the relabeling frame transducer of M . Let $\mu =$

$\tilde{\sigma}(q_1(x_1)) \rightarrow q(\tilde{\delta}(x_1))$ be a rule in R' , where $\tilde{\sigma} \in \Sigma_M^{(1)}$ and $\tilde{\delta} \in \Delta_M^{(1)}$. Then $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$.

Proof. We prove the lemma by contradiction. Let us suppose that $h_{\Sigma_M}(\tilde{\sigma}) \not\approx h_{\Delta_M}(\tilde{\delta})$. Since every rule in R' is useful, there are trees $\tilde{s} \in T_{\Sigma_M}$ and $\tilde{t} \in T_{\Delta_M}$ such that $\tilde{s} \Rightarrow_{rfr(M)}^* q'_0(\tilde{t})$ and the rule μ is applied during this derivation. Then there is a maximal segment $\tilde{\gamma}_s \in F(\tilde{s})$, there are trees $\tilde{\gamma}_t \in (\Delta_M^{(1)})^*$, $\tilde{u}_1 \in \widehat{T}_{\Sigma_M}(X_1)$, $\tilde{u}_2 \in T_{\Sigma_M}$, $\tilde{v}_1 \in \widehat{T}_{\Delta_M}(X_1)$ and $\tilde{v}_2 \in T_{\Delta_M}$ such that $\tilde{s} = \tilde{u}_1[\tilde{\gamma}_s\tilde{u}_2]$, $\tilde{t} = \tilde{v}_1[\tilde{\gamma}_t\tilde{v}_2]$, $occ(\tilde{u}_1, x_1) = occ(\tilde{v}_1, x_1)$, $occ(\tilde{u}_1[\tilde{\gamma}_s], x_1) = occ(\tilde{v}_1[\tilde{\gamma}_t], x_1)$ and the following holds. There is a number $i \in [length(\tilde{\gamma}_s)]$ such that $h_{\Sigma_M}(\tilde{\gamma}_s(i)) \not\approx h_{\Delta_M}(\tilde{\gamma}_t(i))$. Then clearly $h_{\Sigma_M}(\tilde{\gamma}_s) \not\approx h_{\Delta_M}(\tilde{\gamma}_t)$.

On the other hand, by Definition 3.65, $fr(M)$ and $rfr(M)$ are equivalent, which implies that $(\tilde{s}, \tilde{t}) \in \tau_{fr(M)}$. Then it follows from Lemma 3.54 that $h_{\Sigma_M}(\tilde{\gamma}_s) \approx h_{\Delta_M}(\tilde{\gamma}_t)$, a contradiction, which proves the lemma. ■

We will need the last lemma of this subsection to construct a bottom-up relabeling tree transducer \bar{M} from $rfr(M)$ such that $\tau_{\bar{M}} = h_{\Sigma_M}^{-1} \circ \tau_{rfr(M)} \circ h_{\Delta_M}$. Then we will show that this \bar{M} is the relabeling which is equivalent to M .

The transducer $rfr(M)$ scans and writes out symbols which are parts of trees from the domain and the range of τ_M , respectively, as well as $fr(M)$ does. Moreover, it has the property that if $\mu = \tilde{\sigma}(q_1(x_1), \dots, q_l(x_l)) \rightarrow q(\tilde{\delta}(x_1, \dots, x_l))$ is a rule in R' , then the condition $h_{\Sigma_M}(\tilde{\sigma}) \approx h_{\Delta_M}(\tilde{\delta})$ holds. The bottom-up relabeling tree transducer \bar{M} again will scan and write out symbols from Σ and Δ , respectively, like M does. In fact the rule set of \bar{M} will contain a subset R_μ of relabeling rules such that \bar{M} can perform the derivation $h_{\Sigma_M}(\tilde{\sigma})[q_1(x_1), \dots, q_l(x_l)] \Rightarrow_{\bar{M}}^* q(h_{\Delta_M}(\tilde{\delta}))$ using rules only from R_μ . Defining for every rule $\mu \in R'$ the set R_μ , and taking the union of these sets, we will get the rule set of \bar{M} . In the next lemma we show formally how to construct \bar{M} , but first we show an example. Let us consider again our shape preserving bottom-up tree transducer M_b defined in Example 2.4, and its frame transducer $fr(M_b)$ from Example 3.39. It is easy to see that the relabeling frame transducer $rfr(M_b)$ must have a rule $\mu' = \tilde{\sigma}_1(q'_1(x_1), q'_2(x_2) \rightarrow q'(\tilde{\delta}(x_1, x_2)))$. By Example 3.39, $h_{\Sigma_{M_b}}(\tilde{\sigma}_1) = \sigma(x_1, \alpha, x_2)$ and $h_{\Delta_{M_b}}(\tilde{\delta}_1) = \delta(x_1, \beta_1, x_2)$. Then \bar{M}_b will have, among others, the set of relabeling rules $R_{\mu'} = \{\alpha \rightarrow p(\beta_1), \sigma(q'_1(x_1), p(x_2), q'_2(x_3)) \rightarrow q'(\delta(x_1, x_2, x_3))\}$, where p is a new state. Clearly \bar{M}_b can do the computation $h_{\Sigma_{M_b}}(\tilde{\sigma})[q'_1(x_1), q'_2(x_2)] \Rightarrow_{\bar{M}_b}^* q'(h_{\Delta_{M_b}}(\tilde{\delta}))$ applying rules only from the set $R_{\mu'}$.

Lemma 3.68 Let S and D be tree alphabets such that $h_S : T_S(X) \rightarrow T_\Sigma(X)$ and $h_D : T_D(X) \rightarrow T_\Delta(X)$, where Σ and Δ are ranked alphabets. Moreover, let $M = (Q, S, D, q_0, R)$ be a bottom-up relabeling tree transducer such that, for every rule $\mu = \tilde{\sigma}(q_1(x_1), \dots, q_l(x_l)) \rightarrow q(\tilde{\delta}(x_1, \dots, x_l))$ in R , the condition $h_S(\tilde{\sigma}) \approx h_D(\tilde{\delta})$ holds.

Then there is a bottom-up relabeling tree transducer $\bar{M} = (\bar{Q}, \Sigma, \Delta, q_0, \bar{R})$ such that $\tau_{\bar{M}} = h_S^{-1} \circ \tau_M \circ h_D$.

Proof. We define the bottom-up relabeling tree transducer \bar{M} in the following way. First, for every rule $\mu = \tilde{\sigma}(q_1(x_1), \dots, q_l(x_l)) \rightarrow q(\tilde{\delta}(x_1, \dots, x_l))$ in R , we define the sets Q_μ and R_μ as follows.

Since S and D are tree alphabets and $h_S(\tilde{\sigma}) \approx h_D(\tilde{\delta})$, we have that $h_S(\tilde{\sigma}) = \sigma(s_1, \dots, s_k)$ and $h_D(\tilde{\delta}) = \delta(t_1, \dots, t_k)$, for some $k \geq l$, $\sigma \in \Sigma^{(k)}$, $\delta \in \Delta^{(k)}$, $s_1, \dots, s_k \in T_\Sigma(X_l)$, $t_1, \dots, t_k \in T_\Delta(X_l)$. Moreover, $\sigma(s_1, \dots, s_k) \in \hat{T}_\Sigma(X_l)$, $\delta(t_1, \dots, t_k) \in \hat{T}_\Delta(X_l)$ and, for every $i \in [k]$, either $s_i = t_i = x_j$, for some $j \in [l]$, or $s_i \in T_\Sigma$, $t_i \in T_\Delta$ and $s_i \approx t_i$.

Let $I = \{i_1, \dots, i_l\}$, $i_1, \dots, i_l \in [k]$, $i_1 < i_2 < \dots < i_l$ such that, for every $i \in [k]$, $i \in I$ if and only if there is an index $j \in [l]$ such that $s_i = x_j$.

Now let, for every $i \in [k]$, $i \notin I$, $M_\mu^{(i)} = (Q_\mu^{(i)}, \Sigma, \Delta, q_\mu^{(i)}, R_\mu^{(i)})$ be the bottom-up relabeling tree transducer which computes the tree transformation $\{(s_i, t_i)\}$. Furthermore, let $Q_\mu = \bigcup_{i \in [k], i \notin I} Q_\mu^{(i)} \cup \{q, q_1, \dots, q_l\}$ and let

$$R_\mu = \bigcup_{i \in [k], i \notin I} R_\mu^{(i)} \cup \{\sigma(p_1(x_1), \dots, p_k(x_k)) \rightarrow q(\delta(x_1, \dots, x_k))\},$$

where, for every $i \in [k]$, $p_i = q_\mu^{(i)}$, if $i \notin I$, and $p_i = q_j$, if $i = i_j$ for some $j \in [l]$.

Finally let $\bar{Q} = \bigcup_{\mu \in R} Q_\mu$ and $\bar{R} = \bigcup_{\mu \in R} R_\mu$. It should be clear that \bar{M} is a relabeling bottom-up tree transducer and $\tau_{\bar{M}} = h_S^{-1} \circ \tau_M \circ h_D$. ■

We immediately have the following corollary.

Corollary 3.69 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer such that τ_M is infinite and let $rfr(M) = (Q', \Sigma_M, \Delta_M, q'_0, R')$ be the relabeling frame transducer of M . Then there is a bottom-up relabeling tree transducer $\bar{M} = (\bar{Q}, \Sigma, \Delta, q'_0, \bar{R})$ such that $\tau_{\bar{M}} = h_{\Sigma_M}^{-1} \circ \tau_{rfr(M)} \circ h_{\Delta_M}$.

Proof. By Lemma 3.66 and Lemma 3.67, for every rule $\mu = \sigma(q_1(x_1), \dots, q_l(x_l)) \rightarrow q(\delta(x_1, \dots, x_l))$ in R' , the condition $h_{\Sigma_M}(\sigma) \approx h_{\Delta_M}(\delta)$ holds. Then, by Lemma 3.68, there is a bottom-up relabeling tree transducer $\bar{M} = (\bar{Q}, \Sigma, \Delta, q'_0, \bar{R})$ such that $\tau_{\bar{M}} = h_{\Sigma_M}^{-1} \circ \tau_{rfr(M)} \circ h_{\Delta_M}$. ■

Now we can state the second main result of this chapter.

Theorem 3.70 Every shape preserving bottom-up tree transducer M is equivalent to a bottom-up relabeling tree transducer.

Proof. Let M be a shape preserving bottom-up tree transducer. If τ_M is finite then it is easy to see that M is equivalent to a relabeling (cf. the proof of Lemma 3.30 in case

$\gamma = \varepsilon$). Therefore let us suppose that τ_M is infinite. Since M is shape preserving, by Lemma 3.36, it is also transformable. Then let $fr(M)$ be the frame transducer of M . By Lemma 3.52, $fr(M)$ is also shape preserving. Let $rfr(M)$ be the relabeling frame transducer of M . By Corollary 3.69, there is a bottom-up relabeling tree transducer \bar{M} such that $\tau_{\bar{M}} = h_{\Sigma_M}^{-1} \circ \tau_{rfr(M)} \circ h_{\Delta_M}$. By Definition 3.65, $\tau_{rfr(M)} = \tau_{fr(M)}$, and, by Corollary 3.44, $h_{\Sigma_M}^{-1} \circ \tau_{fr(M)} \circ h_{\Delta_M} = \tau_M$. This clearly implies that $\tau_{\bar{M}} = \tau_M$, which proves the theorem. ■

Now, using Theorems 3.64 and 3.70, we can end this chapter with the following characterization of shape preserving tree transducers.

Corollary 3.71 $SHAPE = QREL$.

Proof. By Theorems 3.64 and 3.70 we have that $SHAPE \subseteq QREL$. Now the statement follows from that, clearly, $QREL \subseteq SHAPE$. ■

Chapter 4

Decidability Results

In the previous chapter we showed that shape preserving tree transducers and relabelings are equivalent. In this chapter we show more practical results. Namely, we show that it is decidable if a tree transducer is shape preserving or not and that the equivalence problem for shape preserving tree transducers is decidable.

4.1 Decidability of the Shape Preserving Property of Tree Transducers

In the following we give algorithms which decide whether a tree transducer M is shape preserving or not. Moreover we show that if M is shape preserving then a relabeling \bar{M} equivalent to M can be constructed. We will use this \bar{M} later when we show that the equivalence problem for shape preserving tree transducers is decidable.

First we deal with top-down tree transducers.

4.1.1 The Top-Down Case

We only need a few preparations to give the desired algorithm mentioned above. As we will see later, the most of the results which are necessary to give an equivalent relabeling to a shape preserving top-down tree transducer were already proved in the corresponding subsections of the previous chapter. It remains only to prove some decidability results, which we will do in the following lemmas.

First we show that it is decidable whether a top-down tree transducer is periodic or not.

Lemma 4.1 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a top-down tree transducer. Then it is decidable whether M is periodic or not.

Proof. Let $q(u) \Rightarrow_M^* v[q(x_1)]$ be a derivation of M , where $q \in Q$, $u \in \widehat{T}_\Sigma(X_1)$ such that $\text{length}(\text{occ}(u, x_1)) \leq \|Q\|$, and $v \in \widehat{T}_\Delta(X_1)$. Then two trees $u' \in \widehat{T}_\Sigma(X_1)$ and $v' \in \widehat{T}_\Delta(X_1)$ can be given such that $\text{occ}(u', x_1) = \text{occ}(u, x_1)$, $\text{occ}(v', x_1) = \text{occ}(v, x_1)$, $\text{height}(u') \leq 2\|Q\|$ and $q(u') \Rightarrow_M^* v'[q(x_1)]$. This clearly implies that it is decidable whether M is periodic. ■

Now, we prove a technical lemma which uses standard pumping arguments to show that it is decidable whether a recognizable tree language $L \subseteq T_{\langle \Sigma_\diamond, \Delta_\diamond \rangle}$, where Σ and Δ are ranked alphabets, is bounded or not.

Lemma 4.2 Let Σ and Δ be ranked alphabets and $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, p_0, R_T)$ be a top-down tree automaton. Then it is decidable if $L(T)$ is bounded and, moreover, if $L(T)$ is bounded, then we can compute the smallest k for which $L(T)$ is k -bounded.

Proof. Let $n = \|P\|$ and define for every $p, q \in P$

$$L_{p,q}^{(n)} = \{\gamma \in (\langle \Sigma_\diamond, \Delta_\diamond \rangle^{(1)})^* \mid p(\gamma) \Rightarrow_T^* \gamma q(x_1) \text{ and } \text{length}(\gamma) \leq n\}.$$

It should be clear that $L_{p,q}^{(n)}$ is a finite and computable set.

a) First we show that it is decidable whether $L(T)$ is bounded. To see this we prove that $L(T)$ is bounded if and only if, for every $p \in P$ and $\gamma \in L_{p,p}^{(n)}$ the condition $|\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| = 0$ holds.

Assume that $L(T)$ is bounded and there are $p \in P$ and $\gamma \in L_{p,p}^{(n)}$ such that $|\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| > 0$. Then, since all states in p are useful, there are $s \in \widehat{T}_{\langle \Sigma_\diamond, \Delta_\diamond \rangle}(X_1)$ and $u \in T_{\langle \Sigma_\diamond, \Delta_\diamond \rangle}$ such that $s[\gamma u] \in L(T)$, hence $\gamma \in F(L(T))$. Since $p(\gamma) \Rightarrow_T^* \gamma p(x_1)$, for every $l > 0$, $s[\gamma^l u] \in L(T)$ and thus $\gamma^l \in F(L(T))$. This means that $L(T)$ is not bounded, which is a contradiction.

Next assume that $L(T)$ is not bounded. Let

$$L_n = \{\gamma \in F(L(T)) \mid |\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| > n\}$$

and let γ be an element of L_n with minimal length. Then certainly $\text{length}(\gamma) > n$ and thus there are $\gamma_1, \gamma_2, \gamma_3 \in (\langle \Sigma_\diamond, \Delta_\diamond \rangle^{(1)})^*$ such that $0 < \text{length}(\gamma_2) \leq n$ and $\gamma = \gamma_1 \gamma_2 \gamma_3$, moreover, there are states $p, q \in P$ such that $q(\gamma_1 \gamma_2 \gamma_3) \Rightarrow_T^* \gamma_1 p(\gamma_2 \gamma_3) \Rightarrow_T^* \gamma_1 \gamma_2 p(\gamma_3)$. Then $\gamma_2 \in L_{p,p}^{(n)}$ and $\gamma_1 \gamma_3 \in F(L(T))$. Now it can be seen that $|\text{length}(h_\Sigma(\gamma_2)) - \text{length}(h_\Delta(\gamma_2))| > 0$. Indeed, if $|\text{length}(h_\Sigma(\gamma_2)) - \text{length}(h_\Delta(\gamma_2))| = 0$, then, clearly, $|\text{length}(h_\Sigma(\gamma_1 \gamma_3)) - \text{length}(h_\Delta(\gamma_1 \gamma_3))| > n$ and thus $\gamma_1 \gamma_3 \in L_n$, which is a contradiction because $\text{length}(\gamma_1 \gamma_3) < \text{length}(\gamma)$.

Now since it is decidable if, for every state $p \in P$ and $\gamma \in L_{p,p}^{(n)}$ the condition $|\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| = 0$ holds, it is also decidable if $L(T)$ is bounded.

b) Now assume that $L(T)$ is bounded. Let

$$k = \max\{|\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| \mid \gamma \in L_{p,q}^{(n)}, p, q \in Q\}.$$

Certainly there is no $k' < k$ such that $L(T)$ is k' -bounded. On the other hand, we can show that $L(T)$ is k -bounded.

Let us suppose that $L(T)$ is not k -bounded and let

$$L_k = \{\gamma \in F(L(T)) \mid |\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| > k\}$$

and γ be an element of L_k with minimal length. Now, by the definition of k , $\text{length}(\gamma) > n$ and thus there are $\gamma_1, \gamma_2, \gamma_3 \in (\langle \Sigma_\diamond, \Delta_\diamond \rangle^{(1)})^*$ and states $p, q \in P$ with the same properties as in Case a). Then $|\text{length}(h_\Sigma(\gamma_2)) - \text{length}(h_\Delta(\gamma_2))| = 0$, because $L(T)$ is bounded, see Case a). Moreover, $|\text{length}(h_\Sigma(\gamma_1\gamma_3)) - \text{length}(h_\Delta(\gamma_1\gamma_3))| > k$, hence $\gamma_1\gamma_3 \in L_k$. This is a contradiction because $\text{length}(\gamma_1\gamma_3) < \text{length}(\gamma)$ and γ is an element of L_k with minimal length. Hence $L(T)$ is k -bounded. ■

Now let us consider a top-down tree automaton $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, q_0, R_T)$ such that $L(T)$ is k -bounded, and its shape preserving frame $T' = (P', \langle \Sigma, \Delta \rangle, p'_0, R_{T'})$. By Lemma 3.62, if $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ is shape preserving then $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ holds. In the next lemma we show that it is decidable if the above inclusion holds.

Lemma 4.3 Let $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, q_0, R_T)$ be a top-down tree automaton such that $L(T)$ is k -bounded and $T' = (P', \langle \Sigma, \Delta \rangle, p'_0, R_{T'})$ the shape preserving frame of T . Then it is decidable, if $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ holds.

Proof. We show that $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ if and only if $\text{ran}(\tau_T) \subseteq \text{dom}(\tau_N)$.

Assume that $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ and let $v \in \text{ran}(\tau_T)$. Let $s = h_\Sigma(v)$ and $t = h_\Delta(v)$, then $(s, t) \in h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ and thus $(s, t) \in h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$. Then, there is a $v' \in \text{ran}(\tau_T \circ \tau_N)$ such that $s = h_\Sigma(v')$ and $t = h_\Delta(v')$. Since $v' \in T_{\langle \Sigma, \Delta \rangle}$, by Observation 3.56, $s \approx t$ holds, which implies that for every maximal $\gamma \in F(v)$, $|\text{length}(h_\Sigma(\gamma)) - \text{length}(h_\Delta(\gamma))| = 0$ holds. Consequently, $v \in \text{dom}(\tau_N)$.

Now assume that $\text{ran}(\tau_T) \subseteq \text{dom}(\tau_N)$ and let $(s, t) \in h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$. Then, there is a $v \in \text{ran}(\tau_T)$ such that $s = h_\Sigma(v)$ and $t = h_\Delta(v)$. Now $v \in \text{dom}(\tau_N)$ also holds, which implies that there is a $v' \in \text{ran}(\tau_N)$ such that $\tau_N(v) = v'$. It follows from the construction of N that $s = h_\Sigma(v')$ and $t = h_\Delta(v')$. Then $(s, t) \in h_\Sigma^{-1} \circ (\tau_T \circ \tau_N) \circ h_\Delta$, cf. the proof of Lemma 3.61. Consequently, $(s, t) \in h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$.

Then the decidability of $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ follows from the fact that both $\text{ran}(\tau_T)$ and $\text{dom}(\tau_N)$ are recognizable tree languages and that the inclusion problem is decidable for recognizable tree languages (cf. Theorem 10.3 in Chapter II. of [GS84]).

■

Now we can state our first decidability result.

Theorem 4.4 *It is decidable if a top-down tree transducer $M = (Q, \Sigma, \Delta, q_0, R)$ is shape preserving or not. Moreover, if M is shape preserving, then a relabeling \bar{M} can be constructed such that $\tau_M = \tau_{\bar{M}}$.*

Proof. We give an algorithm that terminates with yes if M is shape preserving, otherwise it terminates with no. Furthermore, if M is shape preserving, then the algorithm outputs a top-down relabeling tree transducer \bar{M} which is equivalent to M . The algorithm is as follows.

1. Check if M is permutation top-down quasirelabeling or not. If not, then halt with no because, by Lemma 3.5, M is not shape preserving.
2. Check whether M is periodic. (By Lemma 4.1, it is decidable if M is periodic.) If M is not periodic, then halt with no because, by Lemma 3.28, M is not shape preserving.
3. Compute the relation \prec on Q (Definition 3.24) according to Lemma 3.27. If there are states q_1, \dots, q_k for some $1 < k \leq \|Q\|$ such that $q_1 = q_k$ and $q_1 \prec q_2 \prec \dots \prec q_k$, then halt with no, because, by Lemma 3.29, M is not shape preserving.
4. Eliminate the permutation rules from M in the following way (cf. Lemma 3.31). Take a permutation rule μ with a maximal state in its left-hand side with respect to \prec . Check, if the condition $(*)$ described in the proof of Lemma 3.31 holds. (Note that the tree language $\text{dom}(\tau_{M, q_{\pi^{-1}(i_j)}})$ is recognizable by Theorem II.1 of [Rou70]. Moreover, since M is linear, $\gamma_{i_j} \text{ran}(\tau_{M, q_{i_j}})$ is also recognizable [Rou70] (cf. also Corollary IV.6.6 in [GS84]). Therefore these languages are effectively computable.) If the condition $(*)$ does not hold, then halt with no, because M is not shape preserving. Otherwise eliminate the rule μ . (Finally M becomes a quasirelabeling.)
5. Construct the top-down tree automaton $T = (P, \langle \Sigma_\diamond, \Delta_\diamond \rangle, p_0, R_T)$ such that $\tau_M = h_\Sigma^{-1} \circ \tau_T \circ h_\Delta$ (Lemma 3.57).
6. Check if $L(T)$ is bounded (Lemma 4.2). If not, then halt with no, because, by Lemma 3.59, M is not shape preserving. If yes, then compute k such that $L(T)$ is k -bounded (Lemma 4.2).
7. Construct the shape preserving frame $T' = (P', \langle \Sigma, \Delta \rangle, p'_0, R_{T'})$ of T (Definition 3.60).
8. Check, if $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta \subseteq h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ (Lemma 4.3). If not, then halt with no, because, by Lemma 3.62, M is not shape preserving.
9. Let \bar{M} be the relabeling equivalent to $h_\Sigma^{-1} \circ \tau_{T'} \circ h_\Delta$ (by Lemma 3.63 \bar{M} can be constructed). Then halt with yes (by Lemmas 3.61 and 3.62, $h_\Sigma^{-1} \circ \tau_T \circ h_\Delta = \tau_{\bar{M}}$).

4.1.2 The Bottom-Up Case

In this subsection we show that the shape preserving property of bottom-up tree transducers is also decidable. Moreover, similarly as in the previous subsection, we show that an equivalent relabeling can be effectively constructed to every shape preserving transformable bottom-up tree transducer.

Using the results obtained in the previous chapter we can easily prove the next lemma.

Lemma 4.5 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a transformable bottom-up tree transducer. Then it is decidable whether M is shape preserving or not. Moreover, if M is shape preserving, then a bottom-up relabeling tree transducer \bar{M} can be constructed such that $\tau_M = \tau_{\bar{M}}$.

Proof. We give an algorithm that terminates with yes if M is shape preserving, otherwise it terminates with no. Moreover, if M is shape preserving, then the algorithm outputs a relabeling \bar{M} which is equivalent to M . The algorithm is as follows.

1. Construct the frame transducer $fr(M)$ of M . (Note that for every rule $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ in R such that $\tau_{M,q}$ is infinite and $\|inf(\mu)\| = 0$, the set $mp(\mu)$ can be easily determined, if we consider a derivation of M in which μ is applied.)
2. Construct a top-down tree transducer M_T such that $\tau_{fr(M)} = \tau_{M_T}$. (By Item (i) of Observation 3.40, $fr(M)$ is a quasirelabeling. Then it is effectively equivalent to a top-down quasirelabeling, cf. the discussion after the definition of quasirelabelings in Chapter 2.)
3. Check whether M_T is shape preserving. (By Theorem 4.4, it is decidable if M_T is shape preserving or not). If M_T is not shape preserving, then halt with no, because then $fr(M)$ is not shape preserving and, by Lemma 3.52, it follows that M is also not shape preserving. Note that since M is transformable, τ_M is infinite.)
4. Construct the top-down relabeling tree transducer \bar{M}_T such that $\tau_{M_T} = \tau_{\bar{M}_T}$. (By Theorem 4.4, \bar{M}_T can be constructed.)
5. Construct the relabeling frame transducer $rfr(M) = (Q', \Sigma_M, \Delta_M, q'_0, R')$ of M ($rfr(M)$ can be effectively constructed from \bar{M}_T , cf. again the discussion after the definition of quasirelabelings in Chapter 2).
6. For every rule $\sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(\delta(x_1, \dots, x_k))$ in R' check whether $h_{\Sigma_M}(\sigma) \approx h_{\Delta_M}(\delta)$. If not, then halt with no because, by Lemma 3.66 or by Lemma 3.67, M is not shape preserving.

7. Construct the bottom-up relabeling tree transducer \bar{M} , such that $h_{\Sigma_M}^{-1} \circ \tau_{r_{fr(M)}} \circ h_{\Delta_M} = \tau_{\bar{M}}$. (By Lemma 3.68, \bar{M} can be constructed.) Then halt with yes. (Now M is equivalent to \bar{M} , because, by Definition 3.65, $\tau_{r_{fr(M)}} = \tau_{fr(M)}$ and, by Corollary 3.44, $h_{\Sigma_M}^{-1} \circ \tau_{fr(M)} \circ h_{\Delta_M} = \tau_M$, which implies that $\tau_{\bar{M}} = \tau_M$. Hence M is shape preserving.)

■

It should be clear that if it is decidable whether a bottom-up tree transducer M is transformable, then the shape preserving property of M is also decidable. So, it remains to show that it is decidable if M is transformable or not. However, to do so we need a rather big preparation.

First we define two special bottom-up tree transducers, namely periodic and occurrence preserving bottom-up tree transducers. We show that the occurrence preserving property is decidable for periodic tree transducers. These properties, briefly, express the following. Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer, $q \in Q$, $u \in \hat{T}_{\Sigma}(X_1)$ and $v \in \hat{T}_{\Delta}(X_1)$. If M is periodic, $length(occ(u, x_1)) \leq ||Q||$ and there is a derivation $u[q(x_1)] \Rightarrow_M^* q(v)$, then $length(occ(u, x_1)) = length(occ(v, x_1))$. If M is occurrence preserving, $\tau_{M,q}$ is infinite and there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$, then $occ(u, x_1)$ and $occ(v, x_1)$ are comparable. It can be seen that in both cases M can have rather restricted trees in its certain derivations. We show that if M is periodic, then it is enough to examine a finite number of derivations of M to decide whether it is occurrence preserving. After this we show that if M is periodic and occurrence preserving, then it is decidable whether M is transformable or not. On the other hand, we will show that if M is shape preserving, then it is periodic and occurrence preserving as well.

Summarizing the above discussion, to decide the shape preserving property of M , we have to do the following. First we decide whether M is periodic and occurrence preserving. If M is not periodic or not occurrence preserving, then M can not be shape preserving. Otherwise we check whether M is transformable. If M is not transformable, then it is again not shape preserving. However, if it is transformable, then we can decide whether it is shape preserving or not (cf. Lemma 4.5).

First we make some observations, which will be helpful in the rest of the Thesis. The first observation expresses an easy fact concerning words.

Observation 4.6 Let $\alpha, \beta \in \mathbb{N}^*$ such that α and β are incomparable. Let α_1 and β_1 be common prefixes of α and β . Let $\alpha'_1, \beta'_1 \in \mathbb{N}^*$ such that $length(\alpha_1) - length(\alpha'_1) = length(\beta_1) - length(\beta'_1)$. Moreover, let $\alpha = \alpha_1\alpha_2$ and $\beta = \beta_1\beta_2$, for some $\alpha_2, \beta_2 \in \mathbb{N}^*$. Then the words $\alpha'_1\alpha_2$ and $\beta'_1\beta_2$ are also incomparable.

Proof. If α'_1 and β'_1 are incomparable, then clearly $\alpha'_1\alpha_2$ and $\beta'_1\beta_2$ are also incomparable, so let us suppose that α'_1 and β'_1 are comparable. Let n be the smallest number in \mathbb{N} such that $\alpha(n) \neq \beta(n)$ and let $m = \text{length}(\alpha_1) - \text{length}(\alpha'_1)$. Clearly $m \leq \text{length}(\alpha_1)$ and, since $n > \text{length}(\alpha_1)$, we have that $m < n$. Moreover $\alpha'_1\alpha_2(n-m) \neq \beta'_1\beta_2(n-m)$ which clearly implies that $\alpha'_1\alpha_2$ and $\beta'_1\beta_2$ are incomparable. ■

The next observation will be useful when we consider certain derivations of a bottom-up tree transducer.

Observation 4.7 Let $u \in \widehat{T}_\Sigma(X_1)$ and $u_1, u_2 \in T_\Sigma(X_1)$ such that $u = u_1[u_2]$. Then $u_1, u_2 \in \widehat{T}_\Sigma(X_1)$.

Proof. We prove the statement by contradiction. Let us suppose that $u_1 \notin \widehat{T}_\Sigma(X_1)$ or $u_2 \notin \widehat{T}_\Sigma(X_1)$. Then there are two possibilities. Either (i) $u_1 \in T_\Sigma$ or $u_2 \in T_\Sigma$, or (ii) u_1 and u_2 contain x_1 at least once and one of them contains x_1 more than once. In Case (i) $u \in T_\Sigma$, while in Case (ii) u contains x_1 more than once. In both cases we have a contradiction, which proves the statement. ■

Throughout the rest of the Thesis, whenever we consider a tree alphabet Σ and trees $u \in \widehat{T}_\Sigma(X_1)$ and $u_1, u_2 \in T_\Sigma(X_1)$ such that $u = u_1[u_2]$, then, by Observation 4.7, we will assume that $u_1, u_2 \in \widehat{T}_\Sigma(X_1)$.

Now we define periodic and occurrence preserving bottom-up tree transducers.

Definition 4.8 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer. We say that M is

- (a) *periodic* if, for every derivation $u[q(x_1)] \Rightarrow_M^* q(v)$, where $q \in Q$, $u \in \widehat{T}_\Sigma(X_1)$ such that $\text{length}(\text{occ}(u, x_1)) \leq \|Q\|$ and $v \in \widehat{T}_\Delta(X_1)$, we have that $\text{length}(\text{occ}(u, x_1)) = \text{length}(\text{occ}(v, x_1))$;
- (b) *occurrence preserving* if it satisfies the following condition. If $u[q(x_1)] \Rightarrow_M^* q_0(v)$ is a derivation of M , where $q \in Q$ such that $\tau_{M,q}$ is infinite, $u \in \widehat{T}_\Sigma(X_1)$ and $v \in \widehat{T}_\Delta(X_1)$, then $\text{occ}(u, x_1)$ and $\text{occ}(v, x_1)$ are comparable.

The following statements are similar to the corresponding ones concerning top-down tree transducers (cf. Lemmas 4.1 and 3.28). They can be proved analogously to the top-down case therefore their proof are omitted.

Lemma 4.9 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer. Then it is decidable whether M is periodic or not. ■

Lemma 4.10 If $M = (Q, \Sigma, \Delta, q_0, R)$ is a shape preserving bottom-up tree transducer, then it is periodic as well. ■

Next we prove two technical lemmas concerning periodic bottom-up tree transducers. We will use these lemmas when we show that it is decidable whether a periodic bottom-up tree transducer is occurrence preserving or not. The proof of the first one relies on the simple idea that a state of a bottom-up tree transducer necessarily repeats during a derivation of an input tree which has a sufficiently large height.

Lemma 4.11 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a periodic bottom-up tree transducer. Moreover, let $u \in \widehat{T}_\Sigma(X_1)$ and $v \in \widehat{T}_\Delta(X_1)$ such that $\text{length}(\text{occ}(u, x_1)) > \|Q\|$ and there is a derivation $u[q(x_1)] \Rightarrow_M^* p(v)$, where $p, q \in Q$. Then there are trees $u' \in \widehat{T}_\Sigma(X_1)$ and $v' \in \widehat{T}_\Delta(X_1)$ such that $\text{length}(\text{occ}(u', x_1)) \leq \|Q\|$, $u'[q(x_1)] \Rightarrow_M^* p(v')$ and $\text{length}(\text{occ}(u, x_1)) - \text{length}(\text{occ}(u', x_1)) = \text{length}(\text{occ}(v, x_1)) - \text{length}(\text{occ}(v', x_1))$.

Proof. It is enough to show that we can give two trees $\bar{u} \in \widehat{T}_\Sigma(X_1)$ and $\bar{v} \in \widehat{T}_\Delta(X_1)$ such that $\text{length}(\text{occ}(\bar{u}, x_1)) < \text{length}(\text{occ}(u, x_1))$, $\bar{u}[q(x_1)] \Rightarrow_M^* p(\bar{v})$ and $\text{length}(\text{occ}(u, x_1)) - \text{length}(\text{occ}(\bar{u}, x_1)) = \text{length}(\text{occ}(v, x_1)) - \text{length}(\text{occ}(\bar{v}, x_1))$. Since $\text{length}(\text{occ}(u, x_1)) > \|Q\|$, there are trees $u_1, u_2, u_3 \in \widehat{T}_\Sigma(X_1)$, $v_1, v_2, v_3 \in \widehat{T}_\Delta(X_1)$ and a state $q' \in Q$ such that $u = u_1[u_2[u_3]]$, $1 \leq \text{length}(\text{occ}(u_2, x_1)) \leq \|Q\|$, $v = v_1[v_2[v_3]]$ and $u[q(x_1)] = u_1[u_2[u_3[q(x_1)]]] \Rightarrow_M^* u_1[u_2[q'(v_3)]] \Rightarrow_M^* u_1[q'(v_2[v_3])] \Rightarrow_M^* p(v_1[v_2[v_3]]) = p(v)$. Let $\bar{u} = u_1[u_3]$ and $\bar{v} = v_1[v_3]$. Since $u_2 \neq x_1$, $\text{length}(\text{occ}(\bar{u}, x_1)) < \text{length}(\text{occ}(u, x_1))$. Moreover, it is easy to see that there is a derivation $\bar{u}[q(x_1)] \Rightarrow_M^* p(\bar{v})$ of M . Furthermore, since M is periodic, $\text{length}(\text{occ}(u_2, x_1)) = \text{length}(\text{occ}(v_2, x_1))$. Then we have that $\text{length}(\text{occ}(u, x_1)) - \text{length}(\text{occ}(\bar{u}, x_1)) = \text{length}(\text{occ}(v, x_1)) - \text{length}(\text{occ}(\bar{v}, x_1))$ which proves the lemma. ■

The following lemma is an easy consequence of the previous one.

Lemma 4.12 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a periodic bottom-up tree transducer. Let $u \in \widehat{T}_\Sigma(X_1)$ and $v \in \widehat{T}_\Delta(X_1)$ such that there is a derivation $u[q(x_1)] \Rightarrow_M^* p(v)$, where $p, q \in Q$. Let $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$. Then $\text{length}(\alpha) - \text{length}(\beta) \leq \|Q\|$.

Proof. We prove the lemma by contradiction. Let us suppose that $\text{length}(\alpha) - \text{length}(\beta) > \|Q\|$. Then clearly $\text{length}(\alpha) > \|Q\|$ and by Lemma 4.11 there are trees $u' \in \widehat{T}_\Sigma(X_1)$ and $v' \in \widehat{T}_\Delta(X_1)$ such that $\text{length}(\text{occ}(u', x_1)) \leq \|Q\|$, $u'[q(x_1)] \Rightarrow_M^* p(v')$ and $\text{length}(\alpha) - \text{length}(\text{occ}(u', x_1)) = \text{length}(\beta) - \text{length}(\text{occ}(v', x_1))$. On the other hand, since $\text{length}(\alpha) - \|Q\| > \text{length}(\beta)$ and $\text{length}(\text{occ}(u', x_1)) \leq \|Q\|$, we have that $\text{length}(\alpha) - \text{length}(\text{occ}(u', x_1)) > \text{length}(\beta)$. Clearly this is a contradiction which proves the lemma. ■

Now, we show that it is decidable if a periodic bottom-up tree transducer is occurrence preserving or not.

Lemma 4.13 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a periodic bottom-up tree transducer. Then it is decidable whether M is occurrence preserving or not.

Proof. Let us assume that M is not occurrence preserving. Then by Item (b) of Definition 4.8 there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$ of M , where $u \in \widehat{T}_\Sigma(X_1)$, $v \in \widehat{T}_\Delta(X_1)$ and $q \in Q$, such that $\tau_{M,q}$ is infinite and the occurrences $\text{occ}(u, x_1)$ and $\text{occ}(v, x_1)$ are incomparable. Let $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$. Without loss of generality, we can assume that $\text{height}(u) \leq \text{length}(\alpha) + \|Q\|$. Indeed, if $\text{height}(u) > \text{length}(\alpha) + \|Q\|$, then we can easily give trees $u' \in \widehat{T}_\Sigma(X_1)$ and $v' \in \widehat{T}_\Delta(X_1)$ such that $\text{occ}(u', x_1) = \alpha$, $\text{occ}(v', x_1) = \beta$, $u'[q(x_1)] \Rightarrow_M^* q_0(v')$ and $\text{height}(u') \leq \text{length}(\alpha) + \|Q\|$.

Moreover, we can assume that α is with the minimal length, i.e. there are no trees $u' \in \widehat{T}_\Sigma(X_1)$, $v' \in \widehat{T}_\Delta(X_1)$ and state $q' \in Q$ such that $\tau_{M,q'}$ is infinite, $u'[q'(x_1)] \Rightarrow_M^* q_0(v')$, $\text{occ}(u', x_1)$ and $\text{occ}(v', x_1)$ are incomparable and $\text{length}(\text{occ}(u', x_1)) < \text{length}(\alpha)$.

We show that $\text{length}(\alpha)$ is bounded by a number which only depends on M . This clearly implies that it is decidable if M is occurrence preserving or not, since it is enough to examine a finite number of its derivations.

Let us consider the derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$ in more detail. Let $u = u_1[\sigma(s_1, \dots, s_{i-1}, u_2, s_{i+1}, \dots, s_k)]$, for some $u_1, u_2 \in \widehat{T}_\Sigma(X_1)$, $k \in \mathbb{N}$, $i \in [k]$, $\sigma \in \Sigma^{(k)}$ and $s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_k \in T_\Sigma$ such that the following holds. There is a rule $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow p(r)$ in R such that the above derivation can be written in the following form.

$$\begin{aligned}
 u[q(x_1)] &= u_1[\sigma(s_1, \dots, u_2[q(x_1)], \dots, s_k)] \\
 &\Rightarrow_M^* u_1[\sigma(q_1(t_1), \dots, q_i(v_2), \dots, q_k(t_k))] \\
 &\Rightarrow_M u_1[p(r[t_1, \dots, v_2, \dots, t_k])] && \text{(rule } \mu) \\
 &\Rightarrow_M^* q_0(v_1[r[t_1, \dots, v_2, \dots, t_k]]) \\
 &= q_0(v),
 \end{aligned}$$

where $v_1, v_2 \in \widehat{T}_\Delta(X_1)$ (cf. Observation 4.7) and $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_k \in T_\Delta$. Moreover, there is an occurrence $\omega \in \text{occ}(r)$ such that $\beta_1\omega$, where $\beta_1 = \text{occ}(v_1, x_1)$, is the longest common prefix of α and β . Let $\bar{\omega} = \beta_1\omega$, $\alpha_1 = \text{occ}(u_1, x_1)$, $\alpha_2 = \text{occ}(u_2, x_1)$ and $\beta_2 = \text{occ}(v_2, x_1)$ (cf. Figure 4.1). It is easy to see that r contains x_i exactly once, otherwise $r[t_1, \dots, t_{i-1}, x_1, t_{i+1}, \dots, t_k] \notin \widehat{T}_\Delta(X_1)$, which contradicts Observation 4.7. Now we distinguish the following two cases.

Case 1: $\bar{\omega}$ is a prefix of α_1 (cf. Figure 4.1). First we show that in this case $u_2 = x_1$. On the contrary, let us suppose that $u_2 \neq x_1$. Let us consider the tree $u^{(1)} = u_1[\sigma(s_1, \dots, s_{i-1}, x_1, s_{i+1}, \dots, s_k)]$. Clearly $\text{length}(\text{occ}(u^{(1)}, x_1)) < \text{length}(\alpha)$, and there is a derivation $u^{(1)}[q_i(x_1)] \Rightarrow_M^* q_0(v^{(1)})$ of M , where the tree $v^{(1)} = v_1[r[t_1, \dots, t_{i-1}, x_1, t_{i+1}, \dots, t_k]]$. Since $\tau_{M,q}$ is infinite, τ_{M,q_i} is also infinite. Moreover it is easy to see that $\text{occ}(u^{(1)}, x_1)$ and $\text{occ}(v^{(1)}, x_1)$ are incomparable. This contradicts

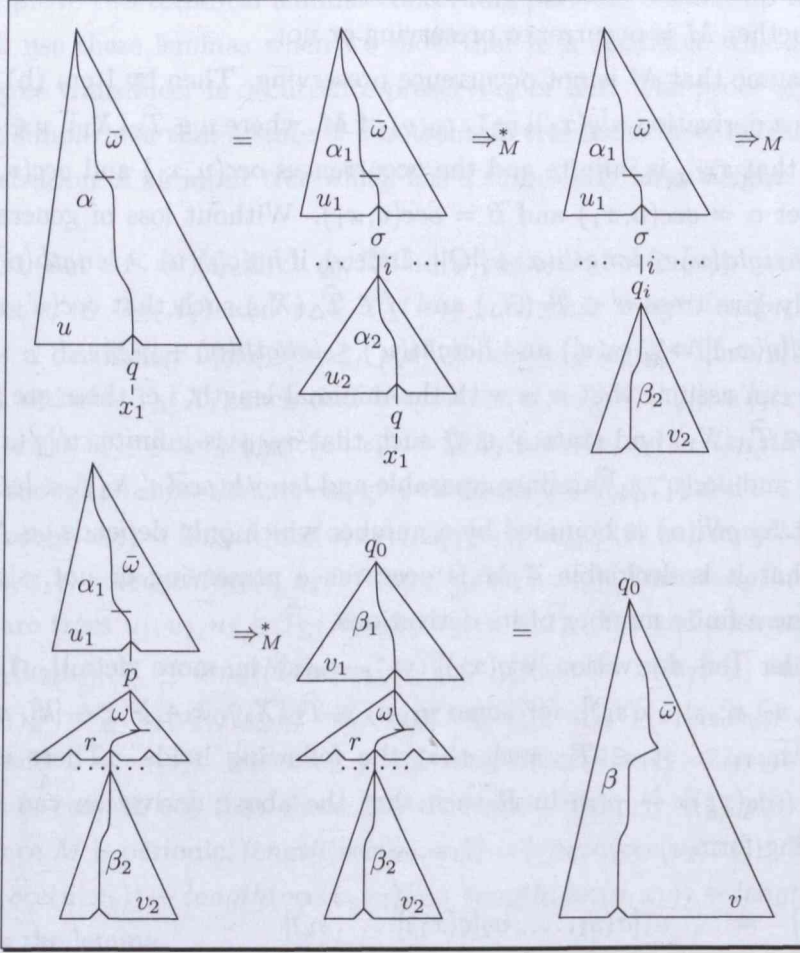


Figure 4.1: $\bar{\omega} = \beta_1 \omega$ is the longest common prefix of α and β in Lemma 4.13.

the fact that $\text{length}(\alpha)$ is minimal, which implies that $u_2 = x_1$.

Since $\bar{\omega}$ is a prefix of α_1 , $u_1 = u_{11}[u_{12}]$, for some $u_{11}, u_{12} \in \hat{T}_\Sigma(X_1)$, such that $\text{occ}(u_{11}, x_1) = \bar{\omega}$. Clearly, $u_{11}[u_{12}[p(x_1)]] \Rightarrow_M^* u_{11}[q'(v_{12})] \Rightarrow_M^* q_0(v_{11}[v_{12}])$, for some $q' \in Q$ and $v_{11}, v_{12} \in \hat{T}_\Delta(X_1)$, such that $v_1 = v_{11}[v_{12}]$. Let $\alpha_{11} = \text{occ}(u_{11}, x_1)$, $\alpha_{12} = \text{occ}(u_{12}, x_1)$, $\beta_{11} = \text{occ}(v_{11}, x_1)$ and $\beta_{12} = \text{occ}(v_{12}, x_1)$ (cf. Figure 4.2).

Now we show that $\text{length}(\alpha_{11}) \leq \|Q\|$. To derive a contradiction, assume that $\text{length}(\alpha_{11}) > \|Q\|$. Then, by Lemma 4.11, there are trees, $u'_{11} \in \hat{T}_\Sigma(X_1)$ and $v'_{11} \in \hat{T}_\Delta(X_1)$ such that the following holds. $\text{length}(\text{occ}(u'_{11}, x_1)) \leq \|Q\|$, $\text{length}(\alpha_{11}) - \text{length}(\text{occ}(u'_{11}, x_1)) = \text{length}(\beta_{11}) - \text{length}(\text{occ}(v'_{11}, x_1))$ and $u'_{11}[q'(x_1)] \Rightarrow_M^* q_0(v'_{11})$. Let $u^{(2)} = u'_{11}[u_{12}[\sigma(s_1, \dots, s_{i-1}, x_1, s_{i+1}, \dots, s_k)]]$. Clearly $\text{length}(\text{occ}(u^{(2)}, x_1)) < \text{length}(\alpha)$ and there is a derivation $u^{(2)}[q_i(x_1)] \Rightarrow_M^* q_0(v^{(2)})$ of M , where $v^{(2)} = v'_{11}[v_{12}[r[t_1, \dots, t_{i-1}, x_1, t_{i+1}, \dots, t_k]]]$. It can be seen also, with the help of Observation 4.6, that $\text{occ}(u^{(2)}, x_1)$ and $\text{occ}(v^{(2)}, x_1)$ are incomparable, which again contradicts that

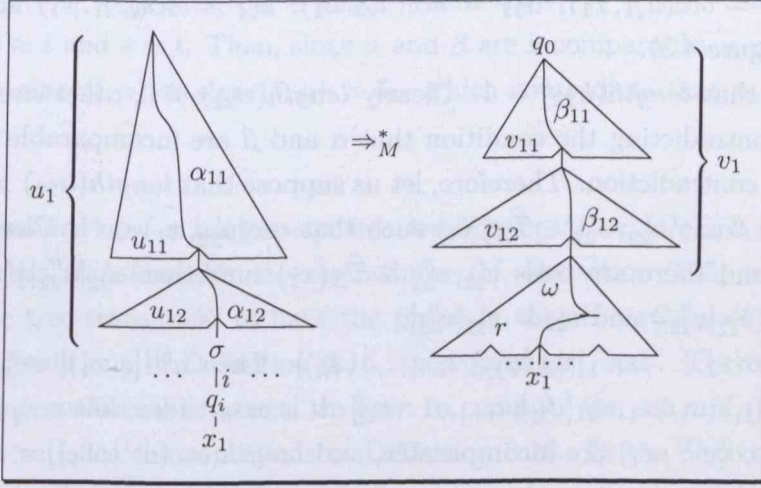


Figure 4.2: Case 1 of Lemma 4.13: $\bar{\omega}(= \beta_{11}\beta_{12}\omega = \alpha_{11})$ is a prefix of $\alpha_1(= \alpha_{11}\alpha_{12})$.

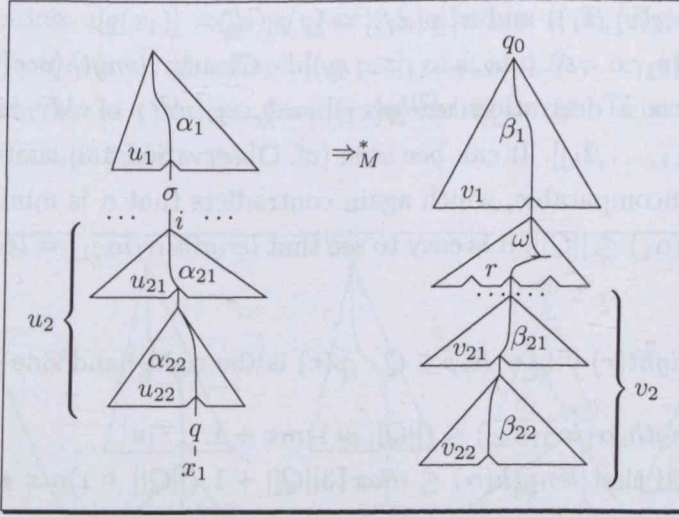


Figure 4.3: Case 2 of Lemma 4.13: $\alpha_1 i$ is a prefix of $\bar{\omega}(= \beta_1 \omega = \alpha_1 i \alpha_{21})$.

α is minimal, thus $\text{length}(\alpha_{11}) \leq \|Q\|$.

Since $\text{length}(\alpha_{11}) \leq \|Q\|$ and $\alpha_{11} = \bar{\omega}$, it follows that $\text{length}(\bar{\omega}) \leq \|Q\|$. Then clearly $\text{length}(\beta_{12}) \leq \|Q\|$, which by Lemma 4.12 implies that $\text{length}(\alpha_{12}) \leq 2\|Q\|$.

Summing up, we have that $\text{length}(\alpha) = \text{length}(\alpha_1) + 1 + \text{length}(\alpha_2) = \text{length}(\alpha_{11}) + \text{length}(\alpha_{12}) + 1 \leq 3\|Q\| + 1$.

Case 2: $\alpha_1 i$ is a prefix of $\bar{\omega}$. Then $u_2 = u_{21}[u_{22}]$, for some $u_{21}, u_{22} \in \hat{T}_\Sigma(X_1)$, such that $\text{occ}(u_1[\sigma(s_1, \dots, s_{i-1}, u_{21}, s_{i+1}, \dots, s_k)], x_1) = \bar{\omega}$. Clearly, $u_{21}[u_{22}[q(x_1)]] \Rightarrow_M^* u_{21}[q'(v_{22})] \Rightarrow_M^* q_i(v_{21}[v_{22}])$, for some $q' \in Q$ and $v_{21}, v_{22} \in \hat{T}_\Delta(X_1)$ such that $v_2 =$

$v_{21}[v_{22}]$. Let $\alpha_{21} = \text{occ}(u_{21}, x_1)$, $\alpha_{22} = \text{occ}(u_{22}, x_1)$, $\beta_{21} = \text{occ}(v_{21}, x_1)$ and $\beta_{22} = \text{occ}(v_{22}, x_1)$ (cf. Figure 4.3).

First we show that $\text{length}(\alpha_{22}) = 1$. Clearly $\text{length}(\alpha_{22}) \neq 0$, otherwise α would be a prefix of β contradicting the condition that α and β are incomparable. Now we show the claim by contradiction. Therefore, let us suppose that $\text{length}(\alpha_{22}) > 1$. Then $u_{22} = u'_{22}[u''_{22}]$, for some $u'_{22}, u''_{22} \in \widehat{T}_\Sigma(X_1)$ such that $\text{occ}(u'_{22}, x_1) \in \mathbb{N}$. Clearly, there is a state $\bar{q} \in Q$ and there are trees $v'_{22}, v''_{22} \in \widehat{T}_\Delta(x_1)$ such that $u'_{22}[u''_{22}[q(x_1)]] \Rightarrow_M^* u'_{22}[\bar{q}(v''_{22})] \Rightarrow_M^* q'(v'_{22}[v''_{22}])$ and $v_{22} = v'_{22}[v''_{22}]$.

Let $u^{(1)} = u_1[\sigma(s_1, \dots, s_{i-1}, u_{21}[u'_{22}], s_{i+1}, \dots, s_k)]$. Then $u^{(1)}[\bar{q}(x_1)] \Rightarrow_M^* q_0(v^{(1)})$, where $v^{(1)} = v_1[r[t_1, \dots, t_{i-1}, v_{21}[v'_{22}], t_{i+1}, \dots, t_k]]$. It is easy to see that $\tau_{M, \bar{q}}$ is infinite, $\text{occ}(u^{(1)}, x_1)$ and $\text{occ}(v^{(1)}, x_1)$ are incomparable, and $\text{length}(\text{occ}(u^{(1)}, x_1)) < \text{length}(\alpha)$ which contradicts that α is minimal.

Now we show that $\text{length}(\alpha_1) \leq \|Q\|$. To derive a contradiction, assume that $\text{length}(\alpha_1) > \|Q\|$. Then, by Lemma 4.11, there are trees, $u'_1 \in \widehat{T}_\Sigma(X_1)$ and $v'_1 \in \widehat{T}_\Delta(X_1)$ such that $\text{length}(\text{occ}(u'_1, x_1)) \leq \|Q\|$, $\text{length}(\alpha_1) - \text{length}(\text{occ}(u'_1, x_1)) = \text{length}(\beta_1) - \text{length}(\text{occ}(v'_1, x_1))$ and $u'_1[p(x_1)] \Rightarrow_M^* q_0(v'_1)$.

Let $u^{(2)} = u'_1[\sigma(s_1, \dots, s_{i-1}, u_2, s_{i+1}, \dots, s_k)]$. Clearly $\text{length}(\text{occ}(u^{(2)}, x_1)) < \text{length}(\alpha)$ and there is a derivation $u^{(2)}[q(x_1)] \Rightarrow_M^* q_0(v^{(2)})$ of M , where $v^{(2)} = v'_1[r[t_1, \dots, t_{i-1}, v_2, t_{i+1}, \dots, t_k]]$. It can be seen (cf. Observation 4.6) that $\text{occ}(u^{(2)}, x_1)$ and $\text{occ}(v^{(2)}, x_1)$ are incomparable, which again contradicts that α is minimal.

Now, since $\text{length}(\alpha_1) \leq \|Q\|$, it is easy to see that $\text{length}(\alpha_1 \alpha_{21}) = \text{length}(\beta_1 \omega) \leq (\|Q\| + 1)mx$, where

$$mx = \max\{\text{height}(r) \mid \exists \mu \in R, p \in Q : p(r) \text{ is the right-hand side of } \mu\}.$$

Then $\text{length}(\alpha) = \text{length}(\alpha_1 \alpha_{21} \alpha_{22}) \leq (\|Q\| + 1)mx + 1$.

So, we have proved that $\text{length}(\alpha) \leq \max\{3\|Q\| + 1, (\|Q\| + 1)mx + 1\}$, i.e. it is bounded by a number which depends on M , what we wanted to show. \blacksquare

In the following lemma we show that a shape preserving bottom-up tree transducer is also occurrence preserving.

Lemma 4.14 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a shape preserving bottom-up tree transducer. Then M is occurrence preserving.

Proof. We prove the lemma by contradiction. Therefore let us suppose that M is not occurrence preserving. Then by Item (b) of Definition 4.8, there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$ of M , where $q \in Q$ such that $\tau_{M, q}$ is infinite, $u \in \widehat{T}_\Sigma(X_1)$ and $v \in \widehat{T}_\Delta(X_1)$, such that α and β are incomparable, where $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$. Since $\tau_{M, q}$ is infinite, there are trees $u_1, \bar{u}_1 \in \text{dom}(\tau_{M, q})$ such that $u_1 \not\approx \bar{u}_1$. Then there are derivations $s = u[u_1] \Rightarrow_M^* u[q(v_1)] \Rightarrow_M^* q_0(v[v_1]) = q_0(t)$ and $\bar{s} =$

$u[\bar{u}_1] \Rightarrow_M^* u[q(\bar{v}_1)] \Rightarrow_M^* q_0(v[\bar{v}_1]) = q_0(\bar{t})$, where $t, \bar{t}, v_1, \bar{v}_1 \in T_\Delta$. Since M is shape preserving, $s \approx t$ and $\bar{s} \approx \bar{t}$. Then, since α and β are incomparable, $u_1 = \text{stree}(s, \alpha) \approx \text{stree}(t, \alpha) = \text{stree}(\bar{t}, \alpha) \approx \text{stree}(\bar{s}, \alpha) = \bar{u}_1$, which contradicts that $u_1 \not\approx \bar{u}_1$, proving the lemma. ■

Let us consider now a bottom-up tree transducer $M = (Q, \Sigma, \Delta, q_0, R)$. In the definition of the transformable tree transducer (cf. Definition 3.35) we required the transformable tree transducer to have the property that if a rule μ of the transducer satisfies the conditions of Definition 3.16, then $\|mp(\mu)\| = 1$. Therefore, to decide whether M is transformable or not, we have to compute the set $mp(\mu)$ for every rule μ in R which satisfies the conditions of Definition 3.16. In the next two lemmas we show that $mp(\mu)$ is computable for periodic and occurrence preserving transducers as follows. Let us suppose that M is periodic and occurrence preserving. Moreover let μ be a rule in R such that $\|inf(\mu)\| = 0$ and $\tau_{M,q}$ is infinite, where q is the state occurring in the right-hand side of μ . Let $\gamma \in mp(\mu)$ be a matching path of μ . We show that there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$ of M , where $u \in \hat{T}_\Sigma(X_1)$ and $v \in \hat{T}_\Delta(X_1)$, such that $\alpha = \beta\gamma$ if γ is a right matching path of μ and $\beta = \alpha\gamma$ otherwise, moreover $length(\alpha) \leq n$ where n is a positive number depending on M . This clearly implies that the set $mp(\mu)$ is computable.

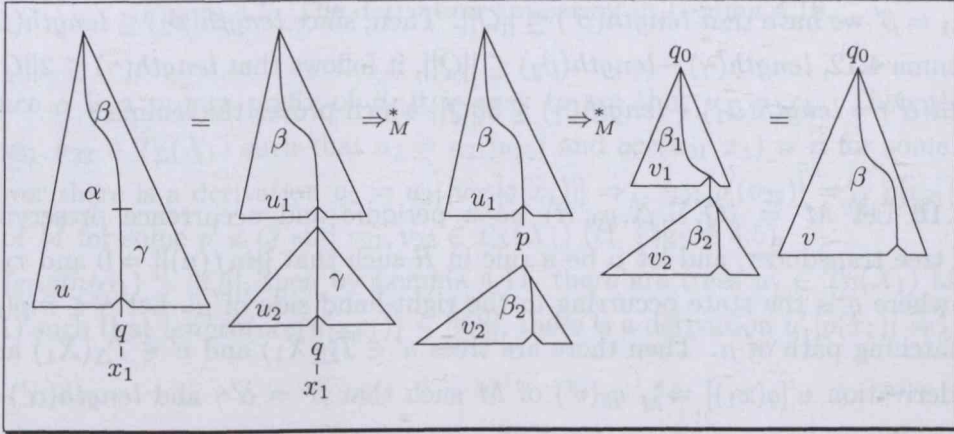


Figure 4.4: The derivation appearing in Lemma 4.15.

Lemma 4.15 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a periodic and occurrence preserving bottom-up tree transducer, and let μ be a rule in R such that $\|inf(\mu)\| = 0$ and $\tau_{M,q}$ is infinite, where q is the state occurring in the right-hand side of μ . Let $\gamma \in mp(\mu)$ be a right matching path of μ . Then there are trees $u' \in \hat{T}_\Sigma(X_1)$ and $v' \in \hat{T}_\Delta(X_1)$ and there

is a derivation $u'[q(x_1)] \Rightarrow_M^* q_0(v')$ of M such that $\alpha' = \beta'\gamma$ and $\text{length}(\alpha') \leq 3||Q||$, where $\alpha' = \text{occ}(u', x_1)$ and $\beta = \text{occ}(v', x_1)$.

Proof. Since γ is a right matching path of μ , by Definition 3.16, there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$ for some $u \in \widehat{T}_\Sigma(X_1)$ and $v \in \widehat{T}_\Delta(X_1)$ such that $\alpha = \beta\gamma$, where $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$.

Let $u = u_1[u_2]$, where $u_1, u_2 \in \widehat{T}_\Sigma(X_1)$, such that $\text{occ}(u_1, x_1) = \beta$. Then $\text{occ}(u_2, x_1) = \gamma$ and there is a state $p \in Q$ such that the derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$ can be written in the form $u[q(x_1)] = u_1[u_2[q(x_1)]] \Rightarrow_M^* u_1[p(v_2)] \Rightarrow_M^* q_0(v_1[v_2]) = q_0(v)$, where $v_1, v_2 \in \widehat{T}_\Delta(X_1)$. Let $\beta_1 = \text{occ}(v_1, x_1)$ and $\beta_2 = \text{occ}(v_2, x_1)$ (cf. Figure 4.4).

If $\text{length}(\beta) > ||Q||$, then by Lemma 4.11, there are trees $u'_1 \in \widehat{T}_\Sigma(X_1)$ and $v'_1 \in \widehat{T}_\Delta(X_1)$ such that $\text{length}(\text{occ}(u'_1, x_1)) \leq ||Q||$, $u'_1[p(x_1)] \Rightarrow_M^* q_0(v'_1)$ and

$$\text{length}(\beta) - \text{length}(\text{occ}(u'_1, x_1)) = \text{length}(\beta_1) - \text{length}(\text{occ}(v'_1, x_1)). \quad (4.1)$$

Now let $\bar{u}_1 = u_1$, $\bar{v}_1 = v_1$ if $\text{length}(\beta) \leq ||Q||$ and let $\bar{u}_1 = u'_1$, $\bar{v}_1 = v'_1$ otherwise. Moreover let $\bar{\alpha}_1 = \text{occ}(\bar{u}_1, x_1)$, $u' = \bar{u}_1[u_2]$ and $v' = \bar{v}_1[v_2]$. Clearly $\text{length}(\bar{\alpha}_1) \leq ||Q||$, $u'[q(x_1)] \Rightarrow_M^* q_0(v')$ and since M is occurrence preserving, α' and β' are comparable, where $\alpha' = \text{occ}(u', x_1)$ and $\beta' = \text{occ}(v', x_1)$. It can be seen that $\alpha' = \bar{\alpha}_1\gamma$. Moreover, by equation 4.1, we have that $\bar{\alpha}_1 = \beta'$. Thus $\alpha' = \beta'\gamma$. It remains to show that $\text{length}(\alpha') \leq 3||Q||$.

Since $\bar{\alpha}_1 = \beta'$ we have that $\text{length}(\beta') \leq ||Q||$. Then, since $\text{length}(\beta_2) \leq \text{length}(\beta')$ and, by Lemma 4.12, $\text{length}(\gamma) - \text{length}(\beta_2) \leq ||Q||$, it follows that $\text{length}(\gamma) \leq 2||Q||$. Then $\text{length}(\alpha') = \text{length}(\bar{\alpha}_1) + \text{length}(\gamma) \leq 3||Q||$ which proves the lemma. ■

Lemma 4.16 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a periodic and occurrence preserving bottom-up tree transducer, and let μ be a rule in R such that $||\text{inf}(\mu)|| = 0$ and $\tau_{M,q}$ is infinite, where q is the state occurring in the right-hand side of μ . Let $\gamma \in \text{mp}(\mu)$ be a left matching path of μ . Then there are trees $u' \in \widehat{T}_\Sigma(X_1)$ and $v' \in \widehat{T}_\Delta(X_1)$ and there is a derivation $u'[q(x_1)] \Rightarrow_M^* q_0(v')$ of M such that $\beta' = \alpha'\gamma$ and $\text{length}(\alpha') \leq (||Q||+1)m\alpha+2||Q||$, where $\alpha' = \text{occ}(u', x_1)$, $\beta' = \text{occ}(v', x_1)$ and $m\alpha = \max\{\text{height}(r) \mid \exists \mu \in R, p \in Q : p(r) \text{ is the right-hand side of } \mu\}$.

Proof. Since γ is a left matching path of μ , by Definition 3.16 there is a derivation $u[q(x_1)] \Rightarrow_M^* q_0(v)$ where $u \in \widehat{T}_\Sigma(X_1)$ and $v \in \widehat{T}_\Delta(X_1)$ such that $\beta = \alpha\gamma$ and α is a proper prefix of β , where $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$.

Now let α_1 be the longest occurrence in $\text{occ}(u)$ such that the following holds. There are trees $u_1, u_2 \in \widehat{T}_\Sigma(X_1)$ and $v_1, v_2 \in \widehat{T}_\Delta(X_1)$ such that $u_1[u_2] = u$, $v_1[v_2] = v$, $\text{occ}(u_1, x_1) = \alpha_1$, $u_1[u_2[q(x_1)]] \Rightarrow_M^* u_1[p(v_2)] \Rightarrow_M^* q_0(v_1[v_2])$, for some $p \in Q$, and $\text{occ}(v_1, x_1)$ is a prefix of α . Let $\alpha_2 = \text{occ}(u_2, x_1)$, $\beta_1 = \text{occ}(v_1, x_1)$ and $\beta_2 = \text{occ}(v_2, x_1)$.

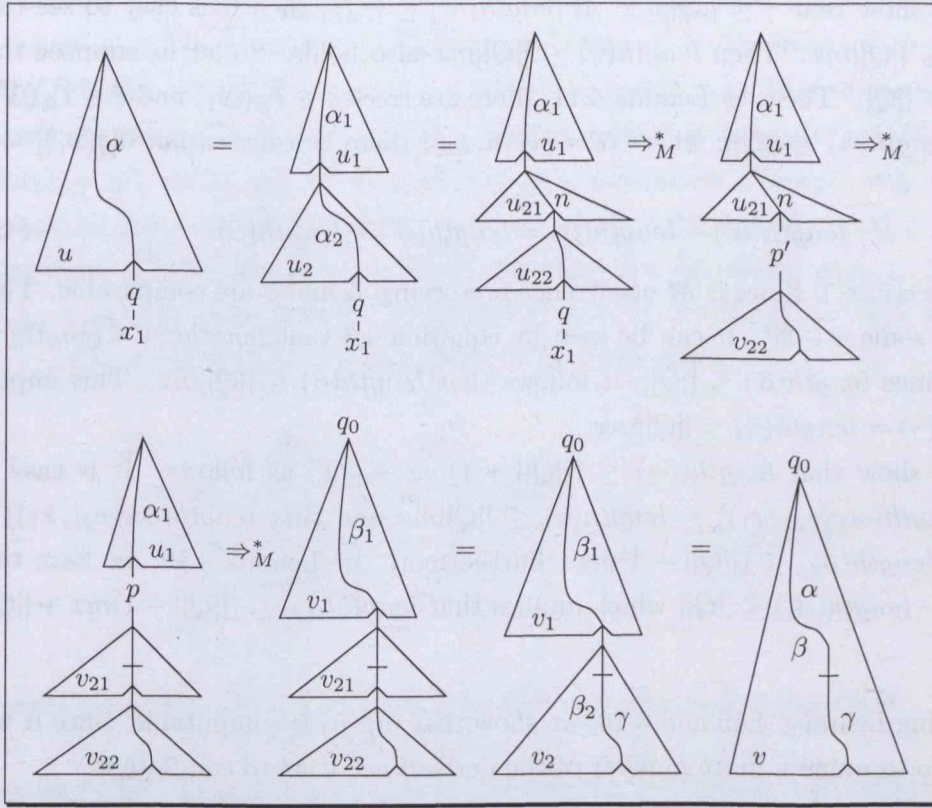


Figure 4.5: The derivation appearing in Lemma 4.16.

Since α is a proper prefix of β , it is easy to see that $u_2 \neq x_1$. Then there are trees $u_{21}, u_{22} \in \hat{T}_\Sigma(X_1)$ such that $u_2 = u_{21}[u_{22}]$ and $\text{occ}(u_{21}, x_1) = n$ for some $n \in \mathbb{N}$. Moreover there is a derivation $u_2 = u_{21}[u_{22}[q(x_1)]] \Rightarrow_M^* u_{21}[p'(v_{22})] \Rightarrow_M^* p(v_{21}[v_{22}]) = p(v_2)$ of M for some $p' \in Q$ and $v_{21}, v_{22} \in \hat{T}_\Delta(X_1)$ (cf. Figure 4.5).

If $\text{length}(\alpha_1) > \|Q\|$, then by Lemma 4.11, there are trees $u'_1 \in \hat{T}_\Sigma(X_1)$ and $v'_1 \in \hat{T}_\Delta(X_1)$ such that $\text{length}(\text{occ}(u'_1, x_1)) \leq \|Q\|$, there is a derivation $u'_1[p(x_1)] \Rightarrow_M^* q_0(v'_1)$ and

$$\text{length}(\alpha_1) - \text{length}(\text{occ}(u'_1, x_1)) = \text{length}(\beta_1) - \text{length}(\text{occ}(v'_1, x_1)). \quad (4.2)$$

Now let $\bar{u}_1 = u_1$ and $\bar{v}_1 = v_1$ if $\text{length}(\alpha_1) \leq \|Q\|$ and let $\bar{u}_1 = u'_1$ and $\bar{v}_1 = v'_1$ otherwise. Moreover, let $\bar{\alpha}_1 = \text{occ}(\bar{u}_1, x_1)$, $u' = \bar{u}_1[u_2]$, $v' = \bar{v}_1[v_2]$, $\alpha' = \text{occ}(u', x_1)$ and $\beta' = \text{occ}(v', x_1)$. Clearly $\text{length}(\bar{\alpha}_1) \leq \|Q\|$ and there is a derivation $u'[q(x_1)] = q_0(v')$ of M . Additionally, since M is occurrence preserving, we have that α' and β' are comparable. Then it can be seen with the help of equation 4.2 that $\beta' = \alpha'\gamma$. It remains to show that $\text{length}(\alpha') = \text{length}(\bar{\alpha}_1) + \text{length}(\alpha_2) \leq (\|Q\| + 1)mx + 2\|Q\|$. Since $\text{length}(\bar{\alpha}_1) \leq \|Q\|$, it is enough to show that $\text{length}(\alpha_2) \leq (\|Q\| + 1)mx + \|Q\|$.

First we show that $\gamma \leq \|Q\|mx$. If $\text{length}(\alpha') \leq \|Q\|$, then it is easy to see that $\text{length}(\beta') \leq \|Q\|mx$. Then $\text{length}(\gamma) \leq \|Q\|mx$ also holds. So let us suppose that $\text{length}(\alpha') > \|Q\|$. Then, by Lemma 4.11, there are trees $\bar{u} \in \widehat{T}_\Sigma(X_1)$ and $\bar{v} \in \widehat{T}_\Delta(X_1)$, such that $\text{length}(\bar{\alpha}) \leq \|Q\|$, where $\bar{\alpha} = \text{occ}(\bar{u}, x_1)$, there is a derivation $\bar{u}[q(x_1)] \Rightarrow_M^* q_0(\bar{v})$ and

$$\text{length}(\alpha') - \text{length}(\bar{\alpha}) = \text{length}(\beta') - \text{length}(\bar{\beta}), \quad (4.3)$$

where $\bar{\beta} = \text{occ}(\bar{v}, x_1)$. Since M is occurrence preserving, $\bar{\alpha}$ and $\bar{\beta}$ are comparable. Then $\bar{\beta} = \bar{\alpha}\bar{\gamma}$, for some $\bar{\gamma} \in \mathbb{N}^*$. It can be seen by equation 4.3 that $\text{length}(\bar{\gamma}) = \text{length}(\gamma)$. Moreover, since $\text{length}(\bar{\alpha}) \leq \|Q\|$, it follows that $\text{length}(\bar{\beta}) \leq \|Q\|mx$. This implies that $\text{length}(\gamma) = \text{length}(\bar{\gamma}) \leq \|Q\|mx$.

Now we show that $\text{length}(\alpha_2) \leq (\|Q\| + 1)mx + \|Q\|$ as follows. It is easy to see that $\text{length}(\text{occ}(v_{22}, x_1)) \leq \text{length}(\gamma) \leq \|Q\|mx$ and that $\text{length}(\text{occ}(v_{21}, x_1)) \leq mx$. Then $\text{length}(\beta_2) \leq (\|Q\| + 1)mx$. Furthermore, by Lemma 4.12, we have that $\text{length}(\alpha_2) - \text{length}(\beta_2) \leq \|Q\|$ which implies that $\text{length}(\alpha_2) \leq (\|Q\| + 1)mx + \|Q\|$. ■

Now, using Lemmas 4.15 and 4.16, we show that $mp(\mu)$ is computable, since it will be enough to examine a finite number of the derivations of M to compute it.

Lemma 4.17 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a periodic and occurrence preserving bottom-up tree transducer, and let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\tau_{M,q}$ is infinite and $\|\text{inf}(\mu)\| \neq 1$. Then $mp(\mu)$ is computable.

Proof. If $\|\text{inf}(\mu)\| > 1$ then $mp(\mu)$ is computable by Definition 3.16. So let us suppose that $\|\text{inf}(\mu)\| = 0$. Let $\gamma \in mp(\mu)$. We give a derivation $s' \Rightarrow_M^* q_0(t')$, where $s' \in T_\Sigma$ and $t' \in T_\Delta$, such that γ can be determined from this derivation and $\text{height}(s')$ is bounded by a number which depends only on M . This will clearly imply that $mp(\mu)$ is computable.

Let us assume that γ is a right matching path (resp. γ is a left matching path) of μ . Then, by Definition 3.16, there is a derivation

$$\begin{aligned} s &= u[\sigma(s_1, \dots, s_k)] \\ &\Rightarrow_M^* u[\sigma(q_1(t_1), \dots, q_k(t_k))] \\ &\Rightarrow_M u[q(r[t_1, \dots, t_k])] && \text{(rule } \mu) \\ &\Rightarrow_M^* q_0(v[r[t_1, \dots, t_k]]) \\ &= q_0(t) \end{aligned}$$

of M , where $s \in T_\Sigma$, $u \in \widehat{T}_\Sigma(X_1)$, $v \in \widehat{T}_\Delta(X_1)$, $s_1, \dots, s_k \in T_\Sigma$, $t_1, \dots, t_k \in T_\Delta$, $t \in T_\Delta$ such that $\alpha = \beta\gamma$ (resp. $\beta = \alpha\gamma$), where $\alpha = \text{occ}(u, x_1)$ and $\beta = \text{occ}(v, x_1)$. Let $Mx = \max\{(\|Q\| + 1)mx + 2\|Q\|, 3\|Q\|\}$, where $mx = \max\{\text{height}(r') \mid \exists \mu' \in R, p \in Q : p(r') \text{ is the right-hand side of } \mu'\}$.

By Lemma 4.15 (resp. by Lemma 4.16), there is a derivation $u'[q(x_1)] \Rightarrow_M^* q_0(v')$ of M , where $u' \in \widehat{T}_\Sigma(X_1)$ and $v' \in \widehat{T}_\Delta(X_1)$, such that $\alpha' = \beta'\gamma$ (resp. $\beta' = \alpha'\gamma$) and $\text{length}(\alpha') \leq Mx$, where $\alpha' = \text{occ}(u', x_1)$ and $\beta' = \text{occ}(v', x_1)$. Then it is easy to see that there are trees $\bar{u}' \in \widehat{T}_\Sigma(X_1)$ and $\bar{v}' \in \widehat{T}_\Delta(X_1)$ such that $\text{height}(\bar{u}') \leq Mx + \|Q\|$, $\text{occ}(\bar{u}', x_1) = \alpha'$, $\text{occ}(\bar{v}', x_1) = \beta'$ and there is a derivation $\bar{u}'[q(x_1)] \Rightarrow_M^* q_0(\bar{v}')$ of M . It can be seen also that there are trees $s'_1, \dots, s'_k \in T_\Sigma$ and $t'_1, \dots, t'_k \in T_\Delta$ such that, for every $i \in [k]$, $\text{height}(s_i) \leq \|Q\|$ and there is a derivation $\sigma(s'_1, \dots, s'_k) \Rightarrow_M^* \sigma(q_1(t'_1), \dots, q_k(t'_k)) \Rightarrow_M q(r[t'_1, \dots, t'_k])$ of M . Let $s' = \bar{u}'[\sigma(s'_1, \dots, s'_k)]$ and $t' = \bar{v}'[r[t'_1, \dots, t'_k]]$. Clearly there is a derivation of M of the form

$$\begin{aligned}
 s' &= \bar{u}'[\sigma(s'_1, \dots, s'_k)] \\
 &\Rightarrow_M^* \bar{u}'[\sigma(q_1(t'_1), \dots, q_k(t'_k))] \\
 &\Rightarrow_M \bar{u}'[q(r[t'_1, \dots, t'_k])] && (\text{rule } \mu) \\
 &\Rightarrow_M^* q_0(\bar{v}'[r[t'_1, \dots, t'_k]]) \\
 &= q_0(t'),
 \end{aligned}$$

and $\text{height}(s') \leq Mx + 2\|Q\| + 1$, which implies that $mp(\mu)$ is computable. \blacksquare

Now let M be a periodic and occurrence preserving bottom-up tree transducer. The next lemma shows that it is decidable whether M is transformable.

Lemma 4.18 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a periodic and occurrence preserving bottom-up tree transducer. It is decidable whether M is transformable.

Proof. Let q be a state in Q . By Corollary 3.12 of [Eng75], $\text{dom}(\tau_{M,q})$ is recognizable, therefore it is decidable if $\tau_{M,q}$ is infinite (cf. Theorem II.10.4 in [GS84] and note that $\tau_{M,q}$ is infinite if and only if $\text{dom}(\tau_{M,q})$ is infinite). Then it is also decidable if τ_M is infinite. Let $\mu = \sigma(q_1(x_1), \dots, q_k(x_k)) \rightarrow q(r)$ be a rule in R such that $\tau_{M,q}$ is infinite. Since $\text{inf}(\mu)$ is computable, it is decidable if $\|\text{inf}(\mu)\| \neq 1$. We show that conditions (i)-(iv) of Definition 3.35 are also decidable. Condition (i) is clearly decidable. By Lemma 4.17, the set $mp(\mu)$ is computable, therefore Condition (ii) is also decidable.

Now, assume that $\|mp(\mu)\| = 1$, $\|\text{inf}(\mu)\| = 0$ and let $\gamma = mp(\mu)$. Using Lemma 4.17 one can decide whether γ is either a right matching path or a left matching path of μ . Moreover, since $\|\text{inf}(\mu)\| = 0$, the sets $S = \{\sigma(s_1, \dots, s_k) \mid \forall i \in [k] : s_i \in \text{dom}(\tau_{M,q_i})\}$ and $T = \{r[t_1, \dots, t_k] \mid \forall i \in [k] : t_i \in \text{ran}(\tau_{M,q_i})\}$ are computable. If γ is a left matching path of μ , then one can decide whether for every $s \in S$, $\gamma \in \text{occ}(s)$ or not. Likewise, if γ is a right matching path of μ , then one can decide whether for every $t \in T$, $\gamma \in \text{occ}(t)$ or not. This implies that Condition (iii) is decidable.

Finally, let us suppose that $\|mp(\mu)\| = 1$ and $\|\text{inf}(\mu)\| > 1$. Let $\gamma = mp(\mu)$. Then one can easily decide whether Condition (iv) holds or not. \blacksquare

Now, using Lemmas 4.5 and 4.18, we can decide whether a bottom-up tree transducer is shape preserving or not as follows.

Theorem 4.19 Let $M = (Q, \Sigma, \Delta, q_0, R)$ be a bottom-up tree transducer. Then it is decidable whether M is shape preserving or not.

Proof. We give an algorithm that terminates with yes if M is shape preserving, otherwise it terminates with no. The algorithm is as follows.

1. Check whether τ_M is finite (cf. the proof of Lemma 4.18). If τ_M is finite, then go to Step 2, otherwise go to Step 3.
2. Check if M is shape preserving or not. (It is clearly decidable whether a finite tree transformation is shape preserving or not.) If M is shape preserving, then halt with yes, otherwise halt with no.
3. Check whether M is periodic and occurrence preserving or not. (By Lemma 4.9 it is decidable whether M is periodic, and if M is periodic, then by Lemma 4.13 it is decidable whether M is occurrence-preserving.) If M is not periodic or it is not occurrence preserving, then halt with no, because by one of the corresponding lemmas (Lemmas 4.10 and 4.14), M is not shape preserving.
4. Check whether M is transformable. (By Lemma 4.18 it is decidable whether a periodic and occurrence preserving bottom-up tree transducer is transformable or not.)
5. If M is not transformable, then halt with no. (Since τ_M is infinite, by Lemma 3.36, M is not shape preserving.)
6. Check whether M is shape preserving (Lemma 4.5). If M is shape preserving, then halt with yes, otherwise halt with no.

■

4.2 Decidability of the Equivalence of Shape Preserving Tree Transducers

Here we show that the equivalence problem for shape preserving tree transducers is decidable. This is a consequence of that, as we have seen in the previous section, for every shape preserving tree transducer, an equivalent relabeling tree transducer can be constructed, and the fact that the equivalence problem for relabelings is decidable.

Corollary 4.20 Let M and N be two shape preserving tree transducers. Then it is decidable whether $\tau_M = \tau_N$.

Proof. As we have seen in the proof of Lemma 4.18 it is decidable if a bottom-up tree transducer computes an infinite tree transformation. Using Theorem II.1 of [Rou70], the analogous result can be shown for top-down tree transducers.

If τ_M and τ_N are finite, then it is clearly decidable if $\tau_M = \tau_N$. If only one of them is finite, then $\tau_M \neq \tau_N$. Therefore assume that τ_M and τ_N are infinite. By Theorem 4.4 and Lemmas 3.36 and 4.5 two relabelings \bar{M} and \bar{N} can be constructed such that $\tau_M = \tau_{\bar{M}}$ and $\tau_N = \tau_{\bar{N}}$. Moreover, by Theorem 1 of [AB93], the equivalence problem of relabelings is decidable, hence it is decidable whether $\tau_{\bar{M}} = \tau_{\bar{N}}$ holds or not. Thus it is also decidable if $\tau_M = \tau_N$. ■

4.2 Decidability of the Equivalence of Shape Preserving Tree Transducers

Here we show that the equivalence problem for shape preserving tree transducers is decidable. This is a consequence of the fact that we have seen, in the previous section, that shape preserving tree transducers, in general, are not collapsing tree transducers (see construction), and the fact that the equivalence problem for collapsing is undecidable.

Chapter 5

Conclusions

As the first main result of the Thesis, it was showed that shape preserving (top-down or bottom-up) tree transducers and relabeling tree transducers compute the same tree transformation class (cf. Corollary 3.71). In this way a dynamic property of tree transformations, i.e., the shape preserving property was characterized, by a static property of tree transducers, i.e. that they are relabelings. Our result also naturally generalizes the analogous characterization of length preserving gsm's.

The second main result of the Thesis is that the shape preserving property is decidable for both top-down tree transducers and bottom-up tree transducers. Moreover, as a byproduct of the above results, it was shown that the equivalence problem of shape preserving tree transducers is also decidable.

In the following, we show two examples of the application of our results. First we recall a result concerning top-down relabeling tree transformations. In [LST98] the smallest class of transductions which contains length-preserving rational transductions and is closed under union, composition and iteration, was considered. They gave several interesting characterizations of this class. Recently Z. Fülöp and A. Terlutte were going to generalize the results of [LST98] to the class of shape preserving top-down tree transducers [FT02]. However, they could generalize the results only to the class of relabeling tree transducers. They gave a characterization of $UCI(QREL)$, where U, C and I stand for union, composition and iteration, respectively, in terms of a short expression built up from $QREL$ with composition and iteration. They also gave a characterization of $UCI(QREL)$ in terms of one-step rewrite relations of very simple term rewrite systems. Now, using Corollary 3.71, $QREL$ in the above description can be replaced by $SHAPE$. In this way, the results of [LST98] can be generalized to shape preserving tree transducers.

Next, let us consider a well known result from the theory of tree transducers that linear tree transformations preserve recognizability of tree languages [Rou70], [Eng75].

Clearly relabeling tree transducers are also linear, thus it easily follows from Corollary 3.71 that shape preserving tree transducers also preserve recognizability.

Finally, we recall another famous result, namely that $BOT = QREL \circ HOM$ (Theorem 3.15 of [Eng75]), which expresses that every tree transformation τ computed by a bottom-up tree transducer can be decomposed as $\tau = \tau_1 \circ \tau_2$, where τ_1 and τ_2 are computed by a relabeling tree transducer and a homomorphism tree transducer, respectively (this latter transducer is a top-down or bottom-up tree transducer which has only one state, can read every input symbol, and does not have different rules with the same left-hand side). By Corollary 3.71 the class $QREL$ in this equation can be replaced by the class $SHAPE$.

Summary

In this Thesis shape preserving top-down and bottom-up tree transducers were studied. As the first main result of the Thesis, we gave a characterization of these transducers by relabelings. In fact we showed that $SHAPE = QREL$ (Corollary 3.71), where $SHAPE$ and $QREL$ are the classes of tree transformations computed by shape preserving top-down or shape preserving bottom-up tree transducers and relabelings, respectively. The shape preserving property is a semantical property of the tree transducers, since it is a property of the tree transformations computed by them. On the other hand, relabelings have strict restrictions on their syntax, i.e. on their rewriting rules. Therefore, the above result characterizes a semantical property of tree transducers by a syntactical one. As the second main result of the Thesis, it was shown that the shape preserving property of both top-down tree transducers and bottom-up tree transducers is decidable (Theorems 4.4 and 4.19).

In Chapter 1 we gave a brief introduction concerning tree transducers. The first section described the role of tree transducers in the theory of syntax-directed translation. In the next section top-down and bottom-up tree transducers were discussed. Then we considered the shape preserving versions of these transducers. We described the reason of that a shape preserving top-down tree transducer can not delete the direct subtrees of a node of an input tree, while shape preserving bottom-up tree transducers can do it. Finally, we described the problems which we have solved in the Thesis, and gave the outline of the Thesis.

In Chapter 2 the necessary definitions and preliminary results were given. Additionally, we gave two examples of shape preserving tree transducers, a top-down and a bottom-up one, and we examined their certain derivations. In fact, we used these transducers and derivations as running examples throughout the Thesis.

Chapter 3 deals with the first main result of the Thesis. Here we proved the above mentioned characterization of shape preserving tree transducers, i.e. that $SHAPE = QREL$. It is obvious that relabelings are shape preserving tree transducers. To show that $SHAPE \subseteq QREL$ we needed the following preparation.

In Subsection 3.1.1 useful properties of shape preserving top-down tree transducers

were given. In fact we showed that, for obvious reasons, a shape preserving top-down tree transducer can not delete or copy the direct subtrees of a node of an input tree. However, it turned out that it can permute those subtrees. Using these results, we showed that every shape preserving top-down tree transducer is a permutation top-down quasirelabeling tree transducer (Lemma 3.5).

In Subsection 3.2.1 we gave an algorithm which eliminates the permutation rules from a shape preserving permutation quasirelabeling. In this way we proved that every shape preserving top-down tree transducer is equivalent to a top-down quasirelabeling (Lemma 3.31).

A top-down quasirelabeling is a tree transducer which scanning an input symbol with rank $k \neq 1$ writes out exactly one output symbol with rank k , and maybe some additional unary output symbols. In Subsection 3.3.1 we developed a method which, roughly, deals with these additional unary symbols. Using this method it was shown that every shape preserving top-down quasirelabeling is equivalent to a top-down relabeling.

Summarizing the above results we could prove that every shape preserving top-down tree transducer is equivalent to a top-down relabeling tree transducer (Theorem 3.64).

In order to show the other main result of this chapter, namely that every shape preserving bottom-up tree transducer is also equivalent to a relabeling, we made the following preparation steps.

In Subsection 3.1.2 useful properties of shape preserving bottom-up tree transducers were given. Then, in Subsection 3.2.2, the concept of transformable tree transducers was introduced (Definition 3.35). Using the results obtained in Subsection 3.1.2 we showed that a shape preserving bottom-up tree transducer is transformable, provided that it computes an infinite tree transformation (Lemma 3.36).

Then we introduced the concept of the frame transducer of a transformable tree transducer (Definition 3.38). It turned out that there is a strong connection between a transformable tree transducer M and its frame transducer $fr(M)$. In fact, we showed that $\tau_M = g^{-1} \circ \tau_{fr(M)} \circ h$, where g and h are certain tree homomorphisms determined by the input and output ranked alphabets of $fr(M)$ (Corollary 3.44).

In Subsection 3.2.3 it was shown that if the tree transformation computed by a shape preserving bottom-up tree transducer M is infinite, then the frame transducer of M is also shape preserving (Lemma 3.52). Using Theorem 3.64, we introduced the relabeling frame transducer of M (Definition 3.65), which is a bottom-up relabeling tree transducer equivalent to the frame transducer of M .

In Subsection 3.3.2 we showed certain properties of the relabeling frame transducer of a shape preserving bottom-up tree transducer. Then, using these properties and the

above results, we concluded the second main result of the chapter, namely that every shape preserving bottom-up tree transducer is equivalent to a bottom-up relabeling tree transducer (Theorem 3.70).

In Chapter 4 some decidability results were given. In Section 4.1 we showed that the shape preserving property of a top-down or bottom-up tree transducer is decidable as follows.

In Subsection 4.1.1 an algorithm was given which decides whether a top-down tree transducer is shape preserving or not. Moreover, if it is shape preserving, then the algorithm outputs a relabeling equivalent to it (Theorem 4.4).

In Subsection 4.1.2 the analogous result was given for bottom-up tree transducers. First, based on the results of Chapter 3, we showed that it is decidable if a transformable bottom-up tree transducer is shape preserving or not (Lemma 4.5). Moreover, we showed, that it is also decidable if a bottom-up tree transducer, which satisfies certain decidable conditions, is shape preserving or not (Lemma 4.18). Using these result, we concluded, that it is decidable if a bottom-up tree transducer is shape preserving or not (Theorem 4.19).

In Lemma 4.5, we also showed that if a transformable bottom-up tree transducer is shape preserving, then an equivalent relabeling can be constructed. Moreover, it is not difficult to see that if the tree transformation computed by a bottom-up tree transducer is shape preserving and finite, then again an equivalent relabeling can be given. Using these results and that every shape preserving bottom-up tree transducer which induces infinite tree transformation is transformable (Lemma 3.36), we concluded that for every shape preserving bottom-up tree transducer, an equivalent relabeling can be constructed.

It was shown in [AB93] that the equivalence problem for relabeling tree transducers is decidable. As a corollary of this result and the fact that we can construct an equivalent relabeling for every shape preserving (top-down or bottom-up) tree transducer, we showed that the equivalence problem of shape preserving tree transducers is also decidable (Corollary 4.20).

In Chapter 5 some conclusions were given. We recalled several results from the theory of tree transducers which concern relabelings. Then, using the equivalence of shape preserving tree transducers and relabelings (Corollary 3.71), we generalized these results to the class of tree transformations computed by shape preserving tree transducers.

Összefoglalás

(Summary in Hungarian)

Az értekezésben alakmegőrző felszálló és leszálló fatranszformátorokat tanulmányoztunk. Az értekezés első fő eredménye ezeknek a fatranszformátoroknak átcímkező fatranszformátorokkal való jellemzése. Lényegében megmutattuk, hogy $SHAPE = QREL$ (3.71. Következmény), ahol $SHAPE$ az alakmegőrző leszálló illetve az alakmegőrző felszálló fatranszformátorok által, $QREL$ pedig az átcímkezők által kiszámolt fatranszformációk osztálya.

Az alakmegőrző tulajdonság egy szemantikus tulajdonsága a fatranszformátoroknak, mivel ez az általuk kiszámított fatranszformációk tulajdonsága. Másrészt, az átcímkezőknek szigorú megkötései vannak a szintaxisukat, azaz az átírási szabályaikat tekintve. Ebből adódóan, a fenti eredmény a fatranszformátorok egy szemantikus tulajdonságát jellemzi egy szintaktikussal. A értekezés második fő eredményeképpen megmutattuk, hogy a felszálló fatranszformátorok és a leszálló fatranszformátorok alakmegőrző tulajdonsága eldönthető (4.4. és 4.19. Tétel).

Az 1. fejezet a fatranszformátorok egy rövid bevezetése. Az első alfejezet leírja a fatranszformátorok szerepét a szintaxis-vezérelt fordításban. Az ezt követő alfejezetben a felszálló és leszálló fatranszformátorokat ismertettük. Ezután ezen fatranszformátorok alakmegőrző változatait tárgyaltuk. Megmutattuk, hogy egy alakmegőrző felszálló fatranszformátor miért nem tudja törölni a bemeneti fa egy szögpontjának közvetlen részfáit, amíg az alakmegőrző leszálló fatranszformátor ezt megteheti. Végül ismertettük azokat a problémákat, amelyeket az értekezésben megoldottunk, és megadtuk az értekezés felépítését.

A 2. fejezetben tárgyaltuk a szükséges definíciókat és jelöléseket. Emellett adtunk két példát alakmegőrző fatranszformátorokra, egy felszállót és egy leszállót, majd megvizsgáltuk ezek néhány derivációját. Később ezeket a fatranszformátorokat és derivációkat, mint alkalmas példákat, többször is felhasználtuk az értekezés folyamán.

A 3. fejezet az értekezés első fő eredményét ismerteti. Bebizonyítottuk a fent említett tulajdonságát az alakmegőrző fatranszformátoroknak, azaz megmutattuk, hogy az

általuk kiszámolt fatranszformációk osztálya megegyezik az átcímkezők által kiszámított osztállyal. Az nyilvánvaló, hogy az átcímkezők alakmegőrző fatranszformátorok. A másik irány megmutatása jelenti a fejezet érdemi részét.

A 3.1.1. alfejezet az alakmegőrző felszálló fatranszformátorok hasznos tulajdonságait tárgyalja. Megmutattuk, hogy nyilvánvaló módon egy alakmegőrző felszálló fatranszformátor nem törölheti vagy másolhatja egy bemeneti fa szögpontjainak közvetlen részfáit. Viszont, az is kiderült, hogy permutálhatja a szögpontok ezen részfáit. Ezen eredmények segítségével megmutattuk, hogy minden alakmegőrző felszálló fatranszformátor egy permutációs kváziátcímkező (3.5. Lemma).

A 3.2.1. alfejezetben egy algoritmust adtunk, amely eltávolítja egy alakmegőrző permutációs kváziátcímkező permutációs szabályait, megőrizve az eredeti fatranszformációt. Így bebizonyítottuk, hogy minden alakmegőrző felszálló fatranszformátor ekvivalens egy kváziátcímkezővel (3.31. Lemma).

A felszálló kváziátcímkező egy olyan fatranszformátor, amely egy $k \neq 1$ rangú input szimbólumot beolvassa, pontosan egy, az input szimbólummal megegyező rangú output szimbólumot ír ki és esetleg még néhány unáris output szimbólumot. A 3.3.1. alfejezetben kidolgoztunk egy módszert, ami ezeket az unáris szimbólumokat kezeli. A módszer segítségével megmutattuk, hogy minden alakmegőrző felszálló kváziátcímkező ekvivalens egy felszálló átcímkezővel.

A fenti eredményeket felhasználva bebizonyítottuk, hogy minden alakmegőrző felszálló fatranszformátor ekvivalens egy felszálló átcímkező fatranszformátorral (3.64. Tétel).

Annak érdekében, hogy megmutassuk a fejezet másik fő eredményét, nevezetesen, hogy minden alakmegőrző leszálló fatranszformátor is ekvivalens egy átcímkezővel, a következő előkészületeket tettük.

A 3.1.2. alfejezet az alakmegőrző leszálló fatranszformátorok hasznos tulajdonságait tárgyalja. A 3.2.2. alfejezetben bevezetjük a transzformálható fatranszformátorokat (3.35. Definíció). A 3.1.2. alfejezet eredményeinek segítségével pedig megmutatjuk, hogy egy végtelen fatranszformációt kiszámító alakmegőrző fatranszformátor transzformálható (3.36. Lemma).

Azután bevezettük a transzformálható fatranszformátorok keret-fatranszformátorának fogalmát (3.38. Definíció). Megmutattunk egy erős kapcsolatot egy M transzformálható fatranszformátor és annak $fr(M)$ keret-fatranszformátora között. Nevezetesen, megmutattuk, hogy $\tau_M = g^{-1} \circ \tau_{fr(M)} \circ h$, ahol g és h olyan fahomomorfizmusok, amelyek $fr(M)$ bemeneti és kimeneti rangolt ábécéinek segítségével határozhatók meg (3.44. Következmény).

A 3.2.3. alfejezetben bebizonyítottuk, hogy ha egy M alakmegőrző leszálló fatranszformátor által kiszámított fatranszformáció végtelen, akkor M keret-fatranszformátora

szintén alakmegőrző (3.52. Lemma). A 3.64. tétel segítségével definiáltuk M átcímkező keret-fatranszformátorát (3.65. Definíció), ami egy, az M keret-fatranszformátorával ekvivalens átcímkező fatranszformátor.

A 3.3.2. alfejezetben megmutattuk a keret-fatranszformátorok néhány fontos tulajdonságát. Felhasználva ezeket a tulajdonságokat, valamint a fenti eredményeket, kimondtuk a fejezet második fő eredményét, nevezetesen, hogy minden alakmegőrző leszálló fatranszformátor ekvivalens egy leszálló átcímkező fatranszformátorral (3.70. Tétel).

A 4. fejezetben eldönthetőségi problémákkal foglalkoztunk. A 4.1. alfejezetben azt mutattuk meg, hogy egy tetszőleges felszálló vagy leszálló fatranszformátorról eldönthető, hogy alakmegőrző-e.

A 4.1.1. alfejezetben foglalkoztunk a felszálló esettel. Megadtunk egy algoritmust, amely eldönti, hogy egy felszálló fatranszformátor alakmegőrző-e. Továbbá, ha a fatranszformátor alakmegőrző, akkor az algoritmus megkonstruálja a vele ekvivalens átcímkezőt (4.4. Tétel).

A 4.1.2. alfejezetben mutattuk meg azt, hogy a leszálló fatranszformátorok alakmegőrző tulajdonsága is eldönthető. A következőképpen jártunk el. Először, a 3. fejezet eredményeit felhasználva, megmutattuk, hogy eldönthető, vajon egy transzformálható leszálló fatranszformátor alakmegőrző-e vagy sem (4.5. Lemma). Megmutattuk továbbá, hogy egy, bizonyos eldönthető tulajdonságokkal rendelkező, leszálló fatranszformátorról eldönthető, hogy transzformálható-e (4.18. Lemma). Ezen eredmények felhasználásával megállapítottuk, hogy a leszálló fatranszformátorok alakmegőrző tulajdonsága is eldönthető (4.19. Tétel).

A 4.5. Lemmában azt is megmutattuk, hogy ha egy transzformálható fatranszformátor alakmegőrző, akkor megkonstruálható egy vele ekvivalens átcímkező. Nem nehéz belátni azt, hogy egy véges alakmegőrző fatranszformációhoz is megadható egy őt kiszámító átcímkező. Ezt felhasználva, valamint azt, hogy minden végtelen alakmegőrző fatranszformációt kiszámító leszálló fatranszformátor transzformálható (3.36. Lemma), megállapítottuk, hogy minden alakmegőrző leszálló fatranszformátorhoz megadható egy vele ekvivalens átcímkező.

Ismert eredmény továbbá, hogy az átcímkező fatranszformátorok ekvivalenciája eldönthető [AB93]. Felhasználva ezt, valamint azt, hogy a fenti eredmények alapján minden alakmegőrző (felszálló vagy leszálló) fatranszformátorhoz megadható egy vele ekvivalens átcímkező, megállapítottuk, hogy az alakmegőrző (felszálló vagy leszálló) fatranszformátorok ekvivalenciája szintén eldönthető (4.20. Következmény).

Az 5. fejezetben megadtuk az értekezés eredményeinek néhány következményét. Felidézünk néhány, a *QREL* osztályra vonatkozó, eredményt és a 3.71. Következmény felhasználásával ezeket az eredményeket általánosítottuk az alakmegőrző fatranszformátorok osztályára.

Bibliography

- [AB93] Y. Andre and F. Bossut. Decidability of equivalence for linear letter to letter top-down tree transducers. In *FCT 93*, pages 142–151, 1993.
- [ASU86] A.V. Aho, R. Sethi, and J.D. Ullman. *Compilers — Principles, Techniques, and Tools*. Addison–Wesley, 1986.
- [AU77] A.V. Aho and J.D. Ullman. *Principles of Compiler Design*. Addison–Wesley, 1977.
- [Bac59] J. W. Backus. The syntax and semantics of the proposed international algebraic language of the zurich ACM-GAMM conference. In *IFIP Congress*, pages 125–131, 1959.
- [Bak78] B. S. Baker. Tree transducers and tree languages. *Inform. and Control*, 37:241–166, 1978.
- [Bak79] B. S. Baker. Composition of top-down and bottom-up tree transductions. *Inform. and Control*, 41:186–213, 1979.
- [Ber79] J. Berstel. *Transductions on Context-Free Languages*. Teubner Stuttgart, 1979.
- [BR82] J. Berstel and C. Reutenauer. Recognizable power series on trees. *Theoret. Comput. Sci.*, 18:115–148, 1982.
- [Cho56] N. Chomsky. Three models for the description of language. *Information Theory, IEEE Transactions on*, 2(3):113–124, 1956.
- [Don65] J. Doner. Decidability of the weak second-order theory of two successors. *Notices Amer. Math. Soc.*, 12:Astract No. 65T 648, 819, 1965.
- [Don70] J. Doner. Tree acceptors and some of their applications. *J. Comput. System Sci.*, 4:406–451, 1970.

- [EFV02] J. Engelfriet, Z. Fülöp, and H. Vogler. Bottom-up and Top-down Tree Series Transformations. *Journal of Automata, Languages and Combinatorics*, 7:11–70, 2002.
- [Eil74] S. Eilenberg. *Automata, Languages and Machines, Vol. A*. Academic Press, 1974.
- [ÉK03] Zoltán Ésik and Werner Kuich. Formal tree series. *Journal of Automata, Languages and Combinatorics*, 8(2):219–285, 2003.
- [EM99] J. Engelfriet and S. Maneth. Macro Tree Transducers, Attribute Grammars, and MSO Definable Tree Translations. *Inform. and Comput.*, 154:34–91, 1999.
- [EM03] J. Engelfriet and S. Maneth. Macro tree translations of linear size increase are mso definable. *SIAM Journal of Computing*, 32(4):950–1006, 2003.
- [Eng75] J. Engelfriet. Bottom-up and top-down tree transformations — A comparison. *Math. Systems Theory*, 9:198–231, 1975.
- [Eng77] J. Engelfriet. Top-down tree transducers with regular look-ahead. *Math. Systems Theory*, 10:289–303, 1977.
- [Eng78] J. Engelfriet. On tree transducers for partial functions. *Inform. Process. Letters*, 7(4):170–172, 1978.
- [Eng82] J. Engelfriet. The Copying Power of One-State Tree Transducers. *J. Comput. System Sci.*, 25(3):418–435, 1982.
- [Ési80] Z. Ésik. Decidability results concerning tree transducers I. *Acta Cybernet.*, 5:1–20, 1980.
- [Ési83] Z. Ésik. Decidability results concerning tree transducers II. *Acta Cybernet.*, 6:303–314, 1983.
- [EV85] J. Engelfriet and H. Vogler. Macro tree transducers. *J. Comput. System Sci.*, 31:71–146, 1985.
- [EV91] J. Engelfriet and H. Vogler. Modular tree transducers. *Theoret. Comput. Sci.*, 78:267–304, 1991.
- [EV94] J. Engelfriet and H. Vogler. The translation power of top-down tree-to-graph transducers. *J. Comput. System Sci.*, 49:258–305, 1994.

- [FG03] Z. Fülöp and Zs. Gazdag. Shape preserving top-down tree transducers. *Theoret. Comput. Sci.*, 304:315–339, 2003.
- [FT02] Z. Fülöp and A. Terlutte. Iterated relabeling tree transducers. *Theoret. Comput. Sci.*, 276:221–244, 2002.
- [Fül81] Z. Fülöp. On attributed tree transducers. *Acta Cybernet.*, 5:261–279, 1981.
- [FV92] Z. Fülöp and S. Vágvolgyi. Decidability of the inclusion in monoids generated by tree transformation classes. In M. Nivat and A. Podelski, editors, *Tree Automata and Languages*, pages 381–408. Elsevier Science Publishers B.V., 1992.
- [FV98] Z. Fülöp and H. Vogler. *Syntax-Directed Semantics — Formal Models Based on Tree Transducers*. Monographs in Theoretical Computer Science, An EATCS Series. Springer-Verlag, 1998.
- [FV03] Z. Fülöp and H. Vogler. Tree Series Transformations that Respect Copying. *Theory of Computing Systems*, 36:247–293, 2003.
- [Gaz06a] Zs. Gazdag. Decidability of the shape preserving property of bottom-up tree transducers. *To appear in International Journal of Foundations of Computer Science*, 2006.
- [Gaz06b] Zs. Gazdag. Shape preserving bottom-up tree transducers. *To appear in Journal of Automata, Languages and Combinatorics*, 2006.
- [Géc81] F. Gécseg. Tree transformations preserving recognizability. In *Finite Algebra and Multiple-Valued Logic, Szeged, 1979, Coll. Math. Soc. J. Bolyai, Volume 28*, pages 251–273. North-Holland, Amsterdam, 1981.
- [Gin66] S. Ginsburg. *The mathematical theory of context free languages*. McGraw-Hill Book Company, New York, 1966.
- [GS84] F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.
- [GS97] F. Gécseg and M. Steinby. Tree languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 1–68. Springer-Verlag, 1997.
- [Iro61] E. T. Irons. A syntax directed compiler for ALGOL 60. *Comm. of the ACM*, 4:51–55, 1961.

- [Joh79] Steven C. Johnson. Yacc: Yet another compiler compiler. In *UNIX Programmer's Manual*, volume 2, pages 353–387. Holt, Rinehart, and Winston, New York, NY, USA, 1979.
- [Knu68] D.E. Knuth. Semantics of context-free languages. *Math. Systems Theory*, 2:127–145, 1968.
- [Kui99] W. Kuich. Tree transducers and formal tree series. *Acta Cybernetica*, 14:135–149, 1999.
- [Leg81] J. Leguy. Transductions rationnelles décroissantes. *R. A. I. R. O. Theoretical Informatics*, 5:141–148, 1981.
- [LST98] M. Latteux, D. Simplot, and A. Terlutte. Iterated length-preserving rational transductions. In *Proc. of MFCS 98*, volume 1450 of *Lecture Notes in Computer Science*, pages 286–295. Springer Verlag, 1998.
- [Man03] S. Maneth. The macro tree transducer hierarchy collapses for functions of linear size increase. In *FSTTCS*, pages 326–337, 2003.
- [Mea55] G. H. Mealy. A method for synthesizing sequential circuits. *Bell System Techn. J.*, 34:1045–79, 1955.
- [NV00] T. Noll and H. Vogler. The universality of higher-ordered attributed tree transducers. *Theory of Computing Systems*, to appear, 2000.
- [Rou68] W. C. Rounds. *Trees, transducers and transformations*. Ph d thesis, Stanford University, USA, 1968.
- [Rou70] W.C. Rounds. Mappings and grammars on trees. *Math. Systems Theory*, 4:257–287, 1970.
- [Tha69] J.W. Thatcher. Generalized² sequential machine maps. IBM Res. Report RC 2466, 1969.
- [Tha70] J.W. Thatcher. Generalized² sequential machine maps. *J. Comput. System Sci.*, 4:339–367, 1970.
- [Tha73] J.W. Thatcher. Tree automata: an informal survey. In A.V. Aho, editor, *Currents in the Theory of Computing*, pages 143–172. Prentice Hall, 1973.
- [TW65] J.W. Thatcher and J.B. Wright. Generalized finite automata. *Notices Amer. Math. Soc.*, 12:Astract No. 65T 649, 820, 1965.

- [TW68] J.W. Thatcher and J.B. Wright. Generalized finite automata theory with application to a decision problem of second-order logic. *Math. Systems Theory*, 2(1):57–81, 1968.

Glossary

\Rightarrow_M^*	Reflexive, transitive closure of \Rightarrow_M , 6, 7, 19
\Rightarrow_M	Derivation relation computed by M , 18
$<$	Strict partial order over state sets, 45
\subset	Proper subset relation, 15
\subseteq	Subset relation, 15
\emptyset	Empty set, 15
$ A $	Cardinality of A , 15
A^*	Set of strings over A , 16
$A^{*,k}$	Set of strings in A^* with length at most k , 16
A^+	$A^* - \{\varepsilon\}$, 16
$A \rightarrow B$	(partial) mapping from A to B , 15
$\text{bn}(s)$	Branch number of s , 26
$\text{dec}(s)$	Decomposition of s , 53
$\text{dom}(\rho)$	Domain of ρ , 15
ε	Empty string, 16
$\text{fr}(M)$	Frame transducer of M , 58
γt	Notation of $\gamma[t]$, 16
$\text{height}(s)$	Height of s , 17
$\text{inf}(\mu)$	$\{i \in [k] \mid \tau_{M,q_i} \text{ is infinite}\}$ (k is the arity of the symbol scanned by μ and q_1, \dots, q_k are the states on the left-hand side of μ), 33
$[k]$	Set of natural numbers $1, 2, \dots, k$, 15
$\text{length}(w)$	Length of w , 16
$\text{mp}(\mu)$	Set of matching paths of μ , 34
\mathbb{N}	Set of all natural numbers, 15
$\text{occ}(s)$	Set of occurrences of s , 17
$\text{occ}(s, x_i)$	Unique occurrence of x_i in s , 17
$Q(A)$	$\{q(a) \mid q \in Q \text{ and } a \in A\}$, 17

$QREL$	Set of tree transformations computed by relabelings, 9, 21
$ran(\rho)$	Range of ρ , 15
$rfr(M)$	Relabeling frame transducer of M , 82
$\rho(a)$	$\{b \mid (a, b) \in \rho\}$, 15
ρ^{-1}	Inverse of ρ , 15
$\rho_1 \circ \rho_2$	Composition of relations ρ_1 and ρ_2 , 11, 15
$R_1 \circ R_2$	Composition of relation classes R_1 and R_2 , 15
$s \approx t$	s and t have the same shape, 17
$SHAPE$	Set of tree transformations computed by shape preserving transducers, 12, 21
$(\Sigma^{(1)})^*$	Notation of $T_{\Sigma^{(1)}}(X_1)$, 16
$\Sigma^{(k)}$	Set of symbols in Σ with rank k , 16
$\sigma^{(k)}$	Symbol with rank k , 16
$\langle \Sigma, \Delta \rangle$	Ranked alphabet obtained from Σ and Δ by $\langle \Sigma, \Delta \rangle^{(k)} = \Sigma^{(k)} \times \Delta^{(k)}$, for every $k \geq 0$, 77
Σ_{\diamond}	Ranked alphabet Σ with a unary symbol \diamond , 77
$stree(s, w)$	Subtree of s at w , 17
τ_M	Tree transformation computed by M , 19
$\tau_{M,q}$	Tree transformation computed by M in a state q , 19
T_{Σ}	Set of trees over Σ , 16
$T_{\Sigma}(A)$	Set of trees over Σ indexed by the elements of A , 16
$\hat{T}_{\Sigma}(X_k)$	Smallest subset of $T_{\Sigma}(X_k)$ containing those trees in $T_{\Sigma}(X)$ which contain the variables from X_k in the order $x_1 < x_2 < \dots < x_k$, 16
$t[t_1, \dots, t_k]$	Tree substitution, 16
t^n	$t[t[\dots[t]]]$, where t occurs n times in the substitution, 16
$w(i)$	i th letter of w , 16
$wbn(s)$	Weighted branch number of s , 26
X	Set of all variable symbols, 16
X_k	Set of variables x_1, x_2, \dots, x_k , 16

Index

- alphabet, 16
 - ranked, 3, 16
 - input, 5, 18
 - output, 5, 18
 - tree, 53
- attributed tree transducer, 4
- Backus Naur Form, 2
- bottom-up
 - quasirelabeling tree transducer, 21
 - relabeling tree transducer, 21
 - tree transducer, 5, 6, 18
 - linear, 21
 - nondeleting, 21
 - shape preserving, 21
- bounded, 79
 - k -, 79
- branch
 - number, 26
 - weighted, 26
 - symbol, 26
- comparable occurrences, 16
- compiler, 1
- composition of relations, 15
- context free grammars, 2
- copying state, 30
- decomposition of a tree, 53
- deleting state, 30
- derivation
 - relation, 18
 - reflexive, transitive closure of, 19
 - tree, 2
- equivalence relation, 16
- equivalent tree transducers, 19
- final state, 6, 18
- frame transducer, 59
 - relabeling, 83
- generalized sequential machine (gsm), 12
 - length preserving, 12
- height of a tree, 17
- incomparable occurrences, 16
- initial state, 5, 18
- inverse of a relation, 15
- k -bounded, 79
- language
 - object, 1
 - programming, 1
 - source, 1
 - target, 1
 - tree, 16
- length of a string, 16
- linear size increase, 11
- macro tree transducer, 4
- mapping, 15
- matching path, 34
 - left, 35
 - right, 35
- maximal segment, 71

- Mealy automaton, 12
- occurrence preserving bottom-up tree transducer, 93
- occurrences, 17
- parsing, 2
 - tree, 2
- part of a tree, 52
- partial
 - mapping, 15
- periodic
 - bottom-up tree transducer, 93
 - top-down tree transducer, 47
- permutation
 - degree, 45
 - of a set, 15
 - quasirelabeling, 20
 - rule, 45
 - state, 45
 - top-down quasirelabeling tree transducer, 20
- prefix, 16
 - proper, 16
- quasirelabeling, 20, 21
 - permutation, 20
- ranked alphabet, 3, 16
 - input, 5, 18
 - output, 5, 18
- recognizable tree language, 10, 21
- relabeling, 20, 21
- relabeling frame transducer, 83
- relation, 15
 - domain of, 15
 - irreflexive, 15
 - range of, 15
 - reflexive, 15
 - symmetric, 15
 - transitive, 16
- rewriting rule, 5, 18
- rule, 18
 - left-hand side, 18
 - permutation, 45
 - rewriting, 5
 - right-hand side, 18
 - useful, 20
- segment
 - maximal, 71
 - of a tree, 71
- semantics
 - of a tree transducer, 5
 - of the compilation, 2
- set of strings (words), 16
- shape
 - preserving
 - bottom-up tree transducer, 21
 - frame, 80
 - top-down tree transducer, 21
 - tree transformation, 17
 - same, 8, 17
- single-use restriction, 11
- state, 18
 - copying, 30
 - deleting, 30
 - designated, 18
 - final, 6, 18
 - initial, 5, 18
 - noncopying, 30
 - nondeleting, 30
 - permutation, 45
 - useful, 20
- strict partial order, 16
- subtree, 17
 - at w , 17
 - proper, 17
- syntax, 1

- directed
 - compilation, 2
 - semantics, 2
- of a tree transducer, 5
- of the source language, 1
- top-down
 - quasirelabeling tree transducer, 20
 - permutation, 20
 - relabeling tree transducer, 20
 - tree automaton, 21
 - tree transducer, 5, 18
 - linear, 20
 - nondeleting, 20
 - shape preserving, 21
 - with regular look-ahead, 10
- transformable tree transducer, 54
- translation of the input program, 2
- tree, 2
 - alphabet, 53
 - derivation, 2
 - homomorphism, 18
 - input, 5, 20
 - language, 16
 - k -bounded, 79
 - bounded, 79
 - recognizable, 10, 21
 - node of, 2
 - output, 5, 20
 - over Σ indexed by A , 16
 - parsing, 2
 - part of, 52
 - root of, 2
 - segment of, 71
 - substitution, 16
 - transformation, 4, 5, 16
 - computed by M , 6, 7, 19
 - shape preserving, 17
- tree transducer, 18
 - attributed, 4
 - bottom-up, 5, 6, 18
 - occurrence preserving, 93
 - periodic, 93
 - macro, 4
 - relabeling, 8
 - shape preserving, 8
 - top-down, 5, 18
 - periodic, 47
 - with regular look-ahead, 10
 - transformable, 54
- useful
 - rule, 20
 - state, 20
- variable symbols, 16
- weighted branch number, 26