Shape Preserving Tree Transducers

Abstract of the Ph.D. Thesis

Zsolt Gazdag

Supervisor:

Prof. Zoltán Fülöp

University of Szeged
Department of Foundations of
Computer Science

2006

Introduction

Tree transducers were introduced in ([Rou70, Tha70] and [Tha73]). Since they serve formal models for syntax-directed compilers, they have been studied intensively (see e.g. [Eng75, Bak78, Bak79, Ési80, Eng82, Ési83, GS84, FV92, FV98] and [GS97]).

In tree transducer theory, trees are usually well formed terms over a finite operator domain, called ranked alphabet. Moreover, they are visualized as finite, labeled, ordered, and directed trees in the graph theoretic sense. A tree transformation is a binary relation over trees. A tree transducer M is a term rewrite system which computes a tree transformation τ_M, called the tree transformation computed by M.

We investigate two basic types of tree transducers, namely top-down tree transducers [Rou70, Tha70] and bottom-up tree transducers [Tha73]. Top-down tree transducers process an input tree from its root towards its leaves. Bottom-up tree transducers work in the opposite direction, i.e they process an input tree from its leaves towards its root. Moreover, a tree transducer processes an input tree according to its rewriting rules. These rules describe the syntax of the transducer, while the transformation computed by the transducer can be regarded as the semantic of the transducer. The classes of tree transformations computed by top-down tree transducers and bottom-up tree transducers are denoted by TOP and BOT, respectively.

We consider shape preserving top-down and bottom-up tree transducers. Two trees have the same shape if they differ only in the labels of their nodes (cf. Figure 1). A tree transformation τ is shape preserving if, for every $(s,t) \in \tau$, the trees s and t have the same shape. A tree transducer is shape preserving if it computes a shape preserving tree transformation. We denote the class of tree transformations computed by shape preserving top-down and shape preserving bottom-up tree transducers by $SHAPE$.

Top-down relabeling tree transducers and bottom-up relabeling tree transducers ([Eng75]) are shape preserving. These transducers scanning an input symbol with a certain rank, write out exactly one output symbol with the same rank. It was shown in [Eng75] that top-down relabeling tree transducers and bottom up relabeling tree transducers compute the same class of tree transformations, which we denote by $QREL$.

I is obvious, that "being relabeling" is a restriction on the syntax of a transducer, since this restriction concerns the rewriting rules of the transducer. On the other hand, the shape preserving property is a property of the semantics of the tree transducer, since it is a restriction on the tree transformation computed by the transducer.

Many results in tree transducer theory belong to the following problem class. Let C be a tree transformation class computed by tree transducers of a certain kind. Moreover, let C' be a (natural) subclass of C. Now, the problem consists of the following two questions. Is there a (natural) syntactic restriction of the model 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 $\tau_M = \tau$? Given a tree transducer M of that certain kind, is the tree transformation τ_M in C'? If the answer to the first question is positive, then we have characterized a semantic restriction made on C by a syntactic restriction made on the tree transducer model.

In the Thesis we show a result which is in the above problem class. In fact, we characterize shape preserving top-down and bottom-up tree transformations by relabeling tree transducers. Obviously, $SHAPE \subseteq TOP \cup BOT$. 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). Since also $QREL \subseteq SHAPE$, 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 have characterized 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).

The results of the Thesis are published in [FG03, Gaz06b] and [Gaz06a]. The numbering of the theorems and lemmas in this abstract is the same as that of in the Thesis.

In the Thesis we use standard term rewriting constructions. Moreover, we also use common tree transducer theoretic tools, for example the concept of tree homomorphisms. In the proofs of the decidability results we often apply a well known method in tree automata and tree transducers theory, called standard pumping arguments. Using this method it will be enough to examine only a finite number of input and output trees of a transducer when we want to decide whether it has a certain property or not.

Results of the Thesis

Definitions and Notations

Let Σ be a finite ranked alphabet and A a set disjoint with Σ. The set of (*finite, labeled and ordered*) *trees over* Σ *indexed by* A is denoted by $T_\Sigma(A)$. T_Σ denotes the set $T_\Sigma(\emptyset)$. For every $k \geq 0$, we denote by $\Sigma^{(k)}$ the set of those symbols in Σ which have rank k.

A *tree language* is a subset of T_Σ and a *tree transformation* is a subset of $T_\Sigma \times T_\Delta$. We will need the set $X = \{x_1, x_2, \ldots\}$ of *variable symbols*. For every $k \geq 0$, we define $X_k = \{x_1, \ldots, x_k\}$.

The *tree substitution* is defined as follows. Let $t \in T_\Sigma(X_k)$ and let $t_1, \ldots, t_k$ be also trees over (maybe other) ranked alphabets. Then $t[t_1, \ldots, t_k]$ 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.

A *tree homomorphisms* $h : T_\Sigma \to T_\Delta$ induced by the mapping $\overline{h} : \Sigma \to T_\Delta(X)$ with the property that if $\sigma \in \Sigma^{(k)}$ for some $k \geq 0$, then $\overline{h}(\sigma) \in T_\Delta(X_k)$, is defined as follows. If $s = \sigma(s_1, \ldots, s_k)$ for some $k \geq 0$, $\sigma \in \Sigma^{(k)}$ and $s_1, \ldots, s_k \in T_\Sigma$, then $h(s) = \overline{h}(\sigma)[h(s_1), \ldots, h(s_k)]$.

Two trees $s \in T_\Sigma$ and $t \in T_\Delta$ *have the same shape*, if they differ only in the labels of their nodes (see Figure 1). A tree transformation $\tau \subseteq T_\Sigma \times T_\Delta$ is *shape preserving* if, for every $(s, t) \in \tau$, s and t have the same shape.

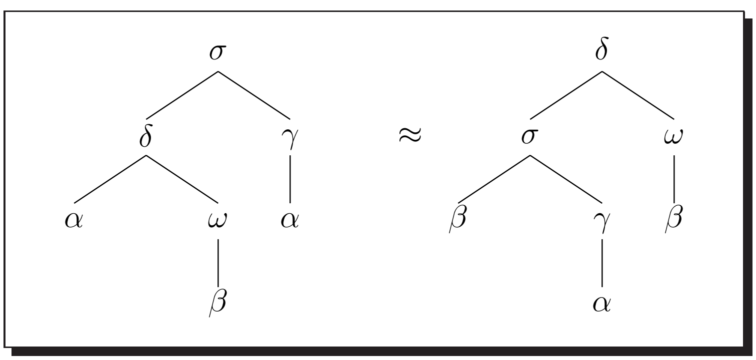

Figure 1: Trees which have the same shape.

Next we define top-down and bottom-up tree transducers.

Definition 1 ([Rou70, Tha70, Tha73]) A *tree transducer* 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 $Q \cap (\Sigma \cup \Delta) = \varnothing$; $q_0 \in Q$ is the *designated state*; and R is a finite set of *rewriting rules* such that either every rule in R is of the form

$$(\dagger) \quad q(\sigma(x_1, \ldots, x_k)) \to r$$

with $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q \in Q$ and $r \in T_\Delta(Q(X_k))$ or every rule in R is of the form

$$(\ddagger) \quad \sigma(q_1(x_1), \ldots, q_k(x_k)) \to q(r)$$

with $k \geq 0$, $\sigma \in \Sigma^{(k)}$, $q, q_1, \ldots, q_k \in Q$ and $r \in T_\Delta(X_k)$. In Case ($\dagger$) M is called a *top-down* tree transducer and q_0 is the *initial state* while in Case ($\ddagger$) M is a *bottom-up* tree transducer and q_0 is the *final state*.

The *derivation relation* induced by a tree transducer M is a binary relation $\Rightarrow_M$ over the set $T_{Q\cup\Sigma\cup\Delta}$ defined as follows. If M is a top-down tree transducer, then for $u, v \in T_{Q\cup\Sigma\cup\Delta}$, we write $u \Rightarrow_M v$ if and only if there is a rule ($\dagger$) in R and v is obtained from u by replacing an occurrence of a subtree $q(\sigma(u_1, \ldots, u_k))$ of u by $r[u_1, \ldots, u_k]$, where $u_1, \ldots, u_k \in T_\Sigma$. If M is a bottom-up tree transducer, then for $u, v \in T_{Q\cup\Sigma\cup\Delta}$, we write $u \Rightarrow_M v$ if and only if there is a rule ($\ddagger$) in R and v is obtained from u by replacing an occurrence of a subtree $\sigma(q_1(v_1), \ldots, q_k(v_k))$ of u by $q(r[v_1, \ldots, v_k])$, where $v_1, \ldots, v_k \in T_\Delta$. The reflexive, transitive closure of $\Rightarrow_M$ is denoted by $\Rightarrow_M^*$. Then the tree transformation computed by M is the relation

$$\tau_M = \begin{cases} \{\, (s,t) \in T_\Sigma \times T_\Delta \mid q_0(s) \Rightarrow_M^* t\}, & \text{if } M \text{ is a top-down tree transducer,} \\ \{\, (s,t) \in T_\Sigma \times T_\Delta \mid s \Rightarrow_M^* q_0(t)\}, & \text{if } M \text{ is a bottom-up tree transducer.} \end{cases}$$

Two tree transducers M and M' are *equivalent* if $\tau_M = \tau_{M'}$.

We say that a tree transducer M is a *top-down relabeling tree transducer* (resp. *bottom-up relabeling tree transducer*) if every rule in R is of the form

$$q(\sigma(x_1, \ldots, x_k)) \to \delta(q_1(x_1), \ldots, q_k(x_k))$$

$$(\text{resp. } \sigma(q_1(x_1), \ldots, q_k(x_k)) \to q(\delta(x_1, \ldots, x_k))),$$

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

In [Eng75] it was shown that relabeling top-down tree transducers are equivalent to relabeling bottom-up tree transducers, therefore we will also refer these transducers as *relabelings*. The set of tree transformations computed by relabelings is denoted by $QREL$.

A tree transducer M is *shape preserving* if τ_M is shape preserving. We denote by $SHAPE$ the set of all tree transformations computed by top-down or bottom-up shape preserving tree transducers.

Characterizing Shape Preserving Tree Transducers

In this chapter of the Thesis we show that $SHAPE = QREL$ holds. The proof of that shape preserving top-down tree transducers are equivalent to relabelings needs the following preparation.

A top-down tree transducer $M = (Q, \Sigma, \Delta, q_0, R)$ is a *permutation quasirelabeling* if the following holds. Let (†) be a rule in R as in Definition 1. Then either (i) $k \neq 1$ and $r = \gamma\delta(\gamma_1 q_1(x_{\pi(1)}), \ldots, \gamma_k q_k(x_{\pi(k)}))$, where $\delta \in \Delta^{(k)}$, $\gamma, \gamma_1, \ldots, \gamma_k$ are sequences of symbols from $(\Delta^{(1)})$ and π is a permutation of the set $\{1, 2, \ldots, k\}$, or (ii) $k = 1$ and $r = \gamma p(x_1)$, where γ is the same as in Case (i).

In Subsection 3.1.1 we show those properties of shape preserving top-down tree transducers with which we can prove the following lemma.

Lemma 3.5 *Every shape preserving top-down tree transducer is a permutation quasirelabeling.*

A top-down tree transducer M is a *top-down quasirelabeling* if it is a permutation quasirelabeling, but only the identity permutation can occur on the right-hand side of its rules.

In Subsection 3.2.1 we show that the permutation rules can be eliminated from a shape preserving permutation quasirelabeling.

Lemma 3.31 *For every shape preserving permutation quasirelabeling an equivalent top-down quasirelabeling can be constructed.*

It Subsection 3.3.1 we develop a method which we can use to show that every shape preserving quasirelabeling is equivalent to a relabeling.

Using the above results we can prove the following theorem.

Theorem 3.64 ([FG03]) *Every shape preserving top-down tree transducer is equivalent to a top-down relabeling tree transducer.*

To show the other main result of this chapter, namely that every shape preserving bottom-up tree transducer is equivalent to a relabeling, we have to make the following preparation.

In Subsection 3.1.2 we show useful properties of shape preserving bottom-up tree transducer. Then, in Subsection 3.2.2, we introduce the concept of *transformable* tree transducers. A transformable tree transducer is a bottom-up tree transducer satisfying certain properties which we will use when we construct a frame transducer of a transformable tree transducer.

Using the results of Subsection 3.1.2, we can prove the following lemma.

Lemma 3.36 *Let M be a shape preserving bottom-up tree transducer. If τ_M is infinite, then M is transformable.*

Then we introduce the concept of the *frame transducer*s of transformable tree transducers. There is a strong connection between a transformable tree transducer M and its frame transducer $fr(M)$, as we will see in the next lemma. Moreover, $fr(M)$ has certain nice properties, for example it is a bottom-up quasirelabeling, which make our further work much easier. (A bottom-up quasirelabeling can be defined analogously to the top-down case.)

Corollary 3.44 *Let M be a transformable bottom-up tree transducer, and let $fr(M)$ be the frame transducer of M. Then $\tau_M = g^{-1} \circ \tau_{fr(M)} \circ h$.*

In the above corollary g and h are certain tree homomorphisms which are determined by the input and output ranked alphabets of $fr(M)$. Moreover, the operation $\circ$ is the usual composition operation of relations.

In Subsection 3.2.3 we prove the following lemma which states a very important property of the frame transducer of a shape preserving bottom-up tree transducer.

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.*

Let M be a shape preserving bottom-up tree transducer such that τ_M is infinite. Using Theorem 3.64, we can define the relabeling frame transducer $rfr(M)$ of M, which is a relabeling equivalent to $fr(M)$.

In Subsection 3.3.2 we give an equivalent relabeling to M using its relabeling frame transducer $rfr(M)$. Therefore we can state the second main result of the chapter.

Theorem 3.70 ([Gaz06b]) *Every shape preserving bottom-up tree transducer M is equivalent to a bottom-up relabeling tree transducer.*

Using Theorems 3.64 and 3.70, we get the following corollary.

Corollary 3.71 $SHAPE = QREL$.

Decidability results

This chapter is devoted to the decidability results of the Thesis. We show that it is decidable whether a given (top-down or bottom-up) tree transducer computes a shape preserving tree transformation.

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. We prove the following theorem by giving an algorithm which decides whether a given top-down tree transducer M is shape preserving, and if M is shape preserving then the algorithm outputs an equivalent relabeling.

Theorem 4.4 ([FG03]) *It is decidable if a top-down tree transducer M is shape preserving or not. Moreover, if M is shape preserving, then a relabeling $\bar{M}$ can be constructed such that $\tau_M = \tau_{\bar{M}}$.*

In Subsection 4.1.2 we deal with bottom-up tree transducers. We show that it is decidable whether a transformable tree transducer M is shape preserving. In fact, similarly to the top-down case, we give an algorithm which decides whether a given transformable bottom-up tree transducer M is shape preserving, and if M is shape preserving then the algorithm outputs an equivalent relabeling.

Lemma 4.5 *Let M 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 $\tau_M = \tau_{\bar{M}}$.*

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 *Let M be a periodic and occurrence preserving bottom-up tree transducer. It is decidable whether M is transformable.*

Using these results we can prove that the shape preserving property of bottom-up tree transducers is decidable.

Theorem 4.19 ([Gaz06a]) *It is decidable if a bottom-up tree transducer is shape preserving.*

By Theorem 3.64, for every shape preserving top-down tree transducer, an equivalent relabeling can be constructed. By Lemmas 3.36 and 4.5, for a given shape preserving bottom-up tree transducer M such that τ_M is infinite, an equivalent relabeling can be constructed. Since it is not difficult to see that if M is a shape preserving bottom-up tree transducer such that τ_M is finite, then a relabeling $\bar{M}$ can be given such that M and $\bar{M}$ are equivalent, we can conclude that for every shape preserving tree transducer, an equivalent relabeling can be constructed.

It was shown in [AB93] that the equivalence problem for relabeling tree transducers is decidable. Now using this result and that we can effectively give a relabeling equivalent to a shape preserving tree transducer, we have the following corollary.

Corollary 4.20 The equivalence problem for shape preserving tree transducers is decidable.

Conclusions

In the Thesis we have characterized shape preserving tree transducers by relabelings. We can use this result to generalize known results which concern relabelings as follows.

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 $SHAPE$.

At the end we mention a connection between the topic of the Thesis and another field of the computer science. In the Thesis we used relabelings to characterize shape preserving tree transducers. Recently, relabelings were also used in a certain type of model checking methods, called regular tree model checking [AJMd02]. In that work the states of a system are represented by trees over a ranked alphabet and transition relations between the states are represented by relabeling tree transducers.

References

[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.

[AJMd02] Parosh Aziz Abdulla, Bengt Jonsson, Pritha Mahata, and Julien d'Orso. Regular tree model checking. In *CAV*, pages 555–568, 2002.

[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.

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

[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.

[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.

[FV92] Z. Fülöp and S. Vágvölgyi. 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.

[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.

[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.

[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.

[Rou70] W.C. Rounds. Mappings and grammars on trees. *Math. Systems Theory*, 4:257–287, 1970.

[Tha70] J.W. Thatcher. Generalized^2 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.