Minimization of Deterministic Bottom-up Tree Transducers

Sylvia Friese[1], Helmut Seidl[1], and Sebastian Maneth[2]

[1] Technische Universität München, Garching, Germany
[2] NICTA and University of New South Wales, Sydney, Australia

Abstract. We show that for every deterministic bottom-up tree transducer, a unique equivalent transducer can be constructed which is minimal. The construction is based on a sequence of normalizing transformations which, among others, guarantee that non-trivial output is produced as early as possible. For a deterministic bottom-up transducer where every state produces either none or infinitely many outputs, the minimal transducer can be constructed in polynomial time.

Key words: Bottom-up Tree Transducers, Minimization, Normal form

1 Introduction

Top-down and bottom-up tree transducers were invented in the 1970s by Rounds and Thatcher [21, 23], and Thatcher [24], respectively. Their expressive powers are incomparable, both for nondeterministic and deterministic transducers [4], similar to the fact that left-to-right and right-to-left string transducers are incomparable (see Section IV.2 in [2]). In 1980 it was shown that equivalence for deterministic transducers is decidable both in the top-down [11] and bottom-up case [25]. Later, a polynomial-time algorithm for single-valued bottom-up transducers has been provided [22]. Recently, it was shown that for total deterministic top-down tree transducers, equivalence can be decided in polynomial time [10]. The proof relies on a new canonical normal form for such transducers, called the *earliest* normal form (inspired by the earliest string transducers of Mohri [20]). The question arises whether deterministic bottom-up tree transducers (btt's) also allow for a similar canonical normal. In this paper we give an affirmative answer to this question.

We show that for every btt there is a unique equivalent bottom-up transducer in normal form. The main idea is to unite states which are equivalent with respect to their behavior on contexts. There are several obstacles for this basic approach. Finite sets of output trees for a given state could be assembled in several different ways. Even if infinitely many outputs may occur, still bounded parts of it could be produced earlier or later. Due to the tree structure of outputs, the output for contexts of states could agree only for one particular pair of output subtrees. In order to remove these obstacles, we present a sequence of normal forms of increasing strength. Generating the unique normal form for a given btt therefore proceeds in four steps: (1) first, we make the transducer *proper*, i.e., we remove all output from states which only produce finitely many different outputs. The output for such states is postponed until a state with infinitely many different outputs or the final function at the root of the input tree. This is similar to the *proper normal form* of [1, 8] (which removes states that produce finitely many outputs, using regular look-ahead). (2) We make the transducer *earliest*, i.e., every state with infinitely many outputs produces output as early as possible during the translation. (3) We remove pairwise *ground context unifiers*. It is only in step (4) that we can apply minimization in the usual way by merging states that are isomorphic. Steps (2)–(4) can be done in polynomial time, i.e., given a proper transducer, its unique minimal transducer is constructed in polynomial time. Constructing a proper transducer (Step 1) may take double-exponential time in the worst case.

Besides equivalence checking, there are many more applications of a canonical normal form. For instance, it allows for a Myhill-Nerode style theorem, which, in turn can be used to build a Gold-style learning algorithm; see [19] where both were done for deterministic top-down tree transducers (ttt's). As another example, the normal form can be used to decide certain (semantic) subclasses of btt's; e.g., we can decide whether a given btt is equivalent to a relabeling, using the normal form. This provides an alternative proof of [16], for the deterministic case.

Related Work. A valid generalization of both, btt's and ttt's is the deterministic top-down tree transducer with regular look-ahead [5]. Even though the equivalence problem for ttt's with regular look-ahead is easily reduced to the one for ttt's [10], it is an intriguing open problem whether ttt's with regular look-ahead have a canonical normal form. Another related model of transformation is the attribute grammar [17], seen as a tree transducer [13, 15]. For attributed tree transducers, decidability of equivalence is still an open problem, but, for the special subclass of "nonnested, separated" attribute grammars (those which can be evaluated in one strict top-down phase followed by one strict bottom-up phase) equivalence is known to be decidable [3]. This class strictly includes ttt's (but not btt's [14]).

There are several other interesting incomparable classes of tree translations for which equivalence is known to be decidable, but no normal form (and no complexity) is known. For instance, MSO-definable tree translations [9]. This class coincides with single-use restricted attribute grammars or macro tree transducers with look-ahead [7]. Is there a canonical normal form for such transducers? Another interesting generalization are tree-to-*string* transducers. It is a long standing open problem [6] whether or not deterministic top-down tree-to-string transducers (ttst's) have decidable equivalence. Recently, for the subcase of *non-copying* ttst's, a unique normal form similar to the earliest normal form was presented [18]. Can their result be extended to the finite-copying case? Another recent result states that functional visibly pushdown transducers have decidable equivalence [12]. This class is closely related to non-copying ttst's. It raises the question whether our normal form for btt's can be extended to functional (but nondeterministic) bottom-up tree transducers.

2 Preliminaries

Bottom-up tree transducers work on ranked trees. In a ranked tree, the number of children of a node is determined by the *rank* of the symbol at that node. A ranked alphabet Σ consists of finitely many symbols. Each symbol $a \in \Sigma$ is equipped with a rank in

$\{0, 1, \dots\}$, where rank 0 indicates that a is the potential label of a leaf. We assume that a ranked alphabet contains at least one symbol of rank 0. The set $\mathcal{T}_\Sigma$ of ranked *trees* over Σ is the set of strings defined by the EBNF with rules $t ::= a(\underbrace{t, \dots, t}_{k \text{ times}})$ for all $k \geq 0$ and $a \in \Sigma$ of rank k. We also write a for the tree $a()$. Note that, since there is at least one symbol of rank 0, $\mathcal{T}_\Sigma \neq \emptyset$. We use the words *tree* and *term* interchangeably.

We consider trees possibly containing a dedicated variable $y \notin \Sigma$ (of rank 0). Let $\mathcal{T}_\Sigma(y)$ denote this set. On $\mathcal{T}_\Sigma(y)$, we define a binary operation "$\cdot$" by: $t_1 \cdot t_2 = t_1[t_2/y]$, i.e., the substitution by t_2 of every occurrence of the variable y in t_1. Note that the result is a ground tree, i.e., does not contain y, iff either $t_1 \in \mathcal{T}_\Sigma$ or $t_2 \in \mathcal{T}_\Sigma$. Moreover, the operation "$\cdot$" is associative with neutral element y. Therefore, the set $\mathcal{T}_\Sigma(y)$ together with the operation "$\cdot$" and y forms a monoid. Let $\hat{\mathcal{T}}_\Sigma(y)$ denote the sub-monoid consisting of all trees which contain at least one occurrence of y. Then $\mathcal{T}_\Sigma(y) = \hat{\mathcal{T}}_\Sigma(y) \cup \mathcal{T}_\Sigma$.

Proposition 1. *[6]*

1. *Let $s, s', t_1, t_2, t'_1, t'_2 \in \mathcal{T}_\Sigma(y)$ with $t_1 \neq t_2$ and $t'_1 \neq t'_2$. Assume that the two equalities $s \cdot t_1 = s' \cdot t'_1$ and $s \cdot t_2 = s' \cdot t'_2$ hold. Then one of the following two assertions are true:*
   (a) $s, s' \in \mathcal{T}_\Sigma$ and $s = s'$; or
   (b) Both trees s and s' contain an occurrence of y, i.e., are from $\hat{\mathcal{T}}_\Sigma(y)$, and $s \cdot u = s'$ or $s = s' \cdot u$ for some $u \in \hat{\mathcal{T}}_\Sigma(y)$.
2. *The sub-monoid $\hat{\mathcal{T}}_\Sigma(y)$ is free.*

Consider the set $\hat{\mathcal{T}}_\Sigma(y)_\perp = \hat{\mathcal{T}}_\Sigma(y) \cup \{\perp\}$ of all trees containing at least one occurrence of the variable y enhanced with an extra bottom element $\perp$ (not in $\Sigma \cup \{y\}$). On this set, we define a partial ordering by $\perp \sqsubseteq t$ for all t, and $t_1 \sqsubseteq t_2$ for $t_1, t_2 \in \hat{\mathcal{T}}_\Sigma(y)$ iff $t_1 = t' \cdot t_2$ for a suitable $t' \in \hat{\mathcal{T}}_\Sigma(y)$. The greatest element with respect to this ordering is y while the least element is given by $\perp$. With respect to this ordering, we observe:

1. Every $t \in \hat{\mathcal{T}}_\Sigma(y)$ has finitely many upper bounds.
2. For every $t_1, t_2 \in \hat{\mathcal{T}}_\Sigma(y)_\perp$, there exists a least upper bound $t_1 \sqcup t_2$ in $\hat{\mathcal{T}}_\Sigma(y)_\perp$.

Since $\hat{\mathcal{T}}_\Sigma(y)_\perp$ also has a least element, namely $\perp$, we conclude that $\hat{\mathcal{T}}_\Sigma(y)_\perp$ is a *complete* lattice satisfying the ascending chain condition, i.e., every set $X \subseteq \hat{\mathcal{T}}_\Sigma(y)_\perp$ has a least upper bound $t = \bigsqcup X$, and there are no infinite strictly ascending sequences $\perp \sqsubseteq t_1 \sqsubseteq t_2 \sqsubseteq \dots$. We call a tree $t \in \hat{\mathcal{T}}_\Sigma(y)$ *irreducible* if $t \neq y$ and $t \sqsubseteq t'$ only holds for $t' \in \{y, t\}$.

Let $\top \notin \Sigma \cup \{y\}$ be a new symbol. Assume that $c_1, c_2 \in \mathcal{T}_\Sigma(y)$ are trees, and that there are trees $s_1, s_2 \in \mathcal{T}_\Sigma(y) \cup \{\top\}$ such that $c_1 \cdot s_1 = c_2 \cdot s_2$. Note, that $c_i \cdot s_i = c_i$, if $c_i \in \mathcal{T}_\Sigma$. We call c_1, c_2 *unifiable* and $\langle s_1, s_2 \rangle$ a unifier of c_1, c_2.

We consider the set $\mathbb{D} = (\{y\} \times \hat{\mathcal{T}}_\Sigma(y)) \cup (\hat{\mathcal{T}}_\Sigma(y) \times \{y\}) \cup (\mathcal{T}_\Sigma \cup \{\top\})^2 \cup \{\perp\}$ of candidate unifiers. The set $\mathbb{D}$ forms a complete lattice w.r.t. the ordering $\leq$ defined by

- $\perp \leq d \leq \langle \top, \top \rangle$ for all $d \in \mathbb{D}$,
- $(d_1, d_2) \leq (d'_1, d'_2)$ if $d_i = d'_i \cdot s$ for all $i \in \{1, 2\}$ for some tree $s \in \mathcal{T}_\Sigma \cup \{y\}$, and
- $(d_1, d_2) \leq (d_1, \top)$ and $(d_1, d_2) \leq (\top, d_2)$ if $d_1, d_2 \in \mathcal{T}_\Sigma$.

The *most general unifier* $\mathsf{mgu}(c_1, c_2) \in \mathbb{D}$ for trees $c_1, c_2 \in \mathcal{T}_\Sigma(y)$ is the greatest unifier of c_1, c_2 w.r.t. the ordering $\leq$. It is $\bot$, if c_1, c_2 are not unifiable. Furthermore, for a set of pairs $C \subseteq \mathcal{T}_\Sigma(y)^2$ the most general unifier $\mathsf{mgu}(C)$ is the least upper bound of the unifiers of pairs in C, i.e., $\mathsf{mgu}(C) = \bigvee\{\mathsf{mgu}(c_1, c_2) \mid (c_1, c_2) \in C\}$.

For $k \in \{0, 1, \ldots\}$ we denote the set $\{x_1, \ldots, x_k\}$ of k distinct variables by $\mathcal{X}_k$. We consider trees with variables at leaves, i.e., trees in $\mathcal{T}_{\Sigma \cup \mathcal{X}_k}$ where each variable x_i has rank 0. Let $z \in \mathcal{T}_{\Sigma \cup \mathcal{X}_k}$ be such a tree. We abbreviate by $z[z_1, \ldots, z_k]$ the substitution $z[z_1/x_1, \ldots, z_k/x_k]$ of trees z_i for the variables x_i ($i = 1, \ldots, k$) in the tree z.

2.1 Bottom-Up Tree Transducers

A *deterministic bottom-up tree transducer* (btt for short) is a tuple $T = (Q, \Sigma, \Delta, R, F)$, where

- Q is a finite set of states,
- Σ and Δ are ranked input and output alphabets, respectively, disjoint with Q,
- R is the (possibly partial) transition function, and
- $F : Q \to \mathcal{T}_\Delta(y)$ is a partial function mapping states to final outputs.

For every input symbol $a \in \Sigma$ of rank k and sequence $q_1, \ldots, q_k$ of states, the transition function R contains at most one transition, which is denoted by $a(q_1, \ldots, q_k) \to q(z)$ where $q \in Q$ and $z \in \mathcal{T}_{\Delta \cup \mathcal{X}_k}$.

For every input symbol $a \in \Sigma$ of rank k and sequence of states $q_1 \ldots q_k$ of Q, let $R(a, q_1 \ldots q_k)$ be the right-hand side of the transition for a and $q_1 \ldots q_k$, if it is defined, and let $R(a, q_1 \ldots q_k)$ be undefined otherwise.

The transducer is *total* if $R(a, q_1 \ldots q_k)$ is defined for all $k \geq 0$, $a \in \Sigma$ of rank k, and sequences of states $q_1 \ldots q_k$. The *size* of T, denoted by $|T|$, is the sum of sizes (= number of symbols) of its final outputs and of the left-hand sides and right-hand sides of its transitions.

Assume that $t \in \mathcal{T}_\Sigma(y)$ and $q \in Q$. The *result* $[\![t]\!]_q^T$ of a computation of T on input t when starting in state q at variable leaves y is defined by induction on the structure of t:

$$\begin{aligned} &[\![y]\!]_q^T &&= q(y) \\ &[\![a(s_1, \ldots, s_k)]\!]_q^T &&= q'(z[z_1, \ldots, z_k]) \\ & && \quad \text{if } \forall i \; [\![s_i]\!]_q^T = q_i(z_i) \text{ and } R(a, q_1 \ldots q_k) = q'(z) \; . \end{aligned}$$

If $[\![t]\!]_q^T = q'(z')$, then z' is called the *output* produced for t. Note that the function $[\![\,.\,]\!]_q^T$ may not be defined for all trees t. The superscript ^{T} can be omitted if T is clear from the context. If $t \in \mathcal{T}_\Sigma$ we also omit the subscript q, i.e., we write $[\![t]\!]^T$ for $[\![t]\!]_q^T$.

The *image* $\tau_q^T(t)$ of the tree t is then defined by $\tau_q^T(t) = z' \cdot z$ iff $[\![t]\!]_q^T = q'(z)$ for some state q' with $F(q') = z'$. We omit the subscript q if the tree t does not contain the variable y.

We say that two btt's T and T' are *equivalent* when they describe the same transformation, i.e., for all $t \in \mathcal{T}_\Sigma$, $\tau^T(t)$ is defined iff $\tau^{T'}(t)$ is defined and are equal.

We also use the following notation. The *language* $\mathcal{L}^T(q)$ of a state q is the set of all ground input trees by which q is reached, i.e., $\mathcal{L}^T(q) = \{t \mid \exists s \in \mathcal{T}_\Delta : [\![t]\!]^T = q(s)\}$. A *context* c is a tree $c \in \hat{\mathcal{T}}_\Sigma(y)$ which contains exactly one occurrence of y. Let $\mathcal{C}_\Sigma$ be the set of all contexts. A tree $c \in \mathcal{C}_\Sigma$ is a context of a state q, if $\tau_q^T(c)$ is defined. Let $\mathcal{C}^T(q)$ denote the set of all contexts of state q. The *length* of a context c is the length of the path from the root to y. If context c has length n, then there are irreducible trees $c_1, \ldots, c_n$ such that $c = c_1 \cdot c_2 \cdot \ldots \cdot c_n$.

3 Trim Transducers

Transducers may contain useless transitions or states and we want to get rid of these while preserving the described transformation. A state q of a btt is *reachable*, if the language $\mathcal{L}^T(q)$ is non-empty. A state q is *meaningful*, if q has at least one context, i.e., $\mathcal{C}^T(q)$ is non-empty. Furthermore, the output at state q is *potentially useful*, if there is a context c of q such that the image $\tau_q^T(c)$ contains the variable y. Otherwise, the output at q is called *useless*. A bottom-up tree transducer T is called *trim* if T has the following properties: (1) every state is reachable, (2) every state is meaningful, (3) if the output at a state q is useless, then for each transition $a(q_1, \ldots, q_k) \rightarrow q(z)$ leading into state q, $z = *$. In this definition, $*$ is a special output symbol which does not occur in any image produced by T. It is well-known that each btt is equivalent to a trim btt.

Proposition 2. *For every bottom-up tree transducer T a bottom-up tree transducer T' can be constructed in polynomial time with the following properties:*

1. *$T'$is equivalent to T;*
2. *$|T'| \leq |T|$;*
3. *T' is trim.*

In the remainder of the paper we consider trim transducers only. For a trim transducer T with set Q of states, we denote by Q_* the set of states with useless output, i.e., for which the output is always $*$.

4 Proper Transducers

For a given trim transducer, there is not necessarily a unique minimal equivalent btt. Finite outputs of subtrees may be distributed over different states.

Example 3. Assume that $\Sigma = \{A, C, H, K, L\}$, $\Delta = \{a, b, h, l, *\}$, $Q_1 = \{q_0, q_1, q_2\}$, and $Q_2 = \{q_0', q_1', q_2'\}$. Consider the following transducers:

$$
\begin{array}{lll|lll}
\multicolumn{3}{l|}{T_1 = (Q_1, \Sigma, \Delta, R_1, F_1) \text{ with}} & \multicolumn{3}{l}{T_2 = (Q_2, \Sigma, \Delta, R_2, F_2) \text{ with}} \\
A(q_1) \rightarrow q_0(x_1) & & H \rightarrow q_1(b) & A(q_1') \rightarrow q_0'(b) & & H \rightarrow q_1'(h) \\
A(q_2) \rightarrow q_0(b) & & K \rightarrow q_1(a) & A(q_2') \rightarrow q_0'(a) & & K \rightarrow q_2'(*) \\
C(q_1) \rightarrow q_0(h) & & L \rightarrow q_2(*) & C(q_1') \rightarrow q_0'(x_1) & & L \rightarrow q_1'(l) \\
C(q_2) \rightarrow q_0(l) & & F_1(q_0) = y & C(q_2') \rightarrow q_0'(h) & & F_2(q_0') = y
\end{array}
$$

The transducer T_1 to the left produces different outputs for H and K while leading to the same state, and produces no output for L while leading into a different state. The transducer T_2 to the right, on the other hand, produces different outputs for H and L

leading into the same state and produces no output for K while leading into a different state. Both transducers are trim and describe the same transformation τ:

$$A(H) \to b \quad A(L) \to b \quad A(K) \to a \quad C(H) \to h \quad C(L) \to l \quad C(K) \to h$$

It is not clear how a unique normal form for τ with less than three states could look like. □

Let T denote a trim transducer with set of states Q. A state $q \in Q$ is called *essential* if the set of results $\{[\![t]\!]^T \mid t \in \mathcal{L}^T(q)\}$ for input trees reaching q is infinite. Otherwise, q is called *inessential*. Note that all states of the transducers in Example 3 are inessential.

A *proper* transducer postpones outputs at inessential states. The trim transducer T is called *proper* if every inessential state does not produce any output, i.e., is in Q_*. For every trim transducer there exists an equivalent proper transducer:

Proposition 4. *[1, 8] For every trim btt T a btt T' can be constructed with the following properties:*

1. *T' is equivalent to T;*
2. *T' is proper.*
3. *$|T'| \leq \mathit{\Gamma} \cdot |T|$*

where $\mathit{\Gamma}$ is the sum of sizes of all outputs produced for inessential states of T.

In the worst case, an inessential state may have exponentially many outputs – even if the input alphabet has maximal rank 1. In case, that both input and output alphabets have symbols of ranks greater than 1, doubly exponentially many outputs of inessential states are possible.

Proof (of Proposition 4 - Sketch). The inessential states are determined by considering the *dependence graph* $G_T = (V, E)$ where V is the set of states of T, and $(q_i, q) \in E$ if there is a transition $a(q_1, \ldots, q_k) \to q(z)$ in T and x_i occurs in z. We split each inessential state q into new states $\langle q, z\rangle$ where $q(z)$ is a possible result for some input tree $t \in \mathcal{L}^T(q)$. If q occurs on a left-hand side of a transition as the state for the i-th argument where the state in the right-hand side is essential, a new transition is generated where q is replaced with $\langle q, z\rangle$ and the corresponding variable x_i is replaced with z. Also, the final function F' should be modified accordingly for inessential states. □

Example 5. Consider again the transducer T_1 (to the left) of Example 3. The dependence graph G_{T_1} is $(\{q_0, q_1, q_2\}, \{(q_1, q_0)\})$. We determine that all states are inessential. The equivalent proper btt $T_1' = (Q_1', \mathit{\Sigma}, \mathit{\Delta}, R_1', F_1')$ has the following set of states:

$$Q_1' = \{\langle q_0, a\rangle, \langle q_0, b\rangle, \langle q_0, h\rangle, \langle q_0, l\rangle, \langle q_1, a\rangle, \langle q_1, b\rangle, \langle q_2, *\rangle\} .$$

Since every state of Q_1' is inessential, the output is postponed to the final function. Whereas, the right-hand sides of the transitions R_1' are of the form $\langle q, z\rangle\,(*)$, e.g.,

$$H \to \langle q_1, b\rangle\,(*) \quad A(\langle q_1, b\rangle) \to \langle q_0, b\rangle\,(*) \quad C(\langle q_1, b\rangle) \to \langle q_0, h\rangle\,(*) .$$

For the final function, we get

$$F_1'(\langle q_0, b\rangle) = b \qquad F_1'(\langle q_0, a\rangle) = a \qquad F_1'(\langle q_0, h\rangle) = h \qquad F_1'(\langle q_0, l\rangle) = l .$$

If we construct T_2' for the transducer T_2 to the right of Example 3, we get an isomorphic transducer. Both transducers are proper and realize the transformation τ^{T_1}. □

5 Earliest Transducers

Assume that we are given a proper btt T. We now want this transducer to produce the output at essential states as *early* as possible. Thereto, we compute the *greatest common suffix* of all non-ground images of contexts for a state q and produce it at q directly.

For an essential state q, let $\mathcal{D}(q)$ denote the set of images $z \in \hat{\mathcal{T}}_\Delta(y)$ produced for contexts of q. Thus, every tree in $\mathcal{D}(q)$ contains an occurrence of the variable y. The *greatest common suffix* of all trees in $\mathcal{D}(q)$ is denoted by $\mathsf{gcs}(q)$, i.e.,

$$\mathsf{gcs}(q) = \bigsqcup \mathcal{D}(q)$$

with respect to the order $\sqsubseteq$ on $\hat{\mathcal{T}}_\Delta(y)$ from Section 2.

Proposition 6. *For a proper btt T, the trees $\mathsf{gcs}(q)$ for all essential states q of T can be computed in polynomial time.*

Proof. Assume that $z \in \mathcal{T}_\Delta(\mathcal{X}_k)$ and that x_i occurs in z. Then $\mathsf{suff}_i(z)$ denotes the largest subtree z_i of $z[y/x_i]$ with the following properties:

- y is the only variable occurring in z_i, i.e., $z_i \in \hat{\mathcal{T}}_\Delta(y)$;
- $z[y/x_i] = z' \cdot z_i$ for some z', i.e., $z' \in \hat{\mathcal{T}}_{\Delta \cup \mathcal{X}_k \setminus \{x_i\}}(y)$.

Then the trees $\mathsf{gcs}(q)$ are the least solution of the inequations

$$\begin{array}{ll} \mathsf{gcs}(q_i) \sqsupseteq \mathsf{suff}_i(\mathsf{gcs}(q) \cdot z), & a(q_1, \ldots, q_k) \to q(z) \in R \text{ and } x_i \text{ occurs in } z, \\ \mathsf{gcs}(q) \;\sqsupseteq z, & F(q) = z \text{ and } y \text{ occurs in } z. \end{array}$$

Since T is proper, this system contains inequations only for essential states q. Since the right-hand sides are monotonic, the system has a unique least solution. Since the complete lattice $\hat{\mathcal{T}}_\Delta(y)_\perp$ satisfies the ascending chain condition, this least solution can effectively be computed. Using a standard worklist algorithm, it can be shown that each inequation is evaluated at most $\mathcal{O}(|T|)$ times. If we represent elements from $\hat{\mathcal{T}}_\Delta(y)$ as sequences of *irreducible* trees, then each right-hand side also can be evaluated in polynomial time. This proves the complexity bound stated in the proposition. □

A proper bottom-up tree transducer T is called *earliest* if the greatest common suffix of every essential state q equals y.

Theorem 7. *For each proper tree transducer T, a tree transducer T' can be constructed in polynomial time with the following properties:*

- *T' is equivalent to T;*
- *T' is earliest.*

Proof (Sketch). Let T be a proper transducer. According to Proposition 6, we can compute the greatest common suffix $\mathsf{gcs}(q)$ for every essential state q of T. The corresponding earliest transducer T' has the same set of states as T as well as the same input and output alphabets, but only differs in the transition function and the final function.

For a right-hand side $q(z)$ of a transition in T, we construct the output of the corresponding transition in T' in two steps. First, we add to z the greatest common suffix corresponding to q, i.e., we define $\bar{z} = \mathsf{gcs}(q) \cdot z$. Then we remove from $\bar{z}$ the greatest common suffices of all states corresponding of all variables occurring in $\bar{z}$ (and z). □

Example 8. Assume that $\Sigma = \{A, B, C, E\}$ and $\Delta = \{d, e\}$. Consider the proper btt $T = (Q, \Sigma, \Delta, R, F)$ with set of (essential) states $Q = \{q_1, q_2\}$ where the final function is $F = \{q_1 \mapsto d(d(y, e), d(y, e))\}$ and the transition function R is given by:

$$\begin{array}{rl} A(q_1, q_2) \to q_1(d(x_2, d(x_1, e))) & \quad E \to q_1(e) \\ B(q_2) \to q_2(d(x_1, d(d(e, e), e))) & \quad C \to q_2(e) \end{array}$$

To compute the greatest common suffices, we consider the following inequations:

$$\begin{array}{lll} \mathsf{gcs}(q_1) & \sqsupseteq \mathsf{suff}_1(\mathsf{gcs}(q_1) \cdot d(x_2, d(x_1, e))) & = d(y, e) \\ \mathsf{gcs}(q_2) & \sqsupseteq \mathsf{suff}_2(\mathsf{gcs}(q_1) \cdot d(x_2, d(x_1, e))) & = y \\ \mathsf{gcs}(q_2) & \sqsupseteq \mathsf{suff}_1(\mathsf{gcs}(q_2) \cdot d(x_1, d(d(e, e), e))) & = \mathsf{gcs}(q_2) \cdot d(y, d(d(e, e), e)) \\ \mathsf{gcs}(q_1) & \sqsupseteq F(q_1) & = d(d(y, e), d(y, e)) \end{array}$$

For q_2, we obtain $\mathsf{gcs}(q_2) = y$. Moreover since $d(y, e) \sqcup d(d(y, e), d(y, e)) = d(y, e)$, we have $\mathsf{gcs}(q_1) = d(y, e)$. The final function of the earliest btt for T' thus is given by $F' = \{q_1 \mapsto d(y, y)\}$. In order to construct the new transition function, first consider the right-hand side for $A(q_1, q_2)$ in R' where $R(A(q_1, q_2)) = q_1(d(x_2, d(x_1, e)))$. In the first step, we construct

$$\bar{z} = \mathsf{gcs}(q_1) \cdot d(x_2, d(x_1, e)) = d(y, e) \cdot d(x_2, d(x_1, e)) = d(d(x_2, d(x_1, e)), e) \ .$$

From this tree, we remove the suffices for q_1 and q_2 at the variables x_1 and x_2, respectively. This results in the tree $u = d(d(x_2, x_1), e)$. Therefore, we obtain the transition

$$A(q_1, q_2) \to q_1(d(d(x_2, x_1), e)) \ .$$

Analogously, we obtain the transitions

$$E \to q_1(d(e, e)) \qquad B(q_2) \to q_2(d(x_1, d(d(e, e), e))) \qquad C \to q_2(e) \ .$$

6 Unified Transducers

For an earliest btt, contexts of states may disagree except for a pair of output trees.

Example 9. Assume that $\Sigma = \{A, \dots, E, G\}$ and $\Delta = \{b, d, e, f, g, \perp\}$. Consider the earliest btt $T = (Q, \Sigma, \Delta, R, F)$ with $Q = \{q_0, q_1, q_1', q_2, q_2', q_3\}$ and R, F given by:

$$\begin{array}{rll} A \to q_0(b) & B(q_0) \to q_0(e(x_1)) & F(q_1) = f(y, b) \\ C(q_0) \to q_1(x_1) & D(q_0) \to q_1'(x_1) & F(q_1') = f(e(y), y) \\ E(q_0) \to q_2(x_1) & G(q_0) \to q_2'(x_1) & F(q_2) = y \\ C(q_2) \to q_1(e(b)) & C(q_2') \to q_1'(b) & F(q_2') = y \\ D(q_1) \to q_3(d(g, x_1)) & D(q_1') \to q_3(d(g, e(x_1))) & F(q_3) = y \end{array}$$

For each context c of q_2, i.e., $c \in \{C(y), D(C(y))\}$, the two states q_2 and q_2' induce the same image: $\tau^T_{q_2}(c) = \tau^T_{q_2'}(c)$. But unfortunately, the successor states q_1 and q_1' do not have this property. Both states are essential and have the same contexts. The images of the context $D(y)$, $\tau^T_{q_1}(D(y)) = d(g, y)$ and $\tau^T_{q_1'}(D(y)) = d(g, e(y))$, differ only in the suffix $e(y)$. The images of the context y are $\tau^T_{q_1}(y) = f(y, b)$ and $\tau^T_{q_1'}(y) = f(e(y), y)$. If y is substituted by b in the image at q_1' and $e(b)$ at q_1, they become equal. Thus, for each context c of q_1, we get $\tau^T_{q_1}(c) \cdot e(b) = \tau^T_{q_1'}(c) \cdot b$. □

Assume that $T = (Q, \Sigma, \Delta, R, F)$ is an earliest btt and that $q_1, q_2 \in Q$ are states. Assume that q_1 and q_2 have the same contexts, i.e., $\mathcal{C}^T(q_1) = \mathcal{C}^T(q_2)$. Then, we define the *most general unifier* of q_1, q_2, $\mathsf{mgu}(q_1, q_2)$, as the most general unifier of the set $C = \{(\tau^T_{q_1}(c), \tau^T_{q_2}(c)) \mid c \in \mathcal{C}^T(q_1)\}$ of pairs of images of contexts of q_1 and q_2, i.e., $\mathsf{mgu}(q_1, q_2) = \mathsf{mgu}(C)$. Otherwise, we set $\mathsf{mgu}(q_1, q_2) = \bot$. We call q_1, q_2 *unifiable* if the most general unifier is not $\bot$.

The most general unifier $\mathsf{mgu}(q_1, q_2) = \langle s_1, s_2 \rangle$ for unifiable states q_1, q_2 has the following properties:

- If q_i is inessential, then for every context c of q_i, $\tau^T_{q_i}(c) \in \mathcal{T}_\Delta$. Therefore, $s_i = \top$.
- Moreover, s_1 contains y iff s_2 contains y. If both s_1 and s_2 contain y, the mgu must equal $\langle y, y \rangle$, otherwise T would not be earliest.

A ground term s is called *realizable* in a state q if s is contained in the set of outputs of q. Note that the ground terms s occurring in most general unifiers of states are, however, not necessarily realizable. The earliest btt T is called *unified earliest* if no ground term in most general unifiers of states of T is realizable. In the following, we show that for every earliest btt, a unified earliest btt can be constructed in polynomial time. For this construction, we require the following observation.

Theorem 10. *Assume that T is an earliest bottom-up tree transducer. Then all most general unifiers $\mathsf{mgu}(q_1, q_2)$ can be constructed in polynomial time.*

Assume now that we are given the most general unifier of an earliest btt T. Then we can construct a unified earliest transducer T' which is equivalent to T. We have:

Theorem 11. *For each earliest btt T, a btt T' can be constructed in polynomial time with the following properties:*

- *T' is equivalent to T;*
- *T' is unified earliest.*

Proof (Sketch). Let T be an earliest btt. We construct the unified earliest btt T'.

First, we introduce new states. Whenever an output s of an input t at state q is produced by T which will contribute to a ground unifier of q, then the computation on t is redirected to a new state $\langle q, s \rangle$ which memorizes s and does produce $*$ only. Instead, the output s is delayed to the images of the contexts. This implies that the new state $\langle q, s \rangle$ is inessential. Furthermore, for states q' used to evaluate subtrees of t whose outputs s' may contribute to s, further states $\langle q', s' \rangle$ should be introduced.

Some of the new states $\langle q, s \rangle$ now may be unreachable. The transducer T' therefore is defined as the trim transducer, obtained according to Proposition 2. □

Example 12. Consider again the transducer $T = (Q, \Sigma, \Delta, R, F)$ of Example 9. The most general unifiers are

$$\mathsf{mgu}(q_1, q_1') = \langle e(b), b \rangle\,, \qquad \mathsf{mgu}(q_2, q_2') = \langle \top, \top \rangle\,,$$

and $\mathsf{mgu}(q, q') = \bot$, otherwise. We get the set $S = \{e(b), b, \bot\}$ of subterms of terms occurring as ground unifiers of states or $\bot$. All states of $Q \times S$ are possible new states. Except from $\langle q_3, b \rangle$ and $\langle q_3, e(b) \rangle$ all are reachable.

Starting with left-hand side A, we get the new transition $A \rightarrow \langle q_0, b\rangle\,(*)$, because $b \in S$. Furthermore, for the transition $B(q_0) \rightarrow q_0(e(x_1))$ we get the transition $B(\langle q_0, b\rangle) \rightarrow \langle q_0, e(b)\rangle\,(*)$, because $e(x_1)[b/x_1] = e(b) \in S$. Now, consider the left-hand side $B(\langle q_0, e(b)\rangle)$. The potential output $e(x_1)[e(b)/x_1] = e(e(b))$ is not in S. Thus, the right-hand side should be $\langle q_0, \bot\rangle\,(e(e(b)))$. And for the left-hand side $B(\langle q_0, \bot\rangle)$, we get the transition $B(\langle q_0, \bot\rangle) \rightarrow \langle q_0, \bot\rangle\,(e(x_1))$. □

7 Minimal Transducers

Last, we merge equivalent states by preserving the properties of an unified earliest btt. Let $\sim'$ denote the smallest equivalence relation with the following properties:

- If $\mathsf{mgu}(q, q') = \langle y, y\rangle$ or $\mathsf{mgu}(q, q') = \langle \top, \top\rangle$ then $q \sim' q'$;
- Assume that $\mathsf{mgu}(q, q_1) = \langle \top, s_1\rangle$ for some ground term s_1. If for all q_2 with $\mathsf{mgu}(q, q_2) = \langle \top, s_2\rangle$ for some $s_2 \neq \top$, $\mathsf{mgu}(q_1, q_2) = \langle y, y\rangle$ holds then $q \sim' q_1$.

The relation $\sim$ is the greatest equivalence relation which is a refinement of $\sim'$ such that, $q_1 \sim q_2$ whenever for every symbol $a \in \Sigma$ of rank k, every $1 \leq \ i \leq \ k$, and all states $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_k \in Q$, the following holds. There is a transition $a(p_1, \ldots, p_{i-1}, q_1, p_{i+1}, \ldots, p_k) \rightarrow q'_1(z_1)$ in R iff there is a transition in R of the form $a(p_1, \ldots, p_{i-1}, q_2, p_{i+1}, \ldots, p_k) \rightarrow q'_2(z_2)$. If such two transitions exist then $q'_1 \sim q'_2$.

A unified earliest transducer $T = (Q, \Sigma, \Delta, R, F)$ is said to be *minimal* iff all distinct states $q_1, q_2 \in Q$ are not equivalent, i.e., $q_1 \not\sim q_2$.

Theorem 13. *For each unified earliest btt T a unified earliest btt T' can be constructed in polynomial time with the following properties:*

- *T' is equivalent to T;*
- *T' is minimal;*
- $|T'| \leq |T|$.

Proof. Let $T = (Q, \Sigma, \Delta, R, F)$ be an earliest btt. By fixpoint iteration, we compute the equivalence relation $\sim$ on Q. Now, we build a transducer T' with the equivalence classes of $\sim$ as states. Let $[q] = \{q' \mid q \sim q'\}$ the equivalence class of q. We call $[q]$ inessential, if all states in $[q]$ are inessential. Otherwise, it is called essential. For each class $[q]$ we mark a representative state $p_q \in Q$, which is essential iff $[q]$ is essential.

Formally, we get $T' = (Q', \Sigma, \Delta, R', F')$ with $Q' = \{[q] \mid q \in Q\}$. The function F' is given by $F'([q]) = F(p_q)$. And for R', assume that $q_1, \ldots, q_k \in Q$ are representatives of their equivalence classes and that $a(q_1, \ldots, q_k) \rightarrow q(z) \in R$.

- If q is essential, then $a([q_1], \ldots, [q_k]) \rightarrow [q](z) \in R'$.
- If p_q is inessential, then $a([q_1], \ldots, [q_k]) \rightarrow [q](z) \in R'$.
- Otherwise, if $\mathsf{mgu}(q, p_q) = \langle \top, s\rangle$, then $a([q_1], \ldots, [q_k]) \rightarrow [q](s) \in R'$.

By induction on the depth of input trees t and length of contexts c, we obtain:

- $\forall t \in \mathcal{T}_\Sigma$: $[\![t]\!]^{T'} = [q](z)$ iff $\exists q' \in [q]$ with $[\![t]\!]^T = \begin{cases} q'(z) & \mathsf{mgu}(q', p_q) = \langle y, y\rangle \\ q'(z) & \mathsf{mgu}(q', p_q) = \langle \top, \top\rangle \\ q'(*) & \mathsf{mgu}(q', p_q) = \langle \top, z\rangle \end{cases}$

- $\forall c \in \hat{\mathcal{T}}_\Sigma(y) : \tau_{[q]}^{T'}(c) = z$ iff $\tau_{p_q}^{T}(c) = z$.

It follows that $\tau^T = \tau^{T'}$ and that T' is trim, proper, earliest, and unified. □

In the following, we show that the equivalent minimal btt for a given earliest unified btt is unique. Let $T_1 = (Q_1, \Sigma, \Delta, R_1, F_1)$ and $T_2 = (Q_2, \Sigma, \Delta, R_2, F_2)$ be two equivalent minimal btt's, i.e. $\tau^{T_1} = \tau^{T_2}$. We abbreviate the output of a tree $t \in \mathcal{T}_\Sigma$ in a transducer T_i as $out^{T_i}(t)$, i.e., it exists a state q in Q_i with $[\![t]\!]^{T_i} = q(out^{T_i}(t))$.

For each state $q \in Q_1$ a state $r_q \in Q_2$ is said to be *related* to q, if both are reached by at least one same input tree, i.e., $\exists t \in \mathcal{L}^{T_1}(q) \cap \mathcal{L}^{T_2}(r_q)$. If the transducers are trim, for each state $q \in Q_1$ exists at least one related state r_q: If q is reachable, there exists $t \in \mathcal{L}^{T_1}(q)$. Since q is meaningful, there should also exist a state $r_q \in Q_2$ with $t \in \mathcal{L}^{T_2}(r_q)$. We will show that there exists exactly one related state for each $q \in Q_1$. That will give us a mapping from T_1 to T_2.

Lemma 14. *Assume T_1 and T_2 are two minimal transducers which are equivalent. Then for each state q of T_1 there exists exactly one related state r_q in T_2. And the following holds:*

- *Every context c of q is a context of r_q and $\tau_q^{T_1}(c) = \tau_{r_q}^{T_2}(c)$;*
- *$\mathcal{L}^{T_1}(q) = \mathcal{L}^{T_2}(r_q)$ and for each input tree t holds $[\![t]\!]^{T_1} = q(z)$ iff $[\![t]\!]^{T_2} = r_q(z)$.*

Theorem 15. *The minimal transducer T for a transformation τ is unique.*

Proof (Sketch). Assume T_1 and T_2 are minimal transducers with $\tau^{T_1} = \tau^{T_2}$, and define a mapping $\varphi : Q_1 \to Q_2$ by $\varphi(q) = r_q$ where r_q is the related state of q. By the previous lemma, this mapping is well-defined and bijective. It remains to show that φ is an isomorphism w.r.t. the transition and final functions, i.e.,

1. $F_1(q)$ is defined iff $F_2(\varphi(q))$ is defined, and if they are defined, $F_1(q) = F_2(\varphi(q))$,
2. $a(q_1, \ldots, q_k) \to q_0(z_0) \in R_1 \Leftrightarrow a(\varphi(q_1), \ldots, \varphi(q_k)) \to \varphi(q_0)(z_0) \in R_2$. □

Summarizing, we obtain from Propositions 2, 4 and Theorems 7, 11, 13 and 15:

Theorem 16. *For each btt T an equivalent minimal transducer can be constructed which is unique up to renaming of states. If the btt T is already proper, the construction can be performed in polynomial time.* □

8 Conclusion

We have provided a normal form for deterministic bottom-up tree transducers which is unique up to renaming of states. In case that the btt is already proper, i.e., does only produce output at essential states, the construction can be performed in polynomial time — given that we represent right-hand sides compactly. Though similar in spirit as the corresponding construction for top-down deterministic transducers, the given construction for btt's is amazingly involved and relies on a long sequence of transformations of the original transducer to rule out anomalies in the behavior of the transducer.

It remains to future work to evaluate in how far our novel normal-form can be applied, e.g., in the context of learning tree-to-tree transformations.

References

1. A. V. Aho and J. D. Ullman. Translations on a context-free grammar. *Inform. and Control*, 19:439–475, 1971.
2. J. Berstel. *Transductions and Context-Free Languages*. Teubner, Stuttgart, 1979.
3. B. Courcelle and P. Franchi-Zannettacci. On the equivalence problem for attribute systems. *Inform. and Control*, 52:275–305, 1982.
4. J. Engelfriet. Bottom-up and top-down tree transformations — a comparison. *Math. Systems Theory*, 9:198–231, 1975.
5. J. Engelfriet. Top-down tree transducers with regular look-ahead. *Math. Systems Theory*, 10:289–303, 1977.
6. J. Engelfriet. Some open questions and recent results on tree transducers and tree languages. In R.V. Book, editor, *Formal language theory; perspectives and open problems*. Academic Press, New York, 1980.
7. J. Engelfriet and S. Maneth. Macro tree transducers, attribute grammars, and MSO definable tree translations. *Inform. and Comput.*, 154:34–91, 1999.
8. J. Engelfriet and S. Maneth. Macro tree translations of linear size increase are MSO definable. *SIAM J. Comput.*, 32:950–1006, 2003.
9. J. Engelfriet and S. Maneth. The equivalence problem for deterministic MSO tree transducers is decidable. *Inform. Proc. Letters*, 100:206–212, 2006.
10. J. Engelfriet, S. Maneth, and H. Seidl. Deciding equivalence of top-down XML transformations in polynomial time. *J. Comput. Syst. Sci.*, 75(5):271–286, 2009.
11. Z. Ésik. Decidability results concerning tree transducers I. *Acta Cybernetica*, 5:1–20, 1980.
12. E. Filiot, J.-F. Raskin, P.-A. Reynier, F. Servais, and J.-M. Talbot. Properties of visibly pushdown transducers. Submitted, 2010.
13. Z. Fülöp. On attributed tree transducers. *Acta Cybernetica*, 5:261–279, 1981.
14. Z. Fülöp and S. Vágvölgyi. Attributed tree transducers cannot induce all deterministic bottom-up tree transformations. *Inform. and Comput.*, 116:231–240, 1995.
15. Z. Fülöp and H. Vogler. *Syntax-Directed Semantics – Formal Models based on Tree Transducers*. EATCS Monographs in Theoretical Computer Science (W. Brauer, G. Rozenberg, A. Salomaa, eds.). Springer-Verlag, 1998.
16. Z. Gazdag. Decidability of the shape preserving property of bottom-up tree transducers. *Int. J. Found. Comput. Sci.*, 17(2):395–414, 2006.
17. D.E. Knuth. Semantics of context-free languages. *Math. Systems Theory*, 2:127–145, 1968. (Corrections in *Math. Systems Theory*, 5:95-96, 1971).
18. G. Laurence, A. Lemay, J. Niehren, S. Staworko, and M. Tommasi. Linear top-down tree-to-word transducers: Characterization and minimization. In *RTA*, 2010. To appear.
19. A. Lemay, S. Maneth, and J. Niehren. A learning algorithm for top-down XML transformations. In *PODS*, 2010. To appear.
20. M. Mohri. Minimization algorithms for sequential transducers. *Theor. Comput. Sci.*, 234:177–201, 2000.
21. W.C. Rounds. Mappings and grammars on trees. *Math. Systems Theory*, 4:257–287, 1970.
22. H. Seidl. Single-valuedness of tree transducers is decidable in polynomial time. *Theor. Comput. Sci.*, 106(1):135–181, 1992.
23. J.W. Thatcher. Generalized^2 sequential machine maps. *J. Comp. Syst. Sci.*, 4:339–367, 1970.
24. 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.
25. Z. Zachar. The solvability of the equivalence problem for deterministic frontier-to-root tree transducers. *Acta Cybernetica*, 4:167–177, 1980.