How to Remove the Look-Ahead of Top-Down Tree Transducers

How to Remove the Look-Ahead of Top-Down Tree Transducers

Joost Engelfriet[1], Sebastian Maneth[2], and Helmut Seidl[3]

[1] LIACS, Leiden University, The Netherlands
engelfri@liacs.nl
[2] School of Informatics, University of Edinburgh, United Kingdom
smaneth@inf.ed.ac.uk
[3] Institut für Informatik, Technische Universität München, Germany
seidl@in.tum.de

Abstract. For a top-down tree transducer with regular look-ahead we introduce the notion of difference bound, which is a number bounding the difference in output height for any two look-ahead states of the transducer. We present an algorithm that, for a given transducer with a known difference bound, decides whether it is equivalent to a transducer without regular look-ahead, and constructs such a transducer if the answer is positive. All transducers are total and deterministic.

1 Introduction

Many simple tree transformations can be modeled by top-down tree transducers, as recently used in XML database theory (e.g., [6,11,13,15]), in computational linguistics (e.g., [12,14]) and in picture generation [4]. A *top-down tree transducer* is a finite-state device that scans the input tree in a (parallel) top-down fashion, simultaneously producing the output tree. The more expressive (but also more complex) top-down tree transducer *with regular look-ahead* [5] consists of a top-down tree transducer and a finite-state bottom-up tree automaton, called the look-ahead automaton. At each input node, the transducer can inspect the look-ahead state (i.e., the state of the automaton) of each child of that node. Consider, e.g., a transducer M_{ex} of which the look-ahead automaton checks whether the input tree has a leaf labeled a; if so, M_{ex} outputs a, otherwise it outputs a copy of the input tree. Clearly, there is no transducer without look-ahead that realizes the same translation as M_{ex}. In general, is there a method to determine for a given top-down tree transducer *with look-ahead* (dtla), whether or not there is an equivalent top-down tree transducer *without look-ahead* (dtop)?

In this paper we provide a general method as discussed above, for total deterministic transducers. However, part of the method is not automatic: it depends on additional knowledge about the given transducer with look-ahead. For transducers with some restrictions concerning the power to copy and erase, that knowledge can also be obtained automatically.

The main notion on which our method is based, is that of a *difference tree* of a dtla M. Consider two trees obtained from one input tree by replacing one leaf

A.M. Shur and M.V. Volkov (Eds.): DLT 2014, LNCS 8633, pp. 103–115, 2014.

by two different look-ahead states of M. Compare now the two output trees of M on these input trees, where M treats the look-ahead state as representing an input subtree for which the look-ahead automaton arrives in that state at the root of the subtree.[1] Removing the largest common prefix of these two output trees (i.e., every node of which every ancestor has the same label in each of the two trees), we obtain a number of output subtrees that we call difference trees of M. Intuitively, the largest common prefix is the part of the output that does not depend on the two possible look-ahead states of the subtree, whereas a difference tree is a part of the output that can be produced because M knows that look-ahead state. Thus, the set $\mathsf{diff}(M)$ of all difference trees of M can be viewed as a measure of the impact of the look-ahead on the behaviour of M. E.g., $\mathsf{diff}(M_{\mathrm{ex}})$ is infinite: it consists of the one-node tree a and all trees of which no leaf is labeled a (with one leaf representing a subtree without a-labeled leaves).

The idea of our method is as follows. For any dtop an equivalent canonical dtop can be constructed [6]. Canonical means that each output node is produced as early as possible, and that different states of the transducer are inequivalent. We can generalize that result to dtlas. Thus, if there is a (canonical) dtop N equivalent to the (canonical) dtla M, then M is at least as early as N: at each moment of the translation, the output of N is a prefix of that of M. The output of N is the part of M's output that does not depend on the look-ahead state. Thus, when removing the output of N from that of M, the remaining trees are difference trees of M. Since N is able to simulate M, it has to store these difference trees in its states. Hence, $\mathsf{diff}(M)$ is finite. In fact, it turns out that the above description of N's behaviour completely determines N, and so N can be constructed from M and $\mathsf{diff}(M)$, and then tested for equivalence with M [9].

A natural number h is a *difference bound* for a dtla M if the following holds: if M has finitely many difference trees, then h is an upper bound on their height. Our first main result is that it is decidable for a given dtla M for which a difference bound is also given, whether M is equivalent to a dtop N, and if so, such a dtop N can be constructed. We do not know whether a difference bound can be computed for every dtla M, but the designer of M will usually be able to determine $\mathsf{diff}(M)$ and hence a difference bound for M. Our second main result is that a difference bound can be computed for dtlas that are linear and nonerasing (or even ultralinear and bounded erasing); the proof is too involved to be presented here. The full version of this paper can be found in [7].

Related Work. For deterministic string transducers it is decidable whether a given transducer with look-ahead is equivalent to a transducer without look-ahead, and if so, such a transducer can be constructed. This was proved in [3] (see also [2, Theorem IV.6.1]), for so-called subsequential functions. For macro tree transducers [8] and streaming tree transducers [1], regular look-ahead can always be removed. The same is true for nondeterministic visibly pushdown transducers [10]; for deterministic visibly pushdown transducers the addition of regular look-ahead increases their power, but the decidability of look-ahead removal for these transducers is not studied in [10].

[1] Since M is total and deterministic, the output trees exist and are unique, respectively.

2 Top-Down Tree Transducers and Difference Trees

We assume the reader to be familiar with top-down tree transducers working on ranked trees: the number of children of a tree node is determined by its label.

A *deterministic top-down tree transducer with regular look-ahead* (*dtla* for short) is a tuple $M = (Q, \Sigma, \Delta, R, A, P, \delta)$ where Q is a finite set of states of rank 1, Σ and Δ are the ranked input and output alphabets, and P is a finite nonempty set of look-ahead states. For every $p \in P$, $A(p)$ is a tree in $\mathcal{T}_\Delta(Q(\{x_0\}))$ called the p-axiom of M.[2] For every $q \in Q$, $a \in \Sigma$ of rank $k \geq 0$, and $p_1, \ldots, p_k \in P$, the set R contains at most one rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta$ where ζ is a tree in $\mathcal{T}_\Delta(Q(X_k))$ denoted by $\mathsf{rhs}(q, a, p_1, \ldots, p_k)$. Finally, δ is the transition function of the (total deterministic bottom-up) look-ahead automaton (P, δ), i.e., $\delta(a, p_1, \ldots, p_k) \in P$ for every $a \in \Sigma$ of rank $k \geq 0$ and $p_1, \ldots, p_k \in P$. The extension of δ to a mapping from $\mathcal{T}_\Sigma$ to P, also denoted by δ, is defined by $\delta(a(s_1, \ldots, s_k)) = \delta(a, \delta(s_1), \ldots, \delta(s_k))$ for $a \in \Sigma$ of rank $k \geq 0$ and $s_1, \ldots, s_k \in \mathcal{T}_\Sigma$. For $p \in P$ we define $[\![p]\!]_M = \{s \in \mathcal{T}_\Sigma \mid \delta(s) = p\}$. The dtla M *realizes* the partial function $[\![M]\!] : \mathcal{T}_\Sigma \to \mathcal{T}_\Delta$, called its *translation*, defined for $s \in \mathcal{T}_\Sigma$ by $[\![M]\!](s) = A(\delta(s))[q(x_0) \leftarrow [\![q]\!]_M(s) \mid q \in Q]$.[3] For $q \in Q$ the partial function $[\![q]\!]_M : \mathcal{T}_\Sigma \to \mathcal{T}_\Delta$ is defined for $s \in \mathcal{T}_\Sigma$ of the form $a(s_1, \ldots, s_k)$ by $[\![q]\!]_M(s) = \mathsf{rhs}(q, a, \delta(s_1), \ldots, \delta(s_k))[q'(x_i) \leftarrow [\![q']\!]_M(s_i) \mid q' \in Q, 1 \leq i \leq k]$. We write $M(s)$ for $[\![M]\!](s)$, and $q_M(s)$ for $[\![q]\!]_M(s)$. Two dtlas M_1 and M_2 are *equivalent* if they realize the same translation, i.e., $\Sigma_{M_1} = \Sigma_{M_2}$, $\Delta_{M_1} = \Delta_{M_2}$ and $[\![M_1]\!] = [\![M_2]\!]$.

Convention. We (can) assume that all states and look-ahead states of M are *reachable*: $p \in P$ is reachable if $[\![p]\!]_M \neq \emptyset$; $q \in Q$ is reachable if q occurs in an axiom, or in the right-hand side of a rule of which the left-hand side starts with a reachable state.

A dtla M is *total* if its translation $[\![M]\!]$ is a total function, i.e., its domain is $\mathcal{T}_\Sigma$. **From now on we only consider total dtlas**.

A *deterministic top-down tree transducer* (*dtop* for short) is a dtla M such that P is a singleton, i.e., $P = \{p\}$. For convenience, we drop (P, δ) from the tuple defining M, write a rule as $q(a(x_1, \ldots, x_k)) \to \zeta$ rather than $q(a(x_1 : p, \ldots, x_k : p)) \to \zeta$, identify A with the unique axiom $A(p)$, and denote ζ by $\mathsf{rhs}(q, a)$.

A dtla M is *proper* (a *dtpla* for short) if it is not a dtop, i.e., if $|P| \geq 2$.

[2] We use variables x_i with $i \in \mathbb{N}$, of rank 0. The set $\{x_0, x_1, x_2, \ldots\}$ is denoted X; for $k \in \mathbb{N}$, $X_k := \{x_1, \ldots, x_k\}$. For a set of trees $\mathcal{T}$, $Q(\mathcal{T})$ is the set of trees $q(t)$ with $q \in Q$ and $t \in \mathcal{T}$, and $\mathcal{T}_\Delta(\mathcal{T})$ is the smallest set of trees $\mathcal{T}'$ containing $\mathcal{T}$ such that $d(t_1, \ldots, t_k) \in \mathcal{T}'$ if $d \in \Delta$ and $t_1, \ldots, t_k \in \mathcal{T}'$ (where d has rank k). We denote $\mathcal{T}_\Delta(\emptyset)$ by $\mathcal{T}_\Delta$. For a tree $t \in \mathcal{T}_\Delta$, we denote by $V(t)$ the set of nodes of t, which are strings of positive natural numbers, i.e., $V(t) \subseteq \mathbb{N}_+^*$ with $\mathbb{N}_+ = \mathbb{N} - \{0\}$. The empty string ε is the root node and, for $i \in \mathbb{N}_+$, vi is the ith child of the node v. Every node $v \in V(t)$ has a label in Δ, denoted $\mathsf{lab}(t, v)$; the subtree of t rooted at v is denoted by t/v. For $d \in \Delta$, we define $V_d(t) = \{v \in V(t) \mid \mathsf{lab}(t, v) = d\}$.

[3] For sets of trees $\mathcal{S}, \mathcal{T}$, a tree $t \in \mathcal{T}$ and a partial function $\psi : \mathcal{S} \to \mathcal{T}$, we define $t[s \leftarrow \psi(s) \mid s \in \mathcal{S}]$ to be the result of replacing every subtree s of t by $\psi(s)$, for every $s \in \mathcal{S}$ (assuming that no tree in $\mathcal{S}$ has a proper subtree in $\mathcal{S}$).

Look-Ahead States in Input Trees. Let $M = (Q, \Sigma, \Delta, R, A, P, \delta)$ be a (total) dtla. To analyze the behaviour of M for different look-ahead states, we consider input trees $\bar{s}$ with occurrences of $p \in P$, viewed as input symbol of rank zero, representing an absent subtree s with $\delta(s) = p$. Intuitively, when M arrives in state q at a p-labeled leaf of $\bar{s}$, we let M output the new symbol $\langle q, p\rangle$ of rank zero, representing the absent output tree $q_M(s)$; thus, input trees $\bar{s} \in \mathcal{T}_\Sigma(P)$ are translated to output trees in $\mathcal{T}_\Delta(Q \times P)$. Formally, we extend M to a dtla $M^\circ = (Q, \Sigma^\circ, \Delta^\circ, R^\circ, A, P, \delta^\circ)$ where $\Sigma^\circ = \Sigma \cup P$, every element of P has rank zero, $\Delta^\circ = \Delta \cup (Q \times P)$, every element of $Q \times P$ has rank zero, $R^\circ = R \cup \{q(p) \to \langle q, p\rangle \mid q \in Q, p \in P, \exists s \in [\![p]\!]_M : q_M(s) \text{ is defined}\}$, and δ° is the extension of δ with $\delta^\circ(p) = p$ for every $p \in P$. For notational simplicity, we will denote $\delta^\circ(\bar{s})$, $M^\circ(\bar{s})$ and $q_{M^\circ}(\bar{s})$ by $\delta(\bar{s})$, $M(\bar{s})$ and $q_M(\bar{s})$, respectively, for every $\bar{s} \in \mathcal{T}_\Sigma(P)$. But note that $[\![p]\!]_M$, $[\![M]\!]$ and $[\![q]\!]_M$ keep their meaning.

A *Σ-context* is a tree in $\mathcal{T}_\Sigma(\{\bot\})$ that contains exactly one occurrence of $\bot$ (which is a new symbol of rank 0). The set of all Σ-contexts is denoted $\mathcal{C}_\Sigma$. For $C \in \mathcal{C}_\Sigma$ and a tree s, the tree $C[s]$ is obtained from the context C by replacing the unique occurrence of $\bot$ in C by s. We consider in particular trees $C[p]$ where $C \in \mathcal{C}_\Sigma$ and $p \in P$. Note that the tree $M(C[p])$ is in $\mathcal{T}_\Delta(Q \times \{p\})$.

Lemma 1. *Let M be a dtla. Let $C \in \mathcal{C}_\Sigma$, $s \in \mathcal{T}_\Sigma(P)$ and $p \in P$ such that $\delta(s) = p$. Then $\delta(C[s]) = \delta(C[p])$ and $M(C[s]) = M(C[p])\big[\langle q, p\rangle \leftarrow q_M(s) \mid q \in Q\big]$.*

Difference Trees and Difference Bounds. For a ranked alphabet Ω, an *Ω-pattern* (or just pattern) is a tree in $\mathcal{T}_\Omega(\{\bot\})$, where $\bot \notin \Omega$ has rank 0. Intuitively, an Ω-pattern is a prefix of a tree in $\mathcal{T}_\Omega$. If $t_0 \in \mathcal{T}_\Omega(\{\bot\})$ contains exactly k occurrences of $\bot$, and $t_1, \ldots, t_k \in \mathcal{T}_\Omega(\{\bot\})$, then the pattern $t = t_0[t_1, \ldots, t_k]$ is obtained from t_0 by replacing the ith occurrence of $\bot$ (in left-to-right order) by t_i. On the set $\mathcal{T}_\Omega(\{\bot\})$ we define a partial order: for patterns t and t', t' is a *prefix* of t, denoted $t' \sqsubseteq t$, if $t = t'[t_1, \ldots, t_k]$ for some patterns $t_1, \ldots, t_k$; equivalently, $V_b(t') \subseteq V_b(t)$ for every $b \in \Omega$. In [6] the inverse of $\sqsubseteq$ is used. Note that $\bot \sqsubseteq t$ for every pattern t. Every nonempty set Π of Ω-patterns has a greatest lower bound $\sqcap\Pi$ in $\mathcal{T}_\Omega(\{\bot\})$, called the *largest common prefix* of the patterns in Π; it is the unique pattern t' such that for every $v \in \mathbb{N}_+^*$ and $b \in \Omega$, $v \in V_b(t')$ if and only if (1) $v \in V_b(t)$ for every $t \in \Pi$ and (2) every proper ancestor of v is in $V(t')$. For instance, $\sqcap\{\sigma(\tau(a), b), \sigma(b, b)\} = \sigma(\tau(a), b) \sqcap \sigma(b, b) = \sigma(\bot, b)$.

We wish to decide whether the dtla M is equivalent to a dtop. Let C be a Σ-context and let $p, p' \in P$. As explained in the Introduction, we are interested in the difference between $M(C[p])$ and $M(C[p'])$, cf. Lemma 1. Intuitively, a dtop N that is equivalent to M does not know whether the subtree s of an input tree $C[s]$ has look-ahead state p or p', and hence, when reading the context C, it can output at most the largest common prefix $M(C[p]) \sqcap M(C[p'])$ of the output trees $M(C[p])$ and $M(C[p'])$. Let v be a node of $M(C[p]) \sqcap M(C[p'])$ with label $\bot$. Then we say that $M(C[p])/v$ is a *difference tree* of M (and hence, by symmetry, so is $M(C[p'])/v$). Thus, a difference tree is a part of the output that can be produced by M because it knows that s has look-ahead state p (or p'). Intuitively, to simulate M, the dtop N must store the difference trees in its state.

Hence, for N to exist, there should be finitely many difference trees (Corollary 1). We denote the set of all difference trees of M by $\mathsf{diff}(M)$, for varying C, p, p' and v. Thus we define $\mathsf{diff}(M) = \{M(C[p])/v \mid C \in \mathcal{C}_\Sigma,\, p \in P,\, \exists p' \in P : v \in V_\perp(M(C[p]) \sqcap M(C[p']))\}$ which is a subset of $\mathcal{T}_\Delta(Q \times P)$. We define the number $\mathsf{maxdiff}(M) \in \mathbb{N} \cup \{\infty\}$ to be the maximal height of all difference trees of M, i.e., $\mathsf{maxdiff}(M) = \sup\{\mathsf{ht}(t) \mid t \in \mathsf{diff}(M)\}$. Intuitively, $\mathsf{maxdiff}(M)$ gives a measure of how much M makes use of its look-ahead information. Obviously, $\mathsf{maxdiff}(M)$ is finite if and only if $\mathsf{diff}(M)$ is finite. A number $h(M) \in \mathbb{N}$ is a *difference bound* for M if either $\mathsf{diff}(M)$ is infinite or $\mathsf{maxdiff}(M) \leq h(M)$. Our first main result is that if a difference bound for M is known, then we can decide whether M is equivalent to a dtop, and if so, construct such a dtop from M (Theorem 2).

Example 1. Let $\Sigma = \Delta = \{\sigma^{(1)}, a^{(0)}, b^{(0)}\}$, the ranked alphabet $\{\sigma, a, b\}$ such that σ has rank 1 and a, b have rank 0. For $n \in \mathbb{N}$, the tree $\sigma(\sigma(\cdots\sigma(a)\cdots))$ with n occurrences of σ is denoted by $\sigma^n a$. Consider a dtla $M = (Q, \Sigma, \Delta, R, A, P, \delta)$ such that $M(\sigma^n a) = a$ and $M(\sigma^n b) = \sigma^n b$ for every $n \in \mathbb{N}$. It is, in fact, the dtla M_{ex} of the Introduction, for this particular input alphabet. Its set of look-ahead states is $P = \{p_a, p_b\}$ with transition function δ defined by $\delta(a) = p_a$, $\delta(b) = p_b$, $\delta(\sigma, p_a) = p_a$ and $\delta(\sigma, p_b) = p_b$. Its set of states is $Q = \{q\}$, its axioms are $A(p_a) = a$ and $A(p_b) = q(x_0)$, and R contains the rules $q(\sigma(x_1 : p_b)) \to \sigma(q(x_1))$ and $q(b) \to b$.

For $C = \sigma^n\perp$, $M(C[p_a]) = a$ and $M(C[p_b]) = \sigma^n\langle q, p_b\rangle$. Since $M(C[p_a]) \sqcap M(C[p_b]) = \perp$, the only node of $M(C[p]) \sqcap M(C[p'])$ with label $\perp$ is ε. Hence, $\mathsf{diff}(M) = \{a\} \cup \{\sigma^n\langle q, p_b\rangle \mid n \in \mathbb{N}\}$ and $\mathsf{maxdiff}(M) = \infty$. Since $\mathsf{diff}(M)$ is infinite, M is not equivalent to a dtop, as will be shown in Corollary 1. □

Example 2. Let $\Sigma = \{\sigma^{(2)}, aa^{(0)}, ab^{(0)}, ba^{(0)}, bb^{(0)}\}$ where we view aa, ab, ba and bb as symbols, and let $\Delta = \Sigma \cup \{\#^{(2)}, a^{(0)}, b^{(0)}\}$ with $\sigma^{(3)}$ instead of $\sigma^{(2)}$. We consider a dtla M such that $M(yz) = yz$ for $y, z \in \{a, b\}$; moreover, $M(\sigma(s_1, s_2)) = \sigma(M(s_1), M(s_2), \#(y, z))$ where $y \in \{a, b\}$ is the first letter of the label of the left-most leaf of $\sigma(s_1, s_2)$ and $z \in \{a, b\}$ is the second letter of the label of its right-most leaf. It has four look-ahead states p_{yz} with $y, z \in \{a, b\}$, such that $\delta(yz) = p_{yz}$ and $\delta(\sigma, p_{wx}, p_{yz}) = p_{wz}$ for all $w, x, y, z \in \{a, b\}$. It has one state q, its axioms are $A(p_{yz}) = q(x_0)$, and its rules are $q(yz) \to yz$ and $q(\sigma(x_1 : p_{wx}, x_2 : p_{yz})) \to \sigma(q(x_1), q(x_2), \#(w, z))$ for all $w, x, y, z \in \{a, b\}$.

Consider a Σ-context C and the trees $M(C[p_{aa}])$ and $M(C[p_{ba}])$. Let u be the node of C with $\mathsf{lab}(C, u) = \perp$. It is easy to see that the nodes of $M(C[p]) \sqcap M(C[p'])$ with label $\perp$ are the node u and all nodes $v \cdot (3, 1)$ such that $v \neq u$ is a node of C and u is the left-most leaf of C/v. That gives the difference trees $M(C[p_{aa}])/u = \langle q, p_{aa}\rangle$, $M(C[p_{ba}])/u = \langle q, p_{ba}\rangle$, $M(C[p_{aa}])/v \cdot (3, 1) = a$ and $M(C[p_{ba}])/v \cdot (3, 1) = b$. Thus, $\mathsf{diff}(M) = \{a, b\} \cup \{\langle q, p_{yz}\rangle \mid y, z \in \{a, b\}\}$ and $\mathsf{maxdiff}(M) = 0$.

A dtop N equivalent to M has states q_0, q_1, q_2, axiom $q_0(x_0)$, and rules $q_0(yz) \to yz$, $q_1(yz) \to y$, $q_2(yz) \to z$ for $y, z \in \{a, b\}$, $q_2(\sigma(x_1, x_2)) \to q_2(x_2)$, $q_1(\sigma(x_1, x_2)) \to q_1(x_1)$, and $q_0(\sigma(x_1, x_2)) \to \sigma(q_0(x_1), q_0(x_2), \#(q_1(x_1), q_2(x_2)))$. □

3 Normal Form

In this section we state a normal form for (total) dtlas M, together with its effect on $\mathsf{maxdiff}(M)$. We start by requiring a simple and technically convenient property so that every state of M only translates input trees that have the same look-ahead state; moreover, the rules satisfy a completeness condition.

A dtla M is *look-ahead uniform* (for short, *la-uniform*) if there is a mapping $\rho : Q \to P$ (called *la-map*) satisfying the following conditions, for $p \in P$ and $q, \bar{q} \in Q$:

(1) If $q(x_0)$ occurs in $A(p)$, then $\rho(q) = p$.

(2) For every rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta$ in R: $\rho(q) = \delta(a, p_1, \ldots, p_k)$, and if $\bar{q}(x_i)$ occurs in ζ then $\rho(\bar{q}) = p_i$.

(3) For every $q \in Q$, $a \in \Sigma$ of rank $k \geq 0$, and $p_1, \ldots, p_k \in P$ such that $\delta(a, p_1, \ldots, p_k) = \rho(q)$, there is a rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta$ in R.

If M is la-uniform, then the domain of $[\![q]\!]_M$ is $[\![\rho(q)]\!]_M$ for every $q \in Q$. This implies that M° is la-uniform with the same la-map ρ as M.

Clearly, the dtla M of Example 1 is la-uniform with $\rho(q) = p_b$, but the dtla M of Example 2 is not la-uniform.

Example 3. We change the dtla M of Example 2 into an la-uniform dtla (still calling it M), with the same look-ahead automaton as M, by adding look-ahead information to its states. Thus, it has set of states $Q = \{q_{yz} \mid y, z \in \{a, b\}\}$ with $\rho(q_{yz}) = p_{yz}$, axioms $A(p_{yz}) = q_{yz}(x_0)$, and rules $q_{yz}(yz) \to yz$ and $q_{wz}(\sigma(x_1 : p_{wx}, x_2 : p_{yz})) \to \sigma(q_{wx}(x_1), q_{yz}(x_2), \#(w, z))$ for all $w, x, y, z \in \{a, b\}$. □

A dtla M is *earliest* if it is la-uniform and, for every state q of M, $\mathsf{rlabs}_M(q) := \{\mathsf{lab}(q_M(s), \varepsilon) \mid s \in [\![\rho(q)]\!]_M\} \subseteq \Delta$ is not a singleton. Thus, M is *not* earliest if it has a state q for which the roots of all output trees $q_M(s)$, $s \in \mathcal{T}_\Sigma$, have the same label; intuitively, the node with that label could be produced earlier by M. A dtla M is *canonical* if it is earliest and $[\![q]\!]_M \neq [\![q']\!]_M$ for all distinct states q, q' of M.

It is easy to see that the dtla M of Example 3 is canonical: for all $y, z \in \{a, b\}$, $\mathsf{rlabs}_M(q_{yz}) = \{yz, \sigma\}$ and $[\![q_{yz}]\!]_M$ is the restriction of $[\![M]\!]$ to $[\![p_{yz}]\!]_M$.

We now present (without proof) the fact that canonicalness is a normal form for dtlas, generalizing the normal form for dtops in [6] for the total case.

Theorem 1. *For every total dtla M, one can construct an equivalent canonical dtla $\mathsf{can}(M)$, with the same look-ahead automaton as M, such that $\mathsf{maxdiff}(M) - 8^{|M|^3} \leq \mathsf{maxdiff}(\mathsf{can}(M)) \leq \mathsf{maxdiff}(M) + 8^{|M|^3}$, where $|M|$ is the size of M.*

4 Difference Tuples

Let M be a dtpla and let $P = \{\hat{p}_1, \ldots, \hat{p}_n\}$, where the order of the look-ahead states is fixed as indicated. Recall that a dtpla is a dtla that is not a dtop, hence $n \geq 2$. For a given context C consider the trees $M(C[\hat{p}_1]), \ldots, M(C[\hat{p}_n])$. Intuitively, the largest common prefix of these trees does *not* depend on the

look-ahead. In contrast, the subtrees that are not part of the largest common prefix, *do* depend on the look-ahead information.

For trees $t_1, \ldots, t_n \in \mathcal{T}_\Delta(Q \times P)$ we define a subset of $\mathcal{T}_\Delta(Q \times P)^n$ as follows: $\mathsf{diftup}(t_1, \ldots, t_n) := \{(t_1/v, \ldots, t_n/v) \mid v \in V_\perp(\sqcap\{t_1, \ldots, t_n\})\}$. We define the *set of difference tuples* of M as $\mathsf{diftup}(M) := \bigcup_{C \in \mathcal{C}_\Sigma} \mathsf{diftup}(M(C[\hat{p}_1]), \ldots, M(C[\hat{p}_n]))$. For a Σ-context C we define the Δ-pattern $\mathsf{pref}(M, C) := \sqcap\{M(C[p]) \mid p \in P\}$.

We wish to decide whether M is equivalent to a dtop. If there exists such a dtop N, then we may expect intuitively for any $s \in \mathcal{T}_\Sigma$, that $N(C[s]) = t[(q_1)_N(s), \ldots, (q_r)_N(s)]$ where $t = \mathsf{pref}(M, C) = \sqcap\{M(C[\hat{p}_1]), \ldots, M(C[\hat{p}_n])\}$ and $r = |V_\perp(t)|$; in other words, since N does not know the look-ahead state $\delta_M(s)$ of s, it translates C into the largest common prefix of the output trees $M(C[\hat{p}_1]), \ldots, M(C[\hat{p}_n])$. Moreover, if the ith occurrence of $\perp$ is at node v_i of t, $1 \leq i \leq r$, then we expect the difference tuple $(M(C[\hat{p}_1])/v_i, \ldots, M(C[\hat{p}_n])/v_i)$ to be stored in the state q_i of N; in this way N is prepared to continue its simulation of M on the subtree s. This is shown in Lemma 4, for canonical M and earliest N. If N is canonical, then its states are in one-to-one correspondence with the difference tuples of M, as shown in Lemma 5.

It is easy to show that every component of a difference tuple is a difference tree, and every difference tree is a subtree of a component of a difference tuple. Consequently, the maximal height of the components of the difference tuples of M is $\mathsf{maxdiff}(M)$, see [7, Lemma 17]. This implies that $\mathsf{diftup}(M)$ is finite if and only if $\mathsf{diff}(M)$ is finite.

Example 4. For the dtla M of Example 1, with the order $P = \{p_a, p_b\}$, we obtain that $\mathsf{diftup}(M) = \{(a, \sigma^n\langle q, p_b\rangle) \mid n \in \mathbb{N}\}$.

For the dtla M of Example 3 (which is the la-uniform version of the dtla of Example 2) it is not difficult to see that $\mathsf{diff}(M) = \{a, b\} \cup \{\langle q_{yz}, p_{yz}\rangle \mid y, z \in \{a, b\}\}$, and that the set $\mathsf{diftup}(M)$ consists of the three 4-tuples (a, a, b, b), (a, b, a, b) and $(\langle q_{aa}, p_{aa}\rangle, \langle q_{ab}, p_{ab}\rangle, \langle q_{ba}, p_{ba}\rangle, \langle q_{bb}, p_{bb}\rangle)$, with $P = \{p_{aa}, p_{ab}, p_{ba}, p_{bb}\}$. □

In the next lemmas, M is a canonical dtpla (with la-map ρ_M) and N a canonical dtop equivalent to M, i.e., $[\![M]\!] = [\![N]\!]$. We assume that the unique look-ahead state of N is $\perp$; for a Σ-context C we of course write C instead of $C[\perp]$.

We first formalize the fact that the translation of an input tree by M is always ahead of its translation by N, in a uniform way. An *aheadness mapping* from N to M is a function $\varphi : Q_N \times P_M \to \mathcal{T}_\Delta(Q_M \times P_M)$ such that for every $C \in \mathcal{C}_\Sigma$ and $p \in P_M$,

$$M(C[p]) = N(C)[\langle q, \perp\rangle \leftarrow \varphi(q, p) \mid q \in Q_N]. \tag{1}$$

Note that $\varphi(q, p)$ is in $\mathcal{T}_\Delta(\{\langle \bar{q}, p\rangle \mid \bar{q} \in Q_M, \rho_M(\bar{q}) = p\})$. Intuitively, φ defines the exact amount in which M is ahead of N, which is independent of C.

For the next lemma it is essential that M is canonical.

Lemma 2. *There is a unique aheadness mapping φ from N to M.*

Proof. We first show that M is ahead of N, i.e., that all output symbols produced by N on a given input context are also produced by M. Let $p \in P_M$ and $C \in \mathcal{C}_\Sigma$.

Claim 1. $V_d(N(C)) \subseteq V_d(M(C[p]))$ for every $d \in \Delta$.
Equivalently, $N(C)[\langle q, \bot\rangle \leftarrow \bot \mid q \in Q_N] \sqsubseteq M(C[p])$.

Proof: By induction on the length of the nodes of $N(C)$. Let $v \in V_d(N(C))$ with $d \in \Delta$. Since the labels of v's proper ancestors are in Δ, $v \in V(M(C[p]))$ by induction. Consider an arbitrary $s \in [\![p]\!]_M$. By Lemma 1, $v \in V_d(N(C[s]))$. Since $[\![M]\!] = [\![N]\!]$, $M(C[s]) = N(C[s])$ and so $v \in V_d(M(C[s]))$. Suppose that $v \notin V_d(M(C[p]))$. Then, again by Lemma 1, v has some label $\langle q, p\rangle$ in $M(C[p])$ such that $q_M(s)$ has root label d. Since this holds for every $s \in [\![p]\!]_M$, we obtain that $\mathsf{rlabs}_M(q) = \{d\}$ contradicting the fact that M is earliest. Note that, since M is la-uniform, $\rho_M(q) = p$.

Next we show that the amount in which M is ahead of N, is independent of C. Let $p \in P_M$, $C_1, C_2 \in \mathcal{C}_\Sigma$, $v_1, v_2 \in \mathbb{N}_+^*$ and $q \in Q_N$.

Claim 2. If $N(C_1)/v_1 = N(C_2)/v_2 = \langle q, \bot\rangle$, then $M(C_1[p])/v_1 = M(C_2[p])/v_2$.

Proof: By Claim 1, v_i is a node of $M(C_i[p])$. Let $t_i \in \mathcal{T}_\Delta(Q_M \times \{p\})$ denote the tree $M(C_i[p])/v_i$. For every $s \in [\![p]\!]_M$, $N(C_1[s])/v_1 = N(C_2[s])/v_2 = q_N(s)$ by Lemma 1, and so $M(C_1[s])/v_1 = M(C_2[s])/v_2$. Hence, again by Lemma 1, $t_1\Psi_s = t_2\Psi_s$ for all $s \in [\![p]\!]_M$, where $\Psi_s = [\langle q, p\rangle \leftarrow q_M(s) \mid q \in Q_M]$. Suppose that $t_1 \neq t_2$. Then there is a leaf v of, e.g., t_1 with label $\langle q_1, p\rangle$ such that v is a node of t_2 with $t_2/v \neq \langle q_1, p\rangle$. If the root label of t_2/v is $d \in \Delta$, then $(q_1)_M(s)$ has root label d for all $s \in [\![p]\!]_M$, contradicting the fact that M is earliest. If t_2/v equals $\langle q_2, p\rangle$ with $q_1 \neq q_2$, then $(q_1)_M(s) = (q_2)_M(s)$ for all $s \in [\![p]\!]_M$. Since $[\![p]\!]_M$ is the domain of both $[\![q_1]\!]_M$ and $[\![q_2]\!]_M$, we obtain that $[\![q_1]\!]_M = [\![q_2]\!]_M$, contradicting the fact that M is canonical.

An aheadness mapping from N to M can now be defined as follows. Let $q \in Q_N$ and $p \in P_M$. Since, by convention, q is reachable, there is a Σ-context C such that $N(C)$ has a node v labeled $\langle q, \bot\rangle$. By Claim 1, v is a node of $M(C[p])$ and we define $\varphi(q, p) = M(C[p])/v$. By Claim 2, the definition of φ does not depend on C and v. It is easy to see that φ is an aheadness mapping, and that it is unique. □

Lemma 3. *For every $s \in \mathcal{T}_\Sigma$ and $q \in Q_N$,*
if $\delta_M(s) = p$, then $q_N(s) = \varphi(q, p)[\langle \bar{q}, p\rangle \leftarrow \bar{q}_M(s) \mid \bar{q} \in Q_M]$.

Proof. Since q is reachable, there exist C, v such that $N(C)/v = \langle q, \bot\rangle$. By (1), $M(C[p])/v = \varphi(q, p)$. Since M and N are equivalent, $N(C[s]) = M(C[s])$. Applying Lemma 1 twice, we obtain that $q_N(s) = N(C[s])/v = M(C[s])/v = (M(C[p])/v)[\langle \bar{q}, p\rangle \leftarrow \bar{q}_M(s) \mid \bar{q} \in Q_M]$, which proves the equation. □

The next lemma expresses our intuition that the output of N on input C is the largest common prefix of the outputs of M on all inputs $C[p]$, $p \in P$, such that the difference tuples of M are stored in the states of N. Its proof uses that N is earliest. For a tree $t \in \mathcal{T}_\Delta(Q_N \times \{\bot\})$ we define the Δ-pattern $t\Phi := t[\langle q, \bot\rangle \leftarrow \bot \mid q \in Q_N]$; similarly, for $t \in \mathcal{T}_\Delta(Q_N(X))$, we define $t\Phi := t[q(x_i) \leftarrow \bot \mid q \in Q_N, i \in \mathbb{N}]$.

Lemma 4. *For every $C \in \mathcal{C}_\Sigma$, $N(C)\Phi = \mathsf{pref}(M, C)$; moreover, for every $v \in \mathbb{N}_+^*$, $q \in Q_N$ and $p \in P_M$, if $N(C)/v = \langle q, \bot\rangle$ then $\varphi(q, p) = M(C[p])/v$.*

Proof. By Equation (1), $N(C)\Phi \sqsubseteq M(C[p])$ for every $p \in P_M$ (cf. Claim 1 in the proof of Lemma 2), and so $N(C)\Phi \sqsubseteq \mathsf{pref}(M, C)$. To show equality, we prove for every $v \in V_\bot(N(C)\Phi)$ that $v \in V_\bot(\mathsf{pref}(M, C))$. Let $N(C)/v = \langle q, \bot\rangle$ for $q \in Q_N$. Then, by Equation (1), $M(C[p])/v = \varphi(q, p)$ for every $p \in P_M$ (which proves the second part of this lemma). Suppose that $v \in V_d(\mathsf{pref}(M, C))$ with $d \in \Delta$. Then $v \in V_d(M(C[p]))$ and so $\mathsf{lab}(\varphi(q, p), \varepsilon) = d$ for every $p \in P_M$. Then, by Lemma 3, $\mathsf{lab}(q_N(s), \varepsilon) = d$ for every $s \in \mathcal{T}_\Sigma$, contradicting the fact that N is earliest. □

If M is equivalent to a dtop, then it is equivalent to a canonical dtop by Theorem 1. By [6, Theorem 15], equivalent canonical dtops are the same (modulo a renaming of states). Thus, if M is equivalent to a dtop, then it is equivalent to a *unique* canonical dtop $\mathsf{td}(M)$. In the next three lemmas we give another proof of this, and we show that $\mathsf{td}(M)$ can be constructed from M and $\mathsf{diftup}(M)$. We start by showing that $Q_{\mathsf{td}(M)}$ can be identified with $\mathsf{diftup}(M)$. The proof uses that N is canonical.

Lemma 5. *For a state $q \in Q_N$, let $\psi(q) = (\varphi(q, \hat{p}_1), \ldots, \varphi(q, \hat{p}_n))$, where $P_M = \{\hat{p}_1, \ldots, \hat{p}_n\}$. Then ψ is a bijection between Q_N and $\mathsf{diftup}(M)$.*

Proof. (i) $\psi(q) \in \mathsf{diftup}(M)$. Proof: There are C, v such that $N(C)/v = \langle q, \bot\rangle$. By Lemma 4, $v \in V_\bot(\mathsf{pref}(M, C))$ and $M(C[\hat{p}_i])/v = \varphi(q, \hat{p}_i)$ for every i. Thus $\psi(q) \in \mathsf{diftup}(M)$. (ii) ψ is surjective. Proof: If $(t_1, \ldots, t_n) \in \mathsf{diftup}(M)$ then there are C, v such that $\mathsf{pref}(M, C)/v = \bot$ and $M(C[\hat{p}_i])/v = t_i$ for every i. By Lemma 4, $N(C)/v = \langle q, \bot\rangle$ for some $q \in Q_N$, and $M(C[\hat{p}_i])/v = \varphi(q, \hat{p}_i)$. Thus, $t_i = \varphi(q, \hat{p}_i)$ for every i. (iii) ψ is injective. Proof: Let $\psi(q_1) = \psi(q_2)$. By Lemma 3, $(q_1)_N(s) = (q_2)_N(s)$ for all $s \in \mathcal{T}_\Sigma$, i.e., $[\![q_1]\!]_N = [\![q_2]\!]_N$, so $q_1 = q_2$ because N is canonical. □

Corollary 1. *Let M be a total dtla. If M is equivalent to a dtop, then $\mathsf{diff}(M)$ is finite.*

Proof. If M is a dtop, then $\mathsf{diff}(M) = \emptyset$. Now let M be a dtpla, equivalent to a dtop. By Theorem 1, the canonical dtla $\mathsf{can}(M)$ is equivalent to a canonical dtop. By Lemmas 2 and 5, $\mathsf{diftup}(\mathsf{can}(M))$ is finite, and so $\mathsf{diff}(\mathsf{can}(M))$ is finite. Hence $\mathsf{diff}(M)$ is finite because $\mathsf{maxdiff}(M) \leq \mathsf{maxdiff}(\mathsf{can}(M)) + 8^{|M|^3}$, cf. Theorem 1. □

Next we show how to compute the axiom of $\mathsf{td}(M)$, representing the states of $\mathsf{td}(M)$ by difference tuples. For a tree $t \in \mathcal{T}_\Delta(Q_M(X))$ we define $t\Omega \in \mathcal{T}_\Delta(Q_M \times P_M)$ by $t\Omega := t[q(x_i) \leftarrow \langle q, \rho_M(q)\rangle \mid q \in Q_M, i \in \mathbb{N}]$; similarly, for $t \in \mathcal{T}_\Delta(Q_N(X))$, we define $t\Omega := t[q(x_i) \leftarrow \langle q, \bot\rangle \mid q \in Q_N, i \in \mathbb{N}]$.

Lemma 6. *$A_N\Phi = \sqcap\{A_M(p)\Omega \mid p \in P_M\}$; moreover, for every $v \in \mathbb{N}_+^*$, $q \in Q_N$ and $p \in P_M$, if $A_N/v = q(x_0)$ then $\varphi(q, p) = A_M(p)\Omega/v$.*

Proof. Clearly, $N(\bot) = A_N\Omega$ and $M(p) = A_M(p)\Omega$ for every $p \in P_M$. Hence by Lemma 4, with $C = \bot$, $A_N\Phi = A_N\Omega\Phi = N(\bot)\Phi = \mathsf{pref}(M, \bot) = \sqcap\{M(p) \mid p \in P_M\} = \sqcap\{A_M(p)\Omega \mid p \in P_M\}$. If $A_N/v = q(x_0)$ then $N(\bot)/v = A_N\Omega/v = \langle q, \bot\rangle$; so by Lemma 4, with $C = \bot$, $\varphi(q,p) = M(p)/v = A_M(p)\Omega/v$ for every $p \in P_M$. □

Finally we show, without proof, how to compute the rules of $\mathsf{td}(M)$. Let M be an la-uniform dtla, Q_N a finite set and $\varphi : Q_N \times P_M \to \mathcal{T}_\Delta(Q_M \times P_M)$ a mapping such that $\varphi(q,p) \in \mathcal{T}_\Delta(\{\langle \bar{q}, p\rangle \mid \bar{q} \in Q_M, \rho_M(\bar{q}) = p\})$ for every $q \in Q_N$ and $p \in P_M$. Then we define for every $q \in Q_N$, $a \in \Sigma$ of rank $k \geq 0$, and $p_1, \ldots, p_k \in P_M$, the tree $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k) := \varphi(q,p)[\langle \bar{q}, p\rangle \leftarrow \mathsf{rhs}_M(\bar{q}, a, p_1, \ldots, p_k) \mid \bar{q} \in Q_M]$ where $p = \delta_M(a, p_1, \ldots, p_k)$. For $k \in \mathbb{N}$, let $[k] = \{1, \ldots, k\}$.

Lemma 7. (1) *For every $q \in Q_N$ and $a \in \Sigma$ of rank $k \geq 0$,*
$\mathsf{rhs}_N(q,a)\Phi = \sqcap\{\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega \mid p_1, \ldots, p_k \in P_M\}$.

(2) *Let $q \in Q_N$, $a \in \Sigma$ of rank $k \geq 0$, and $i \in [k]$. For $j \in [k]$, $j \neq i$, let $s_j \in \mathcal{T}_\Sigma$ and $p_j = \delta_M(s_j)$. Let $\Psi_{iM} = [\bar{q}(x_j) \leftarrow \bar{q}_M(s_j) \mid \bar{q} \in Q_M, j \in [k], j \neq i]\Omega$. For every $v \in V_\bot(\mathsf{rhs}_N(q,a)\Phi)$,*
   (a) *$\mathsf{rhs}_N(q,a)/v \in Q_N(\{x_i\})$ if and only if*
   $v \in V_\bot(\sqcap\{\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_{i-1}, p, p_{i+1}, \ldots, p_k)\Psi_{iM} \mid p \in P_M\})$, and
   (b) *for every $\bar{q} \in Q_N$ and $p \in P_M$, if $\mathsf{rhs}_N(q,a)/v = \bar{q}(x_i)$ then*
   $\varphi(\bar{q}, p) = \mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_{i-1}, p, p_{i+1}, \ldots, p_k)\Psi_{iM}/v$.

By the last three lemmas, every dtpla M that is equivalent to a dtop, is equivalent to a unique canonical dtop $\mathsf{td}(M)$, modulo a renaming of states. Based on these same lemmas, we can now construct $\mathsf{td}(M)$ from any given canonical dtpla M for which $\mathsf{diftup}(M)$ is a given finite set. The construction returns the answer 'no' if M is not equivalent to any dtop. We construct the dtop $N = \mathsf{td}(M)$, if it exists, by taking $Q_N = \mathsf{diftup}(M)$, defining $\varphi : Q_N \times P_M \to \mathcal{T}_\Delta(Q_M \times P_M)$ as $\varphi((t_1, \ldots, t_n), \hat{p}_i) = t_i$ for $i \in [n]$ (in accordance with Lemma 5), and constructing the axiom and rules of N according to Lemmas 6 and 7, respectively (i.e., by viewing the statements of these lemmas as definitions). In Lemma 7(2) we choose s_j arbitrarily but fixed. If the construction of an axiom or a rule fails because a possible state occurring in it (which is a tuple in $\mathcal{T}_\Delta(Q_M \times P_M)^n$) is not a difference tuple of M, then the answer is 'no'. The construction of a rule can also fail (and produce the answer 'no') when a node $v \in V_\bot(\mathsf{rhs}_N(q,a)\Phi)$ is not an element of $V_\bot(\sqcap\{\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_{i-1}, p, p_{i+1}, \ldots, p_k)\Psi_{iM} \mid p \in P_M\})$ for any i, see Lemma 7(2)(a). If the construction of the dtop N succeeds, then it remains to test whether M and N are equivalent (because, by Lemmas 5, 6 and 7, if M is equivalent to a dtop then it is equivalent to N). If they are equivalent then the construction returns the dtop $N = \mathsf{td}(M)$, otherwise the answer is 'no'. Equivalence of dtlas is decidable by [9] (see also [6, Corollary 19]). It is shown in [7, Section 6.1] that there is a simple direct test for equivalence of M and N.

Unfortunately, we do not know whether it is decidable if $\mathsf{diftup}(M)$ is finite, and whether it can be computed if it is finite. We now show that, to determine whether a dtla M is equivalent to a dtop, it suffices to have an upper bound for $\mathsf{maxdiff}(M)$.

Theorem 2. *It is decidable for a given total dtla M and a given difference bound for M whether there exists a dtop N such that $[\![M]\!] = [\![N]\!]$, and if so, such a dtop N can be constructed.*

Proof. Let M be a (total) dtla and let $h(M)$ be a difference bound for M. We may, of course, assume that M is a dtpla. By Theorem 1 we may assume that M is canonical, because $h(M) + 8^{|M|^3}$ is a (computable) difference bound for $\mathsf{can}(M)$.

So, let M be a canonical dtpla and let $h(M)$ be a difference bound for M. This means that if $\mathsf{diftup}(M)$ is finite, then the height of the components of the difference tuples of M is at most $h(M)$. We now decide whether M is equivalent to a dtop by constructing $\mathsf{td}(M)$ as described before this theorem. However, since $\mathsf{diftup}(M)$ is not given, we construct $N = \mathsf{td}(M)$ incrementally, using a variable Q_N to accumulate its states (which are all assumed to be reachable). In accordance with Lemma 5 we take $Q_N \subseteq \mathcal{T}_\Delta(Q_M \times P_M)^n$ and $\varphi((t_1, \ldots, t_n), \hat{p}_i) = t_i$ for $i \in [n]$. We first construct the axiom A_N according to Lemma 6 and initialize the set Q_N with the states, i.e., the tuples in $\mathcal{T}_\Delta(Q_M \times P_M)^n$, that occur in that axiom. If the height of one of the components of one of those tuples is larger than $h(M)$, then either $\mathsf{diftup}(M)$ is infinite or that tuple is not a difference tuple of M, and we stop the construction with answer 'no', indicating that M is not equivalent to any dtop. Then, repeatedly, for every $q \in Q_N$ and $a \in \Sigma$ we construct $\mathsf{rhs}_N(q, a)$ according to Lemma 7, and we add to Q_N the states that occur in that right-hand side. If the construction of $\mathsf{rhs}_N(q, a)$ fails or if the height of one of the components of its states is larger than $h(M)$, then the answer is 'no'. If the construction of the dtop N succeeds, then it remains to test whether M and N are equivalent. □

In the next example we show that without the tests on height, the construction may not halt; in such a case it can be viewed as computing an *infinite* dtop equivalent to M. In Example 6 the construction of N succeeds and N is equivalent to M.

Example 5. Consider the dtla M of Example 1. It is easy to see that M is canonical. In Example 4 we have seen that $\mathsf{diftup}(M) = \{(a, \sigma^n\langle q, p_b\rangle) \mid n \in \mathbb{N}\}$.

We now apply to M the construction of N in the proof of Theorem 2, without the tests on height. By Lemma 6, $A_N\Phi = a \sqcap \langle q, p_b\rangle = \bot$ and so $A_N = q_0(x_0)$ with $\varphi(q_0, p_a) = a$ and $\varphi(q_0, p_b) = \langle q, p_b\rangle$, i.e., $q_0 = (a, \langle q, p_b\rangle)$. Assume now that the algorithm has constructed the state q_n with $\varphi(q_n, p_a) = a$ and $\varphi(q_n, p_b) = \sigma^n\langle q, p_b\rangle$, i.e., q_n is the difference tuple $(a, \sigma^n\langle q, p_b\rangle)$ of M. By Lemma 7(1), $\mathsf{rhs}_N(q_n, b) = \mathsf{rhs}_{M,\varphi}(q_n, b) = \varphi(q_n, p_b)[\langle \bar{q}, p_b\rangle \leftarrow \mathsf{rhs}_M(\bar{q}, b) \mid \bar{q} \in Q_M] = \varphi(q_n, p_b)[\langle q, p_b\rangle \leftarrow b] = \sigma^n b$. Thus, N has the rule $q_n(b) \to \sigma^n b$. Similarly, $\mathsf{rhs}_N(q_n, a) = \mathsf{rhs}_{M,\varphi}(q_n, a) = \varphi(q_n, p_a) = a$ and so N has the rule $q_n(a) \to a$. Next, we compute $\mathsf{rhs}_N(q_n, \sigma)$. To do so we need $\mathsf{rhs}_{M,\varphi}(q_n, \sigma, p)$ for every $p \in P_M$. For $p = p_b$ we have $\mathsf{rhs}_{M,\varphi}(q_n, \sigma, p_b) = \varphi(q_n, p_b)[\langle q, p_b\rangle \leftarrow \mathsf{rhs}_M(q, \sigma, p_b)] = \sigma^n\sigma q(x_1) = \sigma^{n+1}q(x_1)$, and for $p = p_a$ we have $\mathsf{rhs}_{M,\varphi}(q_n, \sigma, p_a) = \varphi(q_n, p_a) = a$. Thus, by Lemma 7(1), $\mathsf{rhs}_N(q_n, \sigma)\Phi = a \sqcap \sigma^{n+1}\langle q, p_b\rangle = \bot$. Hence, $\mathsf{rhs}_N(q_n, \sigma) = q(x_1)$ for some $q \in Q_N$. By Lemma 7(2)(b), $\varphi(q, p_y) = \mathsf{rhs}_{M,\varphi}(q_n, \sigma, p_y)\Omega$ for

$y \in \{a,b\}$ and so $\varphi(q,p_a) = a$ and $\varphi(q,p_b) = \sigma^{n+1}\langle q,p_b\rangle$. Thus, $q = q_{n+1}$ and N has the rule $q_n(\sigma(x_1)) \to q_{n+1}(x_1)$. This shows that the construction does not halt. It can be viewed as constructing the *infinite* dtop N with $Q_N = \{q_n \mid n \in \mathbb{N}\} = \mathsf{diftup}(M)$, $A_N = q_0(x_0)$ and rules $q_n(a) \to a$, $q_n(b) \to \sigma^n b$ and $q_n(\sigma(x_1)) \to q_{n+1}(x_1)$ for every $n \in \mathbb{N}$. Clearly, N is equivalent to M. *With* a given difference bound h, the construction halts when constructing q_{h+1}. □

Example 6. Consider the dtla M of Example 3. As observed after Example 3, M is canonical. We have seen in Example 4 that $\mathsf{diftup}(M)$ consists of the three 4-tuples (a,a,b,b), (a,b,a,b) and $(\langle q_{aa},p_{aa}\rangle, \langle q_{ab},p_{ab}\rangle, \langle q_{ba},p_{ba}\rangle, \langle q_{bb},p_{bb}\rangle)$.

We construct N as in the proof of Theorem 2; since $\mathsf{maxdiff}(M) = 0$, the construction is the same for every difference bound $h(M)$. By Lemma 6, $A_N = q_0(x_0)$ with $\varphi(q_0,p_{yz}) = \langle q_{yz},p_{yz}\rangle$ for $y,z \in \{a,b\}$. Hence, $q_0 = (\langle q_{aa},p_{aa}\rangle, \langle q_{ab},p_{ab}\rangle, \langle q_{ba},p_{ba}\rangle, \langle q_{bb},p_{bb}\rangle)$. Then Lemma 7(1) implies the equalities $\mathsf{rhs}_N(q_0,yz)\Phi = \mathsf{rhs}_{M,\varphi}(q_0,yz) = \varphi(q_0,p_{yz})[\langle q_{yz},p_{yz}\rangle \leftarrow \mathsf{rhs}_M(q_{yz},yz)] = \mathsf{rhs}_M(q_{yz},yz) = yz$, so N has the rules $q_0(yz) \to yz$ for all $y,z \in \{a,b\}$. To compute $\mathsf{rhs}_N(q_0,\sigma)$, observe that for every $w,x,y,z \in \{a,b\}$, $\mathsf{rhs}_{M,\varphi}(q_0,\sigma,p_{wx},p_{yz}) = \varphi(q_0,p_{wz})[\langle q_{wz},p_{wz}\rangle \leftarrow \mathsf{rhs}_M(q_{wz},\sigma,p_{wx},p_{yz})] = \mathsf{rhs}_M(q_{wz},\sigma,p_{wx},p_{yz}) = \sigma(q_{wx}(x_1),q_{yz}(x_2),\#(w,z))$. By Lemma 7(1), $\mathsf{rhs}_N(q_0,\sigma)\Phi = \sigma(\bot,\bot,\#(\bot,\bot))$. Thus, N may have a rule of the form

$$q_0(\sigma(x_1,x_2)) \to \sigma(q_3(x_{i_3}),q_4(x_{i_4}),\#(q_1(x_{i_1}),q_2(x_{i_2}))).$$

Let $s_1 = s_2 = aa$. From Lemma 7(2)(a) we obtain for $v = (3,1)$ that

$$\begin{aligned} i_1 = 1 &\iff v \in V_\bot(\sqcap\{\mathsf{rhs}_{M,\varphi}(q_0,\sigma,p_{wx},p_{aa})\Psi_{1M} \mid w,x \in \{a,b\}\}) \\ &\iff v \in V_\bot(\sqcap\{\sigma(\langle q_{wx},p_{wx}\rangle, aa, \#(w,a)) \mid w,x \in \{a,b\}\}) \end{aligned}$$

if and only if $v \in V_\bot(\sigma(\bot,aa,\#(\bot,a)))$, which is true. So $i_1 = 1$ and $\varphi(q_1,p_{wx}) = w$ for all $w,x \in \{a,b\}$ by Lemma 7(2)(b). Thus, $q_1 = (a,a,b,b)$. Similarly we obtain for $v = (3,2)$ that $i_2 = 2$ and $\varphi(q_2,p_{yz}) = z$, for $v = 1$ that $i_3 = 1$ and $\varphi(q_3,p_{wx}) = \langle q_{wx},p_{wx}\rangle$, and for $v = 2$ that $i_4 = 2$ and $\varphi(q_4,p_{yz}) = \langle q_{yz},p_{yz}\rangle$. Hence $q_2 = (a,b,a,b)$, $q_3 = q_4 = q_0$ and N has the rule

$$q_0(\sigma(x_1,x_2)) \to \sigma(q_0(x_1),q_0(x_2),\#(q_1(x_1),q_2(x_2))).$$

Next we consider q_2. Clearly, both $\mathsf{rhs}_{M,\varphi}(q_2,yz)$ and $\mathsf{rhs}_{M,\varphi}(q_2,\sigma,p_{wx},p_{yz})$ equal z. Thus, N has the rules $q_2(yz) \to z$ and it may have a rule of the form $q_2(\sigma(x_1,x_2)) \to q(x_i)$. Taking again $s_1 = s_2 = aa$, we get that $i = 2$ if and only if ε has label $\bot$ in $\sqcap\{\mathsf{rhs}_{M,\varphi}(q_2,\sigma,p_{aa},p_{yz}) \mid y,z \in \{a,b\}\}$ if and only if $\varepsilon \in V_\bot(a \sqcap b)$, which is true. So $i = 2$ and $\varphi(q,p_{yz}) = z$, which means that $q = q_2$. Hence, N has the rule $q_2(\sigma(x_1,x_2)) \to q_2(x_2)$. Similarly it has the rules $q_1(yz) \to y$ and $q_1(\sigma(x_1,x_2)) \to q_1(x_1)$. So, the construction ends with the dtop N given at the end of Example 2. □

5 Conclusion

A dtla M is *linear* if no right-hand side of a rule contains the same variable twice, and *nonerasing* if no right-hand side of a rule is in $Q(X)$. Our two example dtlas are both.

Theorem 3. *It is decidable for a total linear nonerasing dtla M whether there exists a dtop N such that $[\![M]\!] = [\![N]\!]$, and if so, such a dtop N can be constructed.*

The proof uses (involved) pumping arguments to show that $37 \cdot |M|^5$ is a difference bound for such a dtla M. The same proof holds for dtlas with less stringent restrictions on copying and erasing: total dtlas that are ultralinear and bounded erasing, see [7].

We would like to extend the above result to the nontotal case where a dtla realizes a partial function, to the case where the dtla and the dtop are restricted to a given regular tree language, and to more general dtlas (preferably to all dtlas, of course). Even more generally, we would like to have an algorithm that for a given dtla constructs an equivalent dtla with a minimal number of look-ahead states.

References

1. Alur, R., D'Antoni, L.: Streaming tree transducers. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012, Part II. LNCS, vol. 7392, pp. 42–53. Springer, Heidelberg (2012)
2. Berstel, J.: Transductions and Context-Free Languages. Teubner-Verlag (1979)
3. Choffrut, C.: Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. Theor. Comput. Sci. 5(3), 325–337 (1977)
4. Drewes, F.: Grammatical Picture Generation – A Tree-Based Approach. Springer (2006)
5. Engelfriet, J.: Top-down tree transducers with regular look-ahead. Mathematical Systems Theory 10, 289–303 (1977)
6. Engelfriet, J., Maneth, S., Seidl, H.: Deciding equivalence of top-down XML transformations in polynomial time. J. Comput. Syst. Sci. 75(5), 271–286 (2009)
7. Engelfriet, J., Maneth, S., Seidl, H.: Look-ahead removal for top-down tree transducers. CoRR abs/1311.2400 (2013)
8. Engelfriet, J., Vogler, H.: Macro tree transducers. J. Comput. Syst. Sci. 31(1), 71–146 (1985)
9. Ésik, Z.: Decidability results concerning tree transducers I. Acta Cybern. 5, 1–20 (1980)
10. Filiot, E., Servais, F.: Visibly pushdown transducers with look-ahead. In: Bieliková, M., Friedrich, G., Gottlob, G., Katzenbeisser, S., Turán, G. (eds.) SOFSEM 2012. LNCS, vol. 7147, pp. 251–263. Springer, Heidelberg (2012)
11. Hosoya, H.: Foundations of XML Processing – The Tree-Automata Approach. Cambridge University Press (2010)
12. Knight, K., Graehl, J.: An overview of probabilistic tree transducers for natural language processing. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 1–24. Springer, Heidelberg (2005)
13. Lemay, A., Maneth, S., Niehren, J.: A learning algorithm for top-down XML transformations. In: PODS, pp. 285–296 (2010)
14. Maletti, A., Graehl, J., Hopkins, M., Knight, K.: The power of extended top-down tree transducers. SIAM J. Comput. 39(2), 410–430 (2009)
15. Martens, W., Neven, F., Gyssens, M.: Typechecking top-down XML transformations: Fixed input or output schemas. Inf. Comput. 206(7), 806–827 (2008)