Look-Ahead Removal for Total Deterministic Top-Down Tree Transducers

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

[1] LIACS, Leiden University, The Netherlands
j.engelfriet@liacs.leidenuniv.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. Top-down tree transducers are a convenient formalism for describing tree transformations. They can be equipped with regular look-ahead, which allows them to inspect a subtree before processing it. In certain cases, such a look-ahead can be avoided and the transformation can be realized by a transducer without look-ahead. Removing the look-ahead from a transducer, if possible, is technically highly challenging. For a restricted class of transducers with look-ahead, namely those that are total, deterministic, ultralinear, and bounded erasing, we present an algorithm that, for a given transducer from that class, (1) decides whether it is equivalent to a total deterministic transducer without look-ahead, and (2) constructs such a transducer if the answer is positive. For the whole class of total deterministic transducers with look-ahead we present a similar algorithm, which assumes that a so-called difference bound is known for the given transducer. The designer of a transducer can usually also determine a difference bound for it.

1 Introduction

Many simple tree transformations can be modeled by top-down tree transducers [26, 27]. They are recently used in XML database theory (e.g., [9, 18, 20, 23–25]), in computational linguistics (e.g., [19, 21, 22]) 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 in a (parallel) top-down fashion. A more expressive (but also more complex) model for specifying tree translations is the top-down tree transducer *with regular look-ahead* [6]. It consists of a top-down tree transducer and a finite-state bottom-up tree automaton, called the look-ahead automaton. We may think of its execution in two phases: In a first phase the input tree is relabeled by attaching to each input node the active state of the automaton, called the look-ahead state at that node. In the second phase the top-down tree transducer is executed over the relabeled tree, thus possibly making use of the look-ahead information in the new input labels. As an example, consider a deterministic transducer M_{ex} of which the look-ahead automaton checks whether the input tree contains a leaf labeled a. If so, then M_{ex} outputs a, and otherwise it outputs a copy of the input tree. It should be clear that there is

no deterministic top-down tree transducer (without look-ahead) that realizes the same translation as M_{ex}. The intuitive reason is that in general the complete input tree must be read and buffered in memory, before the appropriate choice of output can be made. How can we formally prove that indeed no deterministic top-down tree transducer (without look-ahead) can realize this translation? In general, is there a method to determine for a given top-down tree transducer *with look-ahead*, whether or not its translation can be realized by a top-down tree transducer *without look-ahead*? And if the answer is yes, can such a transducer be constructed from the given one?

In this paper we give two partial answers to these questions, where we restrict ourselves to total deterministic transducers (which will not be mentioned any more in the remainder of this introduction). For such transducers we provide a general method as discussed above. However, part of the method is not automatic, but depends on additional knowledge about the given transducer with look-ahead (which can usually be determined by the designer of the transducer). For a restricted type of transducers (where the restrictions concern the capability of the transducer to copy and erase) that knowledge can also be obtained automatically, which means that for a thus restricted transducer with look-ahead it is decidable whether its translation can be realized by a (nonrestricted) transducer without look-ahead, and if so, such a transducer can be constructed from the given transducer.

The main notion on which our method is based, is that of a *difference tree* of a top-down tree transducer with regular look-ahead. Consider two trees obtained from one input tree by replacing one of its leaves by two different look-ahead states of the transducer 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 of M arrives in that state at the root of the subtree. Since M is total and deterministic, these output trees exist and are unique, respectively. By removing the largest common prefix of the 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 the look-ahead state of the subtree. Thus, the set 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. For the example transducer M_{ex} above, diff(M_{ex}) 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); thus, diff(M_{ex}) is infinite.

Now the idea of our method is as follows, where we use *dtla* and *dtop* to abbreviate top-down tree transducer with and without look-ahead, respectively. In [9] it was shown that for every dtop an equivalent canonical earliest dtop can be constructed. Earliest means that each output node is produced as early as possible by the transducer, and canonical means that different states of the transducer are inequivalent. We prove that also for every dtla an equivalent canonical earliest dtla (with the same look-ahead automaton) can be constructed, where the earliest and canonical properties are relativized with respect to each look-ahead state. Thus, to devise our method we may restrict attention to canonical earliest transducers. Assume there exists a (canonical earliest) dtop N

equivalent to the (canonical earliest) dtla M. Then the dtla M is at least as early as N. In other words, at each moment of the translation, M may be ahead of N but not vice versa, i.e., the output of N is a prefix of that of M, which is because M has additional information through its look-ahead. 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 must be able to simulate M, it has to store these difference trees in its states. Hence, diff(M) must be finite. Moreover, it turns out that the above description of N's behaviour completely determines N, and so, roughly speaking, N can be constructed from M and diff(M). Note that since diff(M_{ex}) is infinite, the translation of M_{ex} cannot be realized by a dtop.

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; in other words, if a tree in diff(M) has height $> h$, then diff(M) is infinite. 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 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 ultralinear and bounded erasing. Ultralinearity means that the transducer cannot copy an input subtree when it is in a cycle (i.e., in a computation that starts and ends in the same state). Thus it is weaker than the linear property (which forbids copying) but stronger than the finite-copying property [11, 8]. The latter implies that the size of the output tree of an ultralinear dtla is linear in the size of its input tree. Bounded erasing means that the transducer has no cycle in which no output is produced. The proof that a difference bound can be computed for ultralinear and bounded erasing dtlas, is based on pumping arguments that are technically involved.

The paper is structured as follows. Section 2 contains basic terminology, in particular concerning prefixes of trees. Section 3 defines the dtla (deterministic top-down tree transducer with regular look-ahead) and discusses some of its basic properties. It also explains the treatment of look-ahead states that occur in the input tree. In Section 4 we define the notions of difference tree and difference bound, illustrated by some examples. In Section 5 we discuss some normal forms for dtlas, in particular *look-ahead uniformity* which is technically convenient. We prove that for every dtla M there is an equivalent canonical earliest dtla M' (which is also look-ahead uniform), and we show how to compute a difference bound for M' from one of M. Our first main result is proved in Section 6, which is divided in three subsections. Section 6.1 starts with the definition of a *difference tuple* of a dtla M, which generalizes the notion of difference tree by considering all look-ahead states of M rather than just two. If N is a dtop equivalent to M, then its states are in one-to-one correspondence with the difference tuples of M (assuming that both N and M are canonical earliest), see Lemma 26, and its axiom and rules are completely determined by M, see Lemmas 29 and 31. In Section 6.2 we present the algorithm that computes N from M for a given difference bound for M (if such an N exists), still assuming that M is canonical earliest. Section 6.3 states the first main result (for arbitrary M) in Theorem 50. Sections 7– 9 are devoted to the proof of our second main result (Theorem 85). Section 7 continues Section 3 by discussing

some basic properties of dtlas: the *links* that exist between an input tree and its corresponding output tree, and for each node of the output tree, its *origin* in the input tree. In Section 8 the problem of computing a difference bound for a dtla M is reduced to that of computing two related upper bounds: an *output bound* for M and an *ancestral bound* for M. An output bound can be computed for every dtla. Finally, in Section 9, an ancestral bound is computed for every ultralinear and bounded erasing dtla. The computed output bound (in the previous section) and ancestral bound for a dtla M are both based on pumping arguments (simple for the output bound, complicated for the ancestral bound). In both cases a part of the input tree on which M has a cyclic computation, is pumped in such a way that the corresponding output tree contains arbitrarily large difference trees. In the ancestral case the pumping argument is technically based on the fact that M cannot copy and must produce output during its cyclic computations. Since the pumping of trees makes it hard to address nodes by the usual Dewey notation, a *dependency graph* is defined for M such that a cyclic computation of M corresponds to a cycle in its dependency graph; pumping the input tree then corresponds to repeating a cycle in the graph. At the end of Section 9 we consider two other classes of dtlas for which equivalence to a dtop is decidable (and if so, such a dtop can be constructed): *output-monadic* dtlas and *depth-uniform* dtlas. Output-monadic means that every node of an output tree has at most one child. Depth-uniform means, in its simplest form, that all states in the right-hand sides of the rules of the dtla are at the same depth.

Related Work. For deterministic string transducers with regular look-ahead, look-ahead removal is decidable, i.e., 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. We extend that result (for the total case) by proving that look-ahead removal is decidable for output-monadic dtlas.

Look-ahead has been investigated for other types of tree transducers. For macro tree transducers [12, 8] and streaming tree transducers [1], regular look-ahead can always be removed. The same is true for nondeterministic visibly pushdown transducers [15]. 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 [15].

In [16] the deterministic multi bottom-up tree transducer (dmbot) was introduced and shown to have (effectively) the same expressive power as the dtla. Thus, our results can also be viewed as partial answers to the question whether it is decidable for a given dmbot to be equivalent to a dtop.

Note. The results of this paper were first presented at DLT 2014, see [10].

2   Preliminaries

The set of natural numbers is $\mathbb{N} = \{0, 1, 2, \dots\}$, and $\mathbb{N}_+ = \{1, 2, \dots\}$. For $k \in \mathbb{N}$ we define $[k] = \{1, \dots, k\}$; in particular, $[0] = \emptyset$. We assume the usual linear order $\leq$ on $\mathbb{N}$

and on $\mathbb{N} \cup \{\infty\}$, with $n < \infty$ for all $n \in \mathbb{N}$. For a set $S \subseteq \mathbb{N} \cup \{\infty\}$, $\mathsf{lub}\, S$ denotes the least upper bound of the elements of S. If S is finite and nonempty, then $\mathsf{lub}\, S$ is the maximal element of S. Also, $\mathsf{lub}\, \emptyset = 0$.

The domain of a partial function f is denoted $\mathrm{dom}(f)$. For a set A, we denote by A^* the set of sequences, or strings, of elements of A. A string $(a_1, \ldots, a_n) \in A^*$ will be denoted $a_1 \cdots a_n$, unless there is a danger of confusion. The concatenation of two strings u and v is denoted $u \cdot v$ or just uv, and the empty string is denoted ε. A string u is a prefix (postfix) of a string v if there exists a string w such that $v = uw$ ($v = wu$); it is a proper prefix (postfix) if $w \neq \varepsilon$. The length of a string u is denoted $|u|$. The cardinality of a set A is denoted $|A|$.

A directed edge-labeled graph G over a set A (of edge labels) consists of a set V of nodes and a set $E \subseteq V \times A \times V$ of edges. An edge (u, a, v) is said to be an edge with label a that starts at u and ends at v, or shortly, from u to v. A (directed) path π in G is a sequence $e_1 \cdots e_n$ with $n \geq 0$ and $e_i \in E$ for every $i \in [n]$, such that for every $m \in [n-1]$ the edge e_{m+1} starts at the node where e_m ends. If $n = 0$, i.e., $\pi = \varepsilon$, then π is a path from u to u for every $u \in V$. If $n \geq 1$, then π is a path from the start node of e_1 to the end node of e_n. If π is a path from u to v, then it is said to start at u and end at v. A path is a cycle if it is nonempty and starts and ends at the same node.

Top-down tree transducers, which will be recalled in Section 3, work on ranked trees. This means that the number of children of a node of a tree is determined by the symbol at that node. A ranked alphabet Σ is a finite set of symbols such that each symbol $a \in \Sigma$ is implicitly equipped with a rank $\mathsf{rk}(a) \in \mathbb{N}$. For $k \in \mathbb{N}$ we define $\Sigma^{(k)} = \{a \in \Sigma \mid \mathsf{rk}(a) = k\}$. To avoid trivialities, we assume that $\Sigma^{(0)} \neq \emptyset$. To indicate that $\sigma \in \Sigma$ has rank k, we also write it as $\sigma^{(k)}$.

The set $\mathcal{T}_\Sigma$ of (finite, ordered, ranked) trees over the ranked alphabet Σ is the smallest set (of terms) such that $a(t_1, \ldots, t_k) \in \mathcal{T}_\Sigma$ if $k \in \mathbb{N}$, $a \in \Sigma^{(k)}$, and $t_1, \ldots, t_k \in \mathcal{T}_\Sigma$. If $a \in \Sigma^{(0)}$, then we also write a for the tree $a()$. If $a \in \Sigma^{(1)}$ and $t \in \mathcal{T}_\Sigma$, then we also write at for $a(t)$. More generally, for a string $w = a_1 \cdots a_n$ with $n \in \mathbb{N}$ and $a_i \in \Sigma^{(1)}$, we write $w(t) = a_1 \cdots a_n(t)$ for the tree $a_1(a_2(\cdots a_n(t) \cdots))$; in particular $\varepsilon(t) = t$.

We represent the nodes of a tree in Dewey notation, i.e., by strings of positive natural numbers. The empty string ε represents the root node and, for $i \in \mathbb{N}_+$, vi represents the ith child of the node v (and v is the parent of vi). Every node v of a tree t has a label in Σ, denoted $\mathsf{lab}(t, v)$. Formally, the set $V(t) \subseteq \mathbb{N}_+^*$ of nodes (together with their labels) of the tree t is inductively defined as: $V(t) = \{\varepsilon\} \cup \{iv \mid i \in [k], v \in V(t_i)\}$ if $t = a(t_1, \ldots, t_k)$, $a \in \Sigma^{(k)}$, and $t_1, \ldots, t_k \in \mathcal{T}_\Sigma$; moreover, $\mathsf{lab}(t, \varepsilon) = a$ and $\mathsf{lab}(t, iv) = \mathsf{lab}(t_i, v)$. A node u is an ancestor of node v (and v is a descendant of u) if u is a prefix of v; it is a proper ancestor/descendant if it is a proper prefix. For $\Delta \subseteq \Sigma$, we define $V_\Delta(t) = \{v \in V(t) \mid \mathsf{lab}(t, v) \in \Delta\}$; for $a \in \Sigma$, we write $V_a(t)$ instead of $V_{\{a\}}(t)$. The subtree of t rooted at $v \in V(t)$ is denoted by t/v; formally, $t/\varepsilon = t$ and if $t = a(t_1, \ldots, t_k)$ then $t/iv = t_i/v$. The size of t, denoted $\mathsf{size}(t)$, is its number $|V(t)|$ of nodes. The height of t, denoted $\mathsf{ht}(t)$, is the maximal length of its nodes, i.e., $\max\{|v| \mid v \in V(t)\}$. As an example, if $t = \sigma(\sigma(a, b), \tau(b))$, then $V(t) = \{\varepsilon, 1, (1, 1), (1, 2), 2, (2, 1)\}$, $\mathsf{lab}(t, (1, 2)) = b$, $V_b(t) = \{(1, 2), (2, 1)\}$, $t/1 = \sigma(a, b)$, $\mathsf{size}(t) = 6$, and $\mathsf{ht}(t) = 2$. The height of a tuple of trees $\bar{t} = (t_1, \ldots, t_k)$, $k \geq 1$, is defined as $\mathsf{ht}(\bar{t}) = \max\{\mathsf{ht}(t_i) \mid i \in [k]\}$.

Let Δ be a ranked alphabet such that every symbol in $\Sigma \cap \Delta$ has the same rank in Σ and Δ. For a set of trees $\mathcal{T} \subseteq \mathcal{T}_\Delta$, we define $\Sigma(\mathcal{T}) \subseteq \mathcal{T}_{\Sigma \cup \Delta}$ to be the set of trees $a(t_1, \ldots, t_k)$ such that $k \in \mathbb{N}$, $a \in \Sigma^{(k)}$, and $t_1, \ldots, t_k \in \mathcal{T}$, and we define $\mathcal{T}_\Sigma(\mathcal{T}) \subseteq \mathcal{T}_{\Sigma \cup \Delta}$ to be the smallest set of trees $\mathcal{T}'$ such that $\mathcal{T} \cup \Sigma(\mathcal{T}') \subseteq \mathcal{T}'$. Note that $\mathcal{T}_\Sigma(\emptyset) = \mathcal{T}_\Sigma$.

A Σ-*pattern* is an upper portion, or prefix, of a tree in $\mathcal{T}_\Sigma$. Formally the set $\mathcal{P}_\Sigma$ of Σ-patterns is defined to be the set of trees $\mathcal{T}_\Sigma(\{\bot\})$, where $\bot$ is a new symbol of rank zero that is not in Σ. If t_0 is a pattern containing exactly k occurrences of $\bot$, and $t_1, \ldots, t_k$ is a sequence of k patterns, 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. A Σ-*context* is a Σ-pattern that contains exactly one occurrence of $\bot$. The set of all Σ-contexts is denoted $\mathcal{C}_\Sigma$. Thus, for $C \in \mathcal{C}_\Sigma$ and $t \in \mathcal{T}_\Sigma$, the tree $C[t] \in \mathcal{T}_\Sigma$ is obtained from the context C by replacing the unique occurrence of $\bot$ in C by t.

On the set $\mathcal{P}_\Sigma$ we define a partial order $\sqsubseteq$ as follows: for patterns t and t' in $\mathcal{P}_\Sigma$, t' is a *prefix* of t, denoted $t' \sqsubseteq t$, if $t = t'[t_1, \ldots, t_k]$ for suitable patterns $t_1, \ldots, t_k$; equivalently, $V_a(t') \subseteq V_a(t)$ for every $a \in \Sigma$. Obviously, $\bot \sqsubseteq t$ for every pattern t. We note that in [9] the inverse of the partial order $\sqsubseteq$ is used. Every nonempty set Π of Σ-patterns has a greatest lower bound $\sqcap \Pi$ in $\mathcal{P}_\Sigma$, 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 $a \in \Sigma$, $v \in V_a(t')$ if and only if (1) $v \in V_a(t)$ for every $t \in \Pi$ and (2) every proper ancestor of v is in $V(t')$. This implies the following easy lemma.

Lemma 1. *Let Π be a nonempty subset of $\mathcal{T}_\Sigma$, and let $v \in \mathbb{N}_+^*$. Then, $v \in V_\bot(\sqcap \Pi)$ if and only if*

(1) $v \in V(t)$ for every $t \in \Pi$,
(2) $\mathsf{lab}(t_1, \hat{v}) = \mathsf{lab}(t_2, \hat{v})$ for every proper ancestor $\hat{v}$ of v and all $t_1, t_2 \in \Pi$, and
(3) there exist $t_1, t_2 \in \Pi$ such that $\mathsf{lab}(t_1, v) \neq \mathsf{lab}(t_2, v)$.

For instance, $\sqcap\{\sigma(\tau(a), b), \sigma(b, b)\} = \sigma(\tau(a), b) \sqcap \sigma(b, b) = \sigma(\bot, b)$.

For $t, t' \in \mathcal{T}_\Sigma$ and $v \in V(t)$, we denote by $t[v \leftarrow t']$ the tree that is obtained from t by replacing its subtree t/v by t'. More precisely, if C is the unique context in $\mathcal{C}_\Sigma$ such that $C \sqsubseteq t$ and $C/v = \bot$, then $t[v \leftarrow t'] = C[t']$.

Let $\mathcal{S}$ be a subset of $\mathcal{T}_\Sigma$ such that no $s \in \mathcal{S}$ is a subtree of $s' \in \mathcal{S}$ with $s \neq s'$. For a tree $t \in \mathcal{T}_\Sigma$ and a partial function $\psi : \mathcal{S} \to \mathcal{T}_\Sigma$, 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 $s \in \mathcal{S}$. More precisely, $t[s \leftarrow \psi(s) \mid s \in \mathcal{S}] = t[v_1 \leftarrow \psi(t/v_1)] \cdots [v_k \leftarrow \psi(t/v_k)]$ where $\{v_1, \ldots, v_k\} = \{v \in V(t) \mid t/v \in \mathcal{S}\}$. Note that v_i is not an ancestor of v_j, for $i \neq j$, and hence the order of the substitutions $[v_i \leftarrow \psi(t/v_i)]$ is irrelevant. Note also that $t[s \leftarrow \psi(s) \mid s \in \mathcal{S}]$ is defined if and only if $\psi(t/v_i)$ is defined for every $i \in [k]$.

To formulate the rules of top-down tree transducers, we use variables x_i, with $i \in \mathbb{N}$, which are assumed to have rank 0. The set $\{x_0, x_1, x_2, \ldots\}$ of all such variables is denoted X. For $k \in \mathbb{N}$, we denote $\{x_1, \ldots, x_k\}$ by X_k; note that $X_0 = \emptyset$.

3 Deterministic Top-Down Tree Transducers

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, respectively, and P is a finite nonempty set of look-ahead states. The function A maps look-ahead states to trees in $\mathcal{T}_\Delta(Q(\{x_0\}))$; for $p \in P$, the tree $A(p)$ is called the p-axiom of M. The finite set R provides at most one rule

$$q(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta$$

for every state q, every input symbol a of rank $k \geq 0$ and every sequence $p_1, \ldots, p_k$ of look-ahead states. The right-hand side ζ of the rule is a tree in $\mathcal{T}_\Delta(Q(X_k))$, i.e., $\zeta = t[q_1(x_{i_1}), \ldots, q_r(x_{i_r})]$ for some pattern $t \in \mathcal{P}_\Delta$, $r = |V_\perp(t)|$, $q_j \in Q$, and $x_{i_j} \in \{x_1, \ldots, x_k\}$ for $j \in [r]$; we will denote ζ also 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). That means that $\delta(a, p_1, \ldots, p_k) \in P$ for every $k \geq 0$, $a \in \Sigma^{(k)}$, and $p_1, \ldots, p_k \in P$.

Examples of dtlas are given in the next section. Whenever we consider a dtla with the name M, it will be understood that its components are named $(Q, \Sigma, \Delta, R, A, P, \delta)$. When necessary we provide the components of a dtla M with the subscript M. Then we have $Q_M, \Sigma_M, \Delta_M, R_M, \mathsf{rhs}_M$, etc. We denote by $\mathsf{maxrhs}(M)$ the maximal height of the axioms and the right-hand sides of the rules of M.

We now define the semantics of the dtla M, starting with the semantics of its look-ahead automaton (P, δ). The transition function δ gives rise to a function δ^* that maps $\mathcal{T}_\Sigma$ to P. It is defined by $\delta^*(a(s_1, \ldots, s_k)) = \delta(a, \delta^*(s_1), \ldots, \delta^*(s_k))$ for $a \in \Sigma^{(k)}$ and $s_1, \ldots, s_k \in \mathcal{T}_\Sigma$. For convenience, we denote the function δ^* by δ as well. For $p \in P$ we denote by $[\![p]\!]_M$ the set of trees $s \in \mathcal{T}_\Sigma$ that have look-ahead state p, i.e., $\delta(s) = p$; we drop the subscript M from $[\![p]\!]_M$ whenever it is clear from the context. Note that $\{[\![p]\!] \mid p \in P\}$ is a partition of $\mathcal{T}_\Sigma$. For a node u of an input tree $s \in \mathcal{T}_\Sigma$, we also say that $\delta(s/u)$ is the look-ahead state at u.

For $q \in Q$, $s \in \mathcal{T}_\Sigma$, and $u \in V(s)$, we define $\mathsf{rhs}(q, s, u) = \mathsf{rhs}(q, a, p_1, \ldots, p_k)$ where $\mathsf{lab}(s, u) = a \in \Sigma^{(k)}$ and $p_i = \delta(s/ui)$ for every $i \in [k]$. Intuitively, $\mathsf{rhs}(q, s, u)$ is the right-hand side of the rule that is applied when M arrives at node u in state q (if that rule exists); it is uniquely determined by the label of u and the look-ahead states at its children.

A *sentential form* of M for $s \in \mathcal{T}_\Sigma$ is a tree in $\mathcal{T}_\Delta(Q(V(s)))$, where the nodes in $V(s)$ are viewed as symbols of rank 0. For sentential forms ξ, ξ' we write $\xi \Rightarrow_s \xi'$ if there exist $v \in V(\xi)$, $q \in Q$, and $u \in V(s)$ such that

$$\xi/v = q(u) \text{ and } \xi' = \xi[v \leftarrow \mathsf{rhs}(q, s, u)[x_i \leftarrow ui \mid i \in \mathbb{N}_+]].$$

This will be called a computation step of M in state q at nodes u and v. It is easy to see that the rewriting in computations is confluent (i.e., if $\xi \Rightarrow_s^* \xi_1$ and $\xi \Rightarrow_s^* \xi_2$, then there exists a sentential form $\bar{\xi}$ such that $\xi_1 \Rightarrow_s^* \bar{\xi}$ and $\xi_2 \Rightarrow_s^* \bar{\xi}$). Hence, if $\xi \Rightarrow_s^* t \in \mathcal{T}_\Delta$ and $\xi \Rightarrow_s^* \xi'$, then $\xi' \Rightarrow_s^* t$; thus, computations that start with a given sentential form lead to a unique tree in $\mathcal{T}_\Delta$ (if it exists).

The dtla M *realizes* a partial function $[\![M]\!] : \mathcal{T}_\Sigma \to \mathcal{T}_\Delta$, called its *translation*. Let $s \in \mathcal{T}_\Sigma$ and $\delta(s) = p \in P$. The output tree $[\![M]\!](s)$ of the transducer M for the input tree s is the unique tree $t \in \mathcal{T}_\Delta$ such that $A(p)[x_0 \leftarrow \varepsilon] \Rightarrow_s^* t$ (if it exists). For readability, we will write $M(s)$ instead of $[\![M]\!](s)$.

Two dtlas M_1 and M_2 are *equivalent* if they realize the same translation, i.e., if $\Sigma_{M_1} = \Sigma_{M_2}$, $\Delta_{M_1} = \Delta_{M_2}$, and $[\![M_1]\!] = [\![M_2]\!]$.

Intuitively, a sentential form ξ consists of output that has already been produced by M; moreover, $\xi/v = q(u)$ means that M has arrived at node u of the input tree s in state q and, starting in that state, will translate the input subtree s/u into the output subtree $M(s)/v$. Note that several parallel copies of M can arrive at u for different nodes of ξ, i.e., there may exist nodes $v' \neq v$ such that $\xi/v' = q'(u)$, where q' may also be equal to q.

A sentential form ξ for s is *reachable* if $A(p)[x_0 \leftarrow \varepsilon] \Rightarrow_s^* \xi$ where $p = \delta(s)$. Thus, if $M(s)$ is defined and ξ is a reachable sentential form for s, then $\xi \Rightarrow_s^* M(s)$.

We also define the semantics of every state q of M as a partial function $[\![q]\!]_M :$ $\mathcal{T}_\Sigma \to \mathcal{T}_\Delta$ as follows. For $s \in \mathcal{T}_\Sigma$, $[\![q]\!]_M(s)$ is the unique tree $t \in \mathcal{T}_\Delta$ such that $q(\varepsilon) \Rightarrow_s^* t$ (if it exists). For readability, we will write $q_M(s)$ instead of $[\![q]\!]_M(s)$.

The following lemma is easy to prove.

Lemma 2. *Let $s \in \mathcal{T}_\Sigma$ and $t \in \mathcal{T}_\Delta$.*

(1) For every $q \in Q$ and $u \in V(s)$,

$$q(u) \Rightarrow_s^* t \ \ \text{if and only if} \ \ q_M(s/u) = t.$$

(2) For every sentential form ξ for s,

$$\xi \Rightarrow_s^* t \ \ \text{if and only if} \ \ t = \xi[q(u) \leftarrow q_M(s/u) \mid q \in Q, u \in V(s)].$$

(3) If $\delta(s) = p$, then

$$M(s) = A(p)[q(x_0) \leftarrow q_M(s) \mid q \in Q].$$

(4) For every $\bar{q} \in Q$, if $s = a(s_1, \ldots, s_k)$, then

$$\bar{q}_M(s) = \mathsf{rhs}(\bar{q}, a, \delta(s_1), \ldots, \delta(s_k))[q(x_i) \leftarrow q_M(s_i) \mid q \in Q, \, i \in [k]].$$

Proof. (1) follows from the obvious bijection between the nodes of s/u and the nodes of s with prefix u (i.e., the descendants of u in s).

(2) is obvious from (1) and the fact that the computation steps $\xi \Rightarrow_s \xi'$ of M are context-free. In fact, the computations of M on s can be viewed as derivations of a context-free grammar with the set of nonterminals $Q(V(s))$ and with rules $q(u) \to$ $\mathsf{rhs}(q, s, u)[x_i \leftarrow ui \mid i \in \mathbb{N}_+]$.

(3) and (4) follow from (2), taking $\xi = A(p)[x_0 \leftarrow \varepsilon]$ and $\xi = \mathsf{rhs}(\bar{q}, s, \varepsilon)[x_i \leftarrow i \mid i \in [k]]$, respectively. $\square$

Note that (3) and (4) of Lemma 2 form an alternative way of defining the semantics of M (recursively).

Convention. For a given dtla M it can be assumed that it is *reduced*, i.e., that all its states and look-ahead states are *reachable* in the following sense. A look-ahead state p is reachable if $[\![p]\!]_M \neq \emptyset$. A state q is reachable if q occurs in an axiom, or if q occurs in the right-hand side of a rule of which the left-hand side starts with a reachable state. In what follows we will assume all dtlas to be reduced. However, all properties that we will define, also apply to non-reduced dtlas. Whenever we construct a dtla and show

that it has certain properties, we allow the dtla to be non-reduced, but we will ensure that those properties are preserved under reduction (i.e., removing unreachable states and look-ahead states and the rules in which they occur). However, that will not be mentioned explicitly. □

A *deterministic top-down tree transducer* (*dtop* for short) is a dtla M with trivial look-ahead automaton (P, δ), i.e., P is a singleton. Whenever convenient, we drop (P, δ) from the tuple defining M, we identify A with the unique axiom $A(p)$, we write a rule as $q(a(x_1, \ldots, x_k)) \to \zeta$ rather than $q(a(x_1:p, \ldots, x_k:p)) \to \zeta$ (where p is the unique look-ahead state of M) and we 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 $|P| \geq 2$. Obviously, to decide whether M is equivalent to a dtop, we may assume that M is proper.

A dtla M is *total* if $\mathsf{dom}(\llbracket M \rrbracket) = \mathcal{T}_\Sigma$, i.e., if its translation $\llbracket M \rrbracket$ is a total function. Note that it is decidable whether M is total, because $\mathsf{dom}(\llbracket M \rrbracket)$ is effectively a regular tree language (cf. [6, Corollary 2.7]). From now on we mostly consider total dtlas.

A dtla M is *complete* if $\mathsf{rhs}(q, a, p_1, \ldots, p_k)$ is defined for every $q \in Q$, $k \in \mathbb{N}$, $a \in \Sigma^{(k)}$, and $p_1, \ldots, p_k \in P$. By Lemma 2(4), this means that $q_M(s)$ is defined for every $q \in Q$ and $s \in \mathcal{T}_\Sigma$. Thus, by Lemma 2(3), if M is complete, then M is total.

A dtla M is *linear* if (1) for every $p \in P$, the variable x_0 occurs at most once in $A(p)$, and (2) for every rule $q(a(x_1:p_1, \ldots, x_k:p_k)) \to \zeta$, each variable x_i occurs at most once in ζ.

A dtla M is *ultralinear* if there is a mapping $\mu : Q \to \mathbb{N}$ such that for every rule $q(a(x_1:p_1, \ldots, x_k:p_k)) \to \zeta$ the following two properties hold for every $\bar{q}(x_i)$ that occurs in ζ: (1) $\mu(\bar{q}) \geq \mu(q)$, and (2) if $\mu(\bar{q}) = \mu(q)$, then x_i occurs only once in ζ. Obviously, every linear dtla is ultralinear. We note that ultra-linearity was first defined for context-free grammars, in [17].

A dtla M is *nonerasing* if it does not have erasing rules. A rule of M is an *erasing rule* if its right-hand side is in $Q(X)$, i.e., contains no symbols from Δ.

A dtla M is *bounded erasing* (for short, *b-erasing*) if there is no cycle in the directed graph E_M with the set of nodes Q and an edge from q to q' if there is an erasing rule of the form $q(a(x_1:p_1, \ldots, x_k:p_k)) \to q'(x_j)$. Obviously, every nonerasing dtla is b-erasing.

Note that all the above properties, except properness, are preserved under reduction.

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 will consider input trees $\bar{s}$ with occurrences of $p \in P$, viewed as input symbol of rank zero, representing an absent subtree s with $\delta(s) = p$. If M arrives in state q at a p-labeled leaf of $\bar{s}$, then M will output the new symbol $\langle q, p \rangle$ of rank zero, representing the absent output tree $q_M(s)$. In this way, M translates input trees $\bar{s} \in \mathcal{T}_\Sigma(P)$ to output trees in $\mathcal{T}_\Delta(Q \times P)$. Without loss of generality we assume that P and Σ are disjoint, and so are $Q \times P$ and Δ.

Formally, we extend M to a dtla $M^\circ = (Q, \Sigma^\circ, \Delta^\circ, R^\circ, A, P, \delta^\circ)$ where $\Sigma^\circ = \Sigma \cup P$ such that every element of P has rank zero, $\Delta^\circ = \Delta \cup (Q \times P)$ such that every element of $Q \times P$ has rank zero, R° is obtained from R by adding the rules

$q(p) \to \langle q, p \rangle$ for all $q \in Q$ and $p \in P$ such that $q_M(s)$ is defined for some $s \in [\![p]\!]_M$, and δ° is the extension of δ such that $\delta^\circ(p) = p$ 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 input tree $\bar{s} \in \mathcal{T}_\Sigma(P)$. But note that we do *not* drop $^\circ$ from $[\![p]\!]_{M^\circ}$, $[\![M^\circ]\!]$, and $[\![q]\!]_{M^\circ}$, i.e., $[\![p]\!]_M$, $[\![M]\!]$, and $[\![q]\!]_M$ keep their meaning.

We will use the following elementary lemma, which expresses the above intuition.

Lemma 3. *Let M be a total dtla. Let $\bar{s}$ be a tree in $\mathcal{T}_\Sigma(P)$, and for every $p \in P$, let s_p be a tree in $\mathcal{T}_\Sigma(P)$ such that $\delta(s_p) = p$. Then $\delta(\bar{s}[p \leftarrow s_p \mid p \in P]) = \delta(\bar{s})$ and*

$$M(\bar{s}[p \leftarrow s_p \mid p \in P]) = M(\bar{s})\big[\langle q, p \rangle \leftarrow q_M(s_p) \mid q \in Q, p \in P\big]. \qquad (1)$$

Proof. Let $s' = \bar{s}[p \leftarrow s_p \mid p \in P]$. It should be clear that $\delta(s'/v) = \delta(\bar{s}/v)$ for every node v of $\bar{s}$. Let $p' = \delta(s') = \delta(\bar{s})$.

We first assume that $s_p \in \mathcal{T}_\Sigma$ for every $p \in P$. Thus $s' \in \mathcal{T}_\Sigma$, and so $M(s')$ is defined. Let U be the set of nodes u of $\bar{s}$ such that $\mathsf{lab}(\bar{s}, u) \in P$, and for $u \in U$, let $p_u = \mathsf{lab}(\bar{s}, u)$. Now consider a computation $A(p')[x_0 \leftarrow \varepsilon] \Rightarrow^*_{s'} \xi$ of M such that $\xi \in \mathcal{T}_\Delta(Q(U))$ and none of the computation steps is at a node $u \in U$ (and hence not at a descendant of u). Then, by the observation above, also $A(p')[x_0 \leftarrow \varepsilon] \Rightarrow^*_{\bar{s}} \xi$. By Lemma 2(2), $M(s') = \xi[q(u) \leftarrow q_M(s_{p_u}) \mid q \in Q, u \in U]$. Thus, $q_M(s_{p_u})$ is defined for every $q(u)$ that occurs in ξ, and so $q(p_u) \to \langle q, p_u \rangle$ is a rule of M°. Hence $\xi \Rightarrow^*_{\bar{s}} \xi[q(u) \leftarrow \langle q, p_u \rangle \mid q \in Q, u \in U]$ and so $M(\bar{s}) = \xi[q(u) \leftarrow \langle q, p_u \rangle \mid q \in Q, u \in U]$. This proves Equation (1) for the case where $s_p \in \mathcal{T}_\Sigma$. It also implies that M° is total, because for every $\bar{s} \in \mathcal{T}_\Sigma(P)$ one can choose some $s_p \in [\![p]\!]_M$ for each $p \in P$, and then $M(\bar{s}[p \leftarrow s_p \mid p \in P])$ is defined and hence so is $M(\bar{s})$. Since M° is total, the previous argument also proves Equation (1) for the general case where $s_p \in \mathcal{T}_\Sigma(P)$. $\qquad \square$

Note that the proof of Lemma 3 shows that if M is total, then so is M°. From Lemma 3 we immediately obtain the next lemma for Σ-contexts. Note that for every $C \in \mathcal{C}_\Sigma$ and $p \in P$, the tree $M(C[p])$ is in $\mathcal{T}_\Delta(Q \times \{p\})$.

Lemma 4. *Let M be a total 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 $M(C[s]) = M(C[p])\big[\langle q, p \rangle \leftarrow q_M(s) \mid q \in Q\big]$.*

Proof. Apply Lemma 3 with $\bar{s} = C[p]$ and $s_p = s$. $\qquad \square$

In Section 6 the next lemma will be needed.

Lemma 5. *Let M be a total dtop. Then M is complete, and for every $q \in Q$ there exists $C \in \mathcal{C}_\Sigma$ such that $\langle q, p \rangle$ occurs in $M(C[p])$, where $P = \{p\}$.*

Proof. It is convenient to assume that $p = \bot$ (and so $C[p] = C$). We first prove the second statement. Since we assume, by convention, that every state q of M is reachable, we proceed by induction on the definition of reachability. If $q(x_0)$ occurs in the axiom A of M, then $C = \bot$ satisfies the requirement because $M(\bot) = A[\bar{q}(x_0) \leftarrow \langle \bar{q}, \bot \rangle \mid \bar{q} \in Q]$. Now let $q(a(x_1, \ldots, x_k)) \to \zeta$ be a rule of M such that q is reachable, and let $\zeta/z = q'(x_j)$. By induction there exist $C \in \mathcal{C}_\Sigma$ and $v \in V(M(C))$ such that $M(C)/v = \langle q, \bot \rangle$. Let $s_j = \bot$ and choose $s_m \in \mathcal{T}_\Sigma$ for $m \in [k] - \{j\}$. Let C' be the Σ-context

10

$C[a(s_1, \ldots, s_k)]$. By Lemma 4, $M(C') = M(C)[\langle \bar{q}, \perp \rangle \leftarrow \bar{q}_M(a(s_1, \ldots, s_k)) \mid \bar{q} \in Q]$ and by Lemma 2(4), $q_M(a(s_1, \ldots, s_k)) = \zeta[\bar{q}(x_i) \leftarrow \bar{q}_M(s_i) \mid \bar{q} \in Q, i \in [k]]$. Then $M(C')/vz = q_M(a(s_1, \ldots, s_k))/z = q'_M(s_j) = \langle q', \perp \rangle$. So, $\langle q', \perp \rangle$ occurs in $M(C')$.

To show that M is complete, let $q \in Q$ and $a \in \Sigma^{(k)}$. We have shown that there exist $C \in \mathcal{C}_\Sigma$ and $v \in V(M(C))$ such that $M(C)/v = \langle q, \perp \rangle$. Let C' be as above. Then $M(C')/v = q_M(a(s_1, \ldots, s_k))$. Hence $\mathsf{rhs}(q, a)$ is defined by Lemma 2(4). □

4 Difference Trees

Let M be a total dtla. We wish to decide whether 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 the output of M on input $C[p]$ and its output on input $C[p']$, see also Lemma 4. 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 $M(C[p'])$. Recall that $M(C[p])$ denotes $M^\circ(C[p])$, which is defined because M° is total (as shown in the proof of Lemma 3). Let v be a node of $M(C[p]) \sqcap M(C[p'])$ with label $\perp$. 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 (as will be proved in Corollary 28). 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 $\mathsf{lubdiff}(M) \in \mathbb{N} \cup \{\infty\}$ to be the least upper bound of the heights of all difference trees of M, i.e.,

$$\mathsf{lubdiff}(M) = \mathsf{lub}\{\mathsf{ht}(t) \mid t \in \mathsf{diff}(M)\}$$

(for the definition of lub see Section 2). Intuitively, $\mathsf{lubdiff}(M)$ gives a measure of how much the transducer M makes use of its look-ahead information. Clearly, $\mathsf{lubdiff}(M)$ is finite (i.e., is in $\mathbb{N}$) if and only if $\mathsf{diff}(M)$ is finite. We will say that a number $h(M) \in \mathbb{N}$ is a *difference bound* for M if the following holds: if $\mathsf{diff}(M)$ is finite, then $\mathsf{lubdiff}(M) \leq h(M)$. Thus, if $\mathsf{diff}(M)$ is infinite, then any natural number is a difference bound for M; if $\mathsf{diff}(M)$ is finite, then any upper bound of the heights of the (finitely many) difference trees of M is a difference bound for M.

Our first main result (Theorem 50) 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. Our second main result (Theorem 85) is that a difference bound can be computed for every total dtla M that is ultralinear and b-erasing.

A node $v \in V_\perp(M(C[p]) \sqcap M(C[p']))$ will be called a *difference node* of $M(C[p])$ and $M(C[p'])$. It is characterized in the next lemma, which follows immediately from Lemma 1.

Lemma 6. *Let $C \in \mathcal{C}_\Sigma$, $p, p' \in P$, and $v \in \mathbb{N}_+^*$. Then,*
v is a difference node of $M(C[p])$ and $M(C[p'])$ if and only if

(1) $v \in V(M(C[p])) \cap V(M(C[p']))$,
(2) $\mathsf{lab}(M(C[p]), \hat{v}) = \mathsf{lab}(M(C[p']), \hat{v})$ for every proper ancestor $\hat{v}$ of v, and
(3) $\mathsf{lab}(M(C[p]), v) \neq \mathsf{lab}(M(C[p']), v)$.

Note that if v is a difference node of $M(C[p])$ and $M(C[p'])$, then $p \neq p'$. Thus, in the definition of $\mathsf{diff}(M)$ we can assume that $p \neq p'$. Hence, if M is a dtop then $\mathsf{diff}(M) = \emptyset$ and so $\mathsf{lubdiff}(M) = 0$. Note also that, to compute $\mathsf{lubdiff}(M)$ for a dtla M, it suffices to consider difference trees of non-zero height, i.e., difference nodes v that are not leaves of $M(C[p])$.

We now give some examples of dtlas with their sets of difference trees.

Example 7. Let $\Sigma = \Delta = \{\sigma^{(1)}, a^{(0)}, b^{(0)}\}$, which means that Σ and Δ are the ranked alphabet $\{\sigma, a, b\}$ with $\mathsf{rk}(\sigma) = 1$ and $\mathsf{rk}(a) = \mathsf{rk}(b) = 0$. We consider the following total dtla $M = (Q, \Sigma, \Delta, R, A, P, \delta)$ with $M(\sigma^n a) = a$ 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 two axioms are $A(p_a) = a$ and $A(p_b) = q(x_0)$, and its set R of rules contains the two rules $q(\sigma(x_1 : p_b)) \to \sigma(q(x_1))$ and $q(b) \to b$.

Clearly, $\mathcal{C}_\Sigma = \{\sigma^n \perp \mid n \in \mathbb{N}\}$ and for $C = \sigma^n \perp$ we have $M(C[p_a]) = a$ 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 difference node of $M(C[p_a])$ and $M(C[p_b])$ is ε, and we obtain the difference trees $M(C[p_a])$ and $M(C[p_b])$. Hence, $\mathsf{diff}(M) = \{a\} \cup \{\sigma^n \langle q, p_b \rangle \mid n \in \mathbb{N}\}$ and $\mathsf{lubdiff}(M) = \infty$. Since $\mathsf{diff}(M)$ is infinite, M is not equivalent to a dtop, as will be shown in Corollary 28. $\square$

Example 8. Let $\Sigma = \Delta = \{\sigma^{(1)}, \tau^{(1)}, a^{(0)}, b^{(0)}\}$ and consider the following total dtla M. For an input tree s with leaf a, M outputs the top-most 80 unary symbols of s and the leaf a if $\mathsf{size}(s) > 80$, and it outputs s if $\mathsf{size}(s) \leq 80$. The same is true for an input tree with leaf b, with 30 unary symbols instead of 80. The look-ahead automaton of M is similar to the one of the previous example: $P = \{p_a, p_b\}$ with $\delta(a) = p_a$, $\delta(b) = p_b$, and $\delta(\gamma, p) = p$ for $\gamma \in \{\sigma, \tau\}$ and $p \in P$. The set of states is $Q = \{q_i^a \mid i \in [80]\} \cup \{q_i^b \mid i \in [30]\}$. The axioms are $A(p_a) = q_{80}^a(x_0)$ and $A(p_b) = q_{30}^b(x_0)$. For the states with superscript a, M has the rules $q_i^a(a) \to a$ for $i \in [80]$, and for $\gamma \in \{\sigma, \tau\}$, the rules $q_i^a(\gamma(x_1 : p_a)) \to \gamma(q_{i-1}^a(x_1))$ for $1 < i \leq 80$, and $q_1^a(\gamma(x_1 : p_a)) \to \gamma(a)$. The rules for the states with superscript b are similar (in the obvious way, with b instead of a and 30 instead of 80).

Every Σ-context C is of the form $C = w(\perp)$ with $w \in \{\sigma, \tau\}^*$, cf. Section 2 for this notation. For such a context we have $M(C[p_a]) = w(\langle q_{80-|w|}^a, p_a \rangle)$ if $|w| < 80$, and $M(C[p_a]) = w_{[80]}(a)$ if $|w| \geq 80$, where $w_{[80]}$ is the prefix of w of length 80. A similar statement holds for $M(C[p_b])$. Thus, for $|w| < 30$ we obtain that $M(C[p_a]) =$

$w(\langle q^a_{80-|w|}, p_a \rangle)$ and $M(C[p_b]) = w(\langle q^b_{30-|w|}, p_b \rangle)$, hence $M(C[p_a]) \sqcap M(C[p_b]) = w(\bot)$, which gives the difference trees $\langle q^a_{i+50}, p_a \rangle$ and $\langle q^b_i, p_b \rangle$, where $i = 30-|w|$. For $w = w_1 z$ with $|w_1| = 30$ and $|z| < 50$, we get that $M(C[p_a]) = w_1 z(\langle q^a_{80-|w_1 z|}, p_a \rangle)$ and $M(C[p_b]) = w_1(b)$, hence $M(C[p_a]) \sqcap M(C[p_b]) = w_1(\bot)$, with the difference trees $z(\langle q^a_{50-|z|}, p_a \rangle)$ and b. Finally, if $w = w_1 z w_2$ with $|w_1| = 30$ and $|z| = 50$, then $M(C[p_a]) = w_1 z(a)$ and $M(C[p_b]) = w_1(b)$, hence again $M(C[p_a]) \sqcap M(C[p_b]) = w_1(\bot)$, now with the difference trees $z(a)$ and (again) b. Hence, $\mathsf{lubdiff}(M) = 50$.

It is not difficult to see that there exists a dtop N equivalent to M. It outputs the top-most 30 symbols, and then it stores in its state the next 50 (or less) unary symbols of the input tree s, and depending on the leaf label of s, it outputs these symbols and a or it outputs b. Formally, its set of states is $Q_N = Q_{N1} \cup Q_{N2}$ with $Q_{N1} = \{q_i \mid i \in [30]\}$ and $Q_{N2} = \{q_z \mid z \in \{\sigma, \tau\}^*, |z| \leq 50\}$, and its axiom is $q_{30}(x_0)$. The rules of N are the following, for every $q_i \in Q_{N1}$, $q_z \in Q_{N2}$, and $\gamma \in \{\sigma, \tau\}$. First, the rules $q_i(\gamma(x_1)) \to \gamma(q_{i-1}(x_1))$ for $i \neq 1$, and $q_1(\gamma(x_1)) \to \gamma(q_\varepsilon(x_1))$. Second, $q_z(\gamma(x_1)) \to q_{z\gamma}(x_1)$ for $|z| < 50$, and $q_z(\gamma(x_1)) \to q_z(x_1)$ for $|z| = 50$. Third, and finally, $q_i(a) \to a$, $q_i(b) \to b$, $q_z(a) \to z(a)$, and $q_z(b) \to b$. $\qquad\qquad \square$

Example 9. Let $\Sigma = \{\sigma^{(2)}, aa^{(0)}, ab^{(0)}, ba^{(0)}, bb^{(0)}\}$ where we view aa, ab, ba, and bb as symbols, and let $\Delta = \{\sigma^{(3)}, \#^{(2)}, a^{(0)}, b^{(0)}\} \cup \Sigma^{(0)}$. We consider the following total dtla M such that $M(aa) = aa$, $M(ab) = ab$, $M(ba) = ba$, $M(bb) = bb$, and for every $s_1, s_2 \in \mathcal{T}_\Sigma$, $M(\sigma(s_1, s_2)) = \sigma(M(s_1), M(s_2), \#(y, z))$ 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. Its look-ahead automaton has four 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 $C/u = \bot$. It is easy to see that the difference nodes of $M(C[p_{aa}])$ and $M(C[p_{ba}])$ 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 $M(C[p_{ba}])/v \cdot (3, 1) = b$. In this way we obtain that $\mathsf{diff}(M) = \{a, b\} \cup \{\langle q, p_{yz} \rangle \mid y, z \in \{a, b\}\}$. Thus, $\mathsf{lubdiff}(M) = 0$.

Clearly, there is a dtop N equivalent to M. It has three states q, q_1, q_2, axiom $q(x_0)$, and rules $q(yz) \to yz$,

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

$q_i(\sigma(x_1, x_2)) \to q_i(x_i)$ for $i = 1, 2$, $q_1(yz) \to y$, and $q_2(yz) \to z$ for $y, z \in \{a, b\}$. $\qquad\qquad \square$

Example 10. Let $\Sigma = \{\sigma^{(2)}, a^{(0)}\}$ and $\Delta = \{e^{(0)}, o^{(0)}\}$, and consider the following total dtla M that translates every tree $s \in \mathcal{T}_\Sigma$ into e if $\mathsf{size}(s)$ is even and into o if it is odd. Its look-ahead automaton has two states p_e and p_o with $\delta(a) = p_o$, $\delta(\sigma, p_e, p_e) = $

$\delta(\sigma, p_o, p_o) = p_e$, and $\delta(\sigma, p_e, p_o) = \delta(\sigma, p_o, p_e) = p_o$. Its set of states is empty, and its axioms are $A(p_e) = e$ and $A(p_o) = o$.

For every Σ-context C, $\{M(C[p_e]), M(C[p_o])\} = \{e, o\}$. Hence $\mathsf{diff}(M) = \{e, o\}$ and $\mathsf{lubdiff}(M) = 0$. Although $\mathsf{diff}(M)$ is finite, there is obviously no dtop equivalent to M. □

5   Normal Forms

In this section we prove normal forms for total dtlas. For each of these normal forms we consider its effect on $\mathsf{lubdiff}(M)$. We start with a simple normal form in which each axiom consists of one state, more precisely, is in $Q(\{x_0\})$.

A dtla M is *initialized* if for every $p \in P$ there is a state $q_{0,p}$ such that $A(p) = q_{0,p}(x_0)$. The states $q_{0,p}$ are called initial states; they are not necessarily distinct. Note that for an initialized dtla, $M(s) = q_M(s)$ where $q = q_{0,p}$ and $p = \delta(s)$, for every $s \in \mathcal{T}_\Sigma$. The dtlas and dtops in Examples 8 and 9 are initialized. The dtlas in Examples 7 and 10 are not.

Recall that $\mathsf{maxrhs}(M)$ is the maximal height of the axioms and the right-hand sides of the rules of M.

Lemma 11. *For every total dtla M an equivalent initialized dtla M' can be constructed, with the same look-ahead automaton as M, such that*
$|Q_{M'}| = |Q_M| + 1$, $\mathsf{maxrhs}(M') \leq 2 \cdot \mathsf{maxrhs}(M)$, *and*

$$\mathsf{lubdiff}(M') \leq \mathsf{lubdiff}(M) \leq \max\{\mathsf{lubdiff}(M'), \mathsf{maxrhs}(M)\}.$$

If M is ultralinear or b-erasing, then so is M'.

Proof. To construct M' from M, we introduce a new state q_0. For every $a \in \Sigma^{(k)}$ and $p_1, \ldots, p_k \in P$ we add the rule

$$q_0(a(x_1 : p_1, \ldots, x_k : p_k)) \rightarrow A(p)[q(x_0) \leftarrow \mathsf{rhs}(q, a, p_1, \ldots, p_k) \mid q \in Q],$$

where $p = \delta(a, p_1, \ldots, p_k)$. The right-hand side of this rule is defined: since every look-ahead state is reachable, there exist trees s_i with $\delta(s_i) = p_i$, and so $\delta(a(s_1, \ldots, s_k)) = p$; since M is total, $M(a(s_1, \ldots, s_k))$ is defined and hence $\mathsf{rhs}(q, a, p_1, \ldots, p_k)$ is defined for every q that occurs in $A(p)$. After adding the above rules, we change $A(p)$ into $q_0(x_0)$ for every $p \in P$.

Then M' is initialized, with $q_{0,p} = q_0$ for all $p \in P$. It should be clear from Lemma 2(3/4) that M' is equivalent to M. It should also be clear that for every $C \in \mathcal{C}_\Sigma$ and $p \in P$, if $C \neq \bot$ then $M'(C[p]) = M(C[p])$. Moreover, for $C = \bot$ we have $M'(p) = \langle q_0, p \rangle$ and $M(p) = A(p)[q(x_0) \leftarrow \langle q, p \rangle \mid q \in Q]$. Since $\mathsf{ht}(M'(p)/\varepsilon) = 0$ and $\mathsf{ht}(M(p)/v) \leq \mathsf{ht}(A(p)) \leq \mathsf{maxrhs}(M)$ for every $v \in V(M(p))$, this implies the required inequalities for $\mathsf{lubdiff}(M)$ and $\mathsf{lubdiff}(M')$.

If M is ultralinear with mapping μ, then we can assume that $\mu(q) > 0$ for all q, and then M' is ultralinear by extending μ with $\mu(q_0) = 0$. If M is b-erasing then so is M', because a cycle in $E_{M'}$ does not contain q_0 and hence is also a cycle in E_M. □

Note that it follows from the inequalities for $\mathsf{lubdiff}(M)$ and $\mathsf{lubdiff}(M')$ that if $h(M')$ is a difference bound for M', then $\max\{h(M'), \mathsf{maxrhs}(M)\}$ is a difference bound for M. In fact, if $\mathsf{diff}(M)$ is finite, then $\mathsf{diff}(M')$ is finite because $\mathsf{lubdiff}(M') \leq \mathsf{lubdiff}(M)$, hence $\mathsf{lubdiff}(M') \leq h(M')$, from which it follows that $\mathsf{lubdiff}(M) \leq \max\{\mathsf{lubdiff}(M'), \mathsf{maxrhs}(M)\} \leq \max\{h(M'), \mathsf{maxrhs}(M)\}$.

We continue with a basic and technically convenient normal form in which every state of the dtla only translates input trees that have the same look-ahead state; moreover, the rules satisfy a generalized completeness condition. It is closely related to the uniform i-transducer in [9].

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:

(1) For every $p \in P$ and $q \in Q$, 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,
   (a) $\rho(q) = \delta(a, p_1, \ldots, p_k)$ and
   (b) for every $\bar{q} \in Q$ and $i \in [k]$, if $\bar{q}(x_i)$ occurs in ζ, then $\rho(\bar{q}) = p_i$.
(3) For every $q \in Q$, $a \in \Sigma^{(k)}$, 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.

Clearly, the dtla M of Example 7 is la-uniform with $\rho(q) = p_b$, and similarly, the one of Example 8 is la-uniform with $\rho(q_i^y) = p_y$ for $y \in \{a, b\}$. Note that a dtop is la-uniform if and only if it is complete (if and only if it is total, by Lemma 5). In general, an la-uniform dtla is not complete; in fact, it is easy to see that every complete la-uniform dtla is a dtop.

We will need the following straightforward properties of an la-uniform dtla.

Lemma 12. *Let M be an la-uniform dtla with la-map ρ.*

(1) $\mathsf{dom}(\llbracket q \rrbracket_M) = \llbracket \rho(q) \rrbracket_M$ *for every $q \in Q$.*
(2) M *is total.*
(3) M° *is la-uniform with the same la-map ρ as M.*
(4) *Let ξ be a reachable sentential form for $s \in \mathcal{T}_\Sigma$.*
    For all $v \in V(\xi)$, $q \in Q$, and $u \in V(s)$, if $\xi/v = q(u)$ then $\rho(q) = \delta(s/u)$.

Proof. (1) We prove by structural induction on $s \in \mathcal{T}_\Sigma$ that $q_M(s)$ is defined if and only if $\delta(s) = \rho(q)$. Let $s = a(s_1, \ldots, s_k)$ and $\delta(s_i) = p_i$ for $i \in [k]$. Then $\delta(s) = \delta(a, p_1, \ldots, p_k)$. Thus, by conditions (2)(a) and (3) above, $\delta(s) = \rho(q)$ if and only if there is a rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta$ in R. For such a rule, by condition (2)(b) above, if $\bar{q}(x_i)$ occurs in ζ, then $\rho(\bar{q}) = p_i$ and hence, by induction, $\bar{q}_M(s_i)$ is defined. It now follows from Lemma 2(4) that $\delta(s) = \rho(q)$ if and only if $q_M(s)$ is defined.

(2) This is immediate from (1) and Lemma 2(3), by condition (1) above.

(3) By (1), the set R° of rules of M° is obtained from R by adding all the rules $q(p) \to \langle q, p \rangle$ such that $\rho(q) = p$. Hence ρ also satisfies conditions (2) and (3) above for M° (and condition (1) above, because M° has the same axioms as M).

(4) The easy proof is by induction on the length of the computation $A(\delta(s))[x_0 \leftarrow \varepsilon] \Rightarrow_s^* \xi$, using conditions (1) and (2) above. $\square$

15

Note that by (3) of this lemma, for every $C \in \mathcal{C}_\Sigma$ and $p \in P$, the tree $M(C[p])$ is in $\mathcal{T}_\Delta(Q_p \times \{p\})$ where $Q_p = \{q \in Q \mid \rho(q) = p\}$.

We now prove that la-uniformity is a normal form for total dtlas.

Lemma 13. *For every total dtla M an equivalent la-uniform dtla M' can be constructed, with the same look-ahead automaton as M, such that $|Q_{M'}| = |Q_M| \cdot |P_M|$, $\mathsf{maxrhs}(M') = \mathsf{maxrhs}(M)$, and $\mathsf{lubdiff}(M') = \mathsf{lubdiff}(M)$. If M is initialized, ultralinear or b-erasing, then so is M'.*

Proof. We observe that it may be assumed that M is complete: if $\mathsf{rhs}(q, a, p_1, \ldots, p_k)$ is undefined, then we add the (dummy) rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \to d$ where d is any element of $\Delta^{(0)}$.

We construct M' as follows. The state set of M' is $Q_{M'} = Q \times P$. Every axiom $A(p)$ of M is changed into $A(p)[q(x_0) \leftarrow \langle q, p \rangle(x_0) \mid q \in Q]$, and every rule

$$q(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta$$

is changed into the rule

$$\langle q, p \rangle(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta[\bar{q}(x_i) \leftarrow \langle \bar{q}, p_i \rangle(x_i) \mid \bar{q} \in Q, \, i \in [k]]$$

where $p = \delta(a, p_1, \ldots, p_k)$.

It should be clear that M' satisfies conditions (1) and (2) of the definition of la-uniformity with la-map ρ such that $\rho(\langle q, p \rangle) = p$; since M is complete, condition (3) is also satisfied. Hence M' is la-uniform.

Since M and M' are total, so are M° and $(M')^\circ$ (see the proof of Lemma 3). Obviously, for every computation of $(M')^\circ$ on an input tree $\bar{s} \in \mathcal{T}_\Sigma(P)$, one obtains a computation of M° on that input tree by changing every $\langle q, p \rangle(u)$ that occurs in a sentential form into $q(u)$, and every $\langle \langle q, p \rangle, p \rangle$ into $\langle q, p \rangle$. Hence $M'(\bar{s}) = M(\bar{s})[\langle q, p \rangle \leftarrow \langle \langle q, p \rangle, p \rangle \mid q \in Q, p \in P]$. This implies that M' is equivalent to M. It also implies that $\mathsf{lubdiff}(M') = \mathsf{lubdiff}(M)$, as can easily be verified.

Obviously, if M is ultralinear with mapping $\mu_M : Q \to \mathbb{N}$, then so is M' with the mapping μ such that $\mu(\langle q, p \rangle) = \mu_M(q)$. Moreover, if there is an edge from $\langle q, p \rangle$ to $\langle q', p_j \rangle$ in $E_{M'}$, then there is an edge from q to q' in E_M. Hence, if M is b-erasing, then so is M'. $\qquad\square$

Note that since $\mathsf{lubdiff}(M') = \mathsf{lubdiff}(M)$, the la-uniform dtla M' has the same difference bounds as M.

Example 14. The dtla M of Example 9 is not la-uniform. We change it into an la-uniform dtla by the construction in the proof of Lemma 13 (but keep calling it M). Then it has set of states $Q = \{q_{yz} \mid y, z \in \{a, b\}\}$ where q_{yz} abbreviates $\langle q, p_{yz} \rangle$, so $\rho(q_{yz}) = p_{yz}$. Its axioms are $A(p_{yz}) = q_{yz}(x_0)$, and its rules are $q_{yz}(yz) \to yz$ 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 $w, x, y, z \in \{a, b\}$. $\qquad\square$

16

From now on we mainly consider la-uniform dtlas. For an la-uniform dtla M, its la-map will be denoted ρ (or ρ_M when necessary).

Finally we generalize the normal form for dtops in [9] to total dtlas. For this normal form it is essential that dtlas need not be initialized, i.e., that arbitrary axioms are allowed.

A dtla M is *earliest* if it is la-uniform and, for every state q of M, the set

$$\mathsf{rlabs}_M(q) := \{\mathsf{lab}(q_M(s), \varepsilon) \mid s \in \mathsf{dom}(\llbracket q \rrbracket_M)\} \subseteq \Delta$$

is not a singleton. This is equivalent with requiring that $\sqcap\{q_M(s) \mid s \in \mathsf{dom}(\llbracket q \rrbracket_M)\} = \bot$, cf. the definition of earliest in [9]. In other words, M is *not* earliest if it has a state q for which the roots of all output trees $q_M(s)$ have the same label; intuitively, the node with that label could be produced earlier in the computation of M.

A dtla M is *canonical* if it is earliest and $\llbracket q \rrbracket_M \neq \llbracket q' \rrbracket_M$ for all distinct states q, q' of M. Since it is required that M is la-uniform, the earliest and canonical properties are appropriately relativized with respect to each look-ahead state, see Lemma 12(1).

It is easy to see that the dtla M of Example 8 is canonical. It is earliest because $\mathsf{rlabs}_M(q_i^a) = \{\sigma, \tau, a\}$ and $\mathsf{rlabs}_M(q_i^b) = \{\sigma, \tau, b\}$. For an input tree $w(a)$, $\llbracket q_i^a \rrbracket_M$ outputs the first i symbols of w and the leaf a if $|w| \geq i$, and it outputs $w(a)$ if $|w| < i$. For an input tree $w(b)$, $\llbracket q_i^a \rrbracket_M$ is undefined. And the analogous statement holds for q_i^b.

Similarly, the dtla M of Example 14 (which is the la-uniform version of the dtla of Example 9) is canonical: for all $y, z \in \{a, b\}$, $\mathsf{rlabs}_M(q_{yz}) = \{yz, \sigma\}$ and $\llbracket q_{yz} \rrbracket_M$ is the restriction of $\llbracket M \rrbracket$ to $\llbracket p_{yz} \rrbracket_M$.

The term 'canonical' suggests that any two equivalent canonical dtlas M_1 and M_2 are the same, modulo a renaming of their states and look-ahead states. That is indeed true for dtops, as shown in [9, Theorem 15], but it does *not* hold for arbitrary dtlas: for instance, the dtla M of Example 14 and the dtop N of Example 9 are both canonical, and they are equivalent but not the same. It is, however, true if M_1 and M_2 have the same look-ahead automaton (by a proof similar to the one of [9, Theorem 15]), but that fact will not be needed in what follows.

For an la-uniform dtla M the sets $\mathsf{rlabs}_M(q)$ can be computed in a standard way. In fact, consider the directed graph with set of nodes $Q \cup \Delta$ and with the following edges: for every rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta$ of M, if $\mathsf{lab}(\zeta, \varepsilon) = d \in \Delta$ then there is an edge $q \to d$, and if $\zeta = q'(x_j)$ then there is an edge $q \to q'$. Note that the subgraph induced by Q is the graph E_M, as in the definition of a b-erasing dtla in Section 3. It is straightforward to show that $\mathsf{rlabs}_M(q) = \{d \in \Delta \mid q \to^* d\}$, as follows:

($\subseteq$) Structural induction on $s = a(s_1, \ldots, s_k)$, such that $q_M(s)$ has root label d. By Lemma 2(4), $q_M(s) = \zeta[\bar{q}(x_i) \leftarrow \bar{q}_M(s_i) \mid \bar{q} \in Q, i \in [k]]$ where $\zeta = \mathsf{rhs}(q, s, \varepsilon)$. If ζ has root label $d \in \Delta$, then there is an edge $q \to d$. If $\zeta = q'(x_j)$, then $q_M(s) = q'_M(s_j)$ and so $q \to q' \to^* d$ by induction.

($\supseteq$) Induction on the length of $q \to^* d$. If $q \to d$ then $q_M(a(s_1, \ldots, s_k)) = \zeta[\bar{q}(x_i) \leftarrow \bar{q}_M(s_i) \mid \bar{q} \in Q, i \in [k]]$ by Lemma 2(4), for any $s_i \in \llbracket p_i \rrbracket$, and so $q_M(a(s_1, \ldots, s_k))$ has root label d. We have used that M is la-uniform: if $\bar{q}(x_i)$ occurs in ζ, then $\rho(\bar{q}) = p_i$ and hence $\bar{q}_M(s_i)$ is defined by Lemma 12(1). If $q \to q' \to^* d$ then $q_M(a(s_1, \ldots, s_k)) = q'_M(s_j)$ has root label d, because by induction there exists s_j such that $q'_M(s_j)$ has root label d.

We now prove that canonicalness is a normal form for total dtlas. For an la-uniform dtla M, let $\mathsf{fix}(M)$ be a fixed subset of $\mathcal{T}_\Sigma$ such that for every $p \in P$ there is a unique $s \in \mathsf{fix}(M)$ with $\delta(s) = p$. Thus, $\mathsf{fix}(M)$ is a set of representatives of the equivalence classes $[\![p]\!]$, $p \in P$. Since every $[\![p]\!]$ is a regular tree language, a particular $\mathsf{fix}(M)$ can be computed from M. For every $p \in P$, let s_p be the unique tree in $\mathsf{fix}(M)$ with $\delta(s_p) = p$. We define

$$\mathsf{sumfix}(M) = \sum_{q \in Q} \mathsf{size}(q_M(s_{\rho(q)})).$$

Note that $\mathsf{sumfix}(M)$ is in $\mathbb{N}$ and can be computed from M.

Theorem 15. *For every la-uniform dtla M an equivalent canonical dtla $\mathsf{can}(M)$ can be constructed, with the same look-ahead automaton as M, such that*

$$\mathsf{lubdiff}(M) - \mathsf{sumfix}(M) \leq \mathsf{lubdiff}(\mathsf{can}(M)) \leq \mathsf{lubdiff}(M) + \mathsf{sumfix}(M).$$

Proof. We first prove the statement of this theorem for the case where M is earliest. Since the equivalence of two dtlas is decidable (see [13] and [9, Corollary 19]), it is decidable for two states q, q' of M whether or not $[\![q]\!]_M = [\![q']\!]_M$. If this holds, then q' can be replaced by q in every axiom and every right-hand side of a rule, thus making q' unreachable and hence superfluous. Since in $M(C[p])$ every $\langle q', p \rangle$ is replaced by $\langle q, p \rangle$, $\mathsf{lubdiff}(M)$ does not change. Thus, repeating this procedure one obtains a canonical dtla $\mathsf{can}(M)$ equivalent to M, with $\mathsf{lubdiff}(\mathsf{can}(M)) = \mathsf{lubdiff}(M)$.

For the interested reader we observe that for an earliest dtla M the equivalence relation $\equiv$ on Q defined by $q \equiv q'$ if and only if $[\![q]\!]_M = [\![q']\!]_M$, can in fact easily be computed by fixpoint iteration, because it is the largest equivalence relation on Q such that if $q \equiv q'$ then (a) $\rho(q) = \rho(q')$ and (b) if $\mathsf{rhs}(q, a, p_1, \ldots, p_k) = t[q_1(x_{i_1}), \ldots, q_r(x_{i_r})]$ where $t \in \mathcal{P}_\Sigma$ and $r = |V_\perp(t)|$, then $\mathsf{rhs}(q', a, p_1, \ldots, p_k) = t[q_1'(x_{i_1}), \ldots, q_r'(x_{i_r})]$ with $q_j \equiv q_j'$ for every $j \in [r]$. The straightforward proof of this is left to the reader, cf. the proof of [9, Theorem 13]. Thus, the full dtla equivalence test of [13, 9] is not needed.

It remains to be proved that every la-uniform dtla M can be transformed into an equivalent earliest dtla M', with the same look-ahead automaton, such that the distance between $\mathsf{lubdiff}(M')$ and $\mathsf{lubdiff}(M)$ is at most $\mathsf{sumfix}(M)$. If M is not earliest, then we obtain M' by repeatedly applying the following transformation step.

Transformation. We transform M into a dtla N with the same look-ahead automaton. Let Q_1 be the (nonempty) set of states $q \in Q$ such that $\mathsf{rlabs}_M(q)$ is a singleton, and for every $q \in Q_1$ let $\mathsf{rlabs}_M(q) = \{d_q\}$ and $m_q = \mathsf{rk}(d_q)$. The set of states of N is

$$Q_N := (Q - Q_1) \cup \{\langle q, i \rangle \mid q \in Q_1, i \in [m_q]\}.$$

When M arrives in state $q \in Q_1$ at a node u of an input tree s, N will first output the symbol d_q and then arrive at node u in the states $\langle q, 1 \rangle, \ldots, \langle q, m_q \rangle$, to compute the direct subtrees of the tree $q_M(s/u)$, where $\langle q, i \rangle$ computes the ith direct subtree $q_M(s/u)/i$. So, to describe N, we define for every tree $\zeta \in \mathcal{T}_\Delta(Q(\Omega))$ where Ω is any set of symbols of rank 0, the tree $\zeta \Phi_\Omega = \zeta[q(\omega) \leftarrow d_q(\langle q, 1 \rangle(\omega), \ldots, \langle q, m_q \rangle(\omega)) \mid q \in Q_1, \omega \in \Omega]$. For every $p \in P$, the p-axiom of N is $A(p)\Phi_{\{x_0\}}$. Every rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \rightarrow \zeta$ is changed into the rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \rightarrow \zeta \Phi_{X_k}$ if

18

$q \in Q - Q_1$, and into the m_q rules $\langle q, i \rangle (a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta \Phi_{X_k} / i$ (with $i \in [m_q]$) if $q \in Q_1$. Note that in the latter case the root of $\zeta \Phi_{X_k}$ has label d_q and so its ith direct subtree is well defined. That is clear if $\mathsf{lab}(\zeta, \varepsilon) = d_q$. If $\zeta = q'(x_j)$, then $q' \in Q_1$ and $d_{q'} = d_q$; so $\zeta \Phi_{X_k} = d_q(\langle q', 1 \rangle (x_j), \ldots, \langle q', m_q \rangle (x_j))$ and one obtains the rules $\langle q, i \rangle (a(x_1 : p_1, \ldots, x_k : p_k)) \to \langle q', i \rangle (x_j)$ for $i \in [m_q]$.

If M has la-map ρ, then N is la-uniform with la-map ρ_N such that $\rho_N(q) = \rho(q)$ for $q \in Q - Q_1$ and $\rho_N(\langle q, i \rangle) = \rho(q)$ for $q \in Q_1$ and $i \in [m_q]$.

It should be clear intuitively that N is equivalent to M. Formally it can easily be shown for every $s \in \mathcal{T}_\Sigma$ and every reachable sentential form ξ of M for s, that $\xi \Phi_{V(s)}$ is a reachable sentential form of N for s (where each computation step of M is simulated by one or m_q computation steps of N), and hence $N(s) = M(s)$. We will compute $\mathsf{lubdiff}(N)$ below; to do that we need to extend the previous statement to trees $\bar{s} \in \mathcal{T}_\Sigma(P)$. Let Ψ be the substitution $[\langle q, p \rangle \leftarrow d_q(\langle \langle q, 1 \rangle, p \rangle, \ldots, \langle \langle q, m_q \rangle, p \rangle) \mid q \in Q_1, p \in P]$. Then, for every reachable sentential form ξ of M° for $\bar{s}$, the tree $\xi \Phi_{V(s)} \Psi$ is a reachable sentential form of N° for $\bar{s}$, and hence $N(\bar{s}) = M(\bar{s}) \Psi$.

Repetition. Using Lemma 2(3/4), it can easily be shown for every $q \in Q$ and $s \in \llbracket \rho(q) \rrbracket$, that $q_N(s) = q_M(s)$ if $q \notin Q_1$, and that $\langle q, i \rangle_N(s) = q_M(s)/i$ for every $i \in [m_q]$ if $q \in Q_1$. From this (and assuming that $\mathsf{fix}(N) = \mathsf{fix}(M)$), it should be clear that $\mathsf{sumfix}(N) < \mathsf{sumfix}(M)$, because, for $q \in Q_1$ and $s \in \llbracket \rho(q) \rrbracket$,

$$\sum_{i \in [m_q]} \mathsf{size}(\langle q, i \rangle_N(s)) = \sum_{i \in [m_q]} \mathsf{size}(q_M(s)/i) = \mathsf{size}(q_M(s)) - 1.$$

Hence, the repetition of the above transformation stops after at most $\mathsf{sumfix}(M)$ steps, with an earliest dtla M' equivalent to M.

Difference trees. It now remains to prove that the distance between $\mathsf{lubdiff}(N)$ and $\mathsf{lubdiff}(M)$ is at most 1, i.e., that $\mathsf{lubdiff}(N) \leq \mathsf{lubdiff}(M) + 1$ and $\mathsf{lubdiff}(M) \leq \mathsf{lubdiff}(N) + 1$. Consider $C \in \mathcal{C}_\Sigma$ and $p, p' \in P$ with $p \neq p'$. Recall from above that $N(C[p]) = M(C[p]) \Psi = M(C[p])[\langle q, p \rangle \leftarrow d_q(\langle \langle q, 1 \rangle, p \rangle, \ldots, \langle \langle q, m_q \rangle, p \rangle) \mid q \in Q_1]$ and similarly for p'. We observe that if v is a node of $M(C[p])$, then each proper ancestor of v has the same label in $M(C[p])$ and $N(C[p])$, and similarly for p'.

Let v be a difference node of $N(C[p])$ and $N(C[p'])$ that is not a leaf of $N(C[p])$. Then $v \in V(M(C[p]))$. If also $v \in V(M(C[p']))$, then Lemma 6 implies that v is also a difference node of $M(C[p])$ and $M(C[p'])$ (in fact, by the above observation every proper ancestor of v has the same label in $M(C[p])$ and $M(C[p'])$; if v would have the same label in $M(C[p])$ and $M(C[p'])$, then that label would be in Δ because $p \neq p'$, and hence v would have the same label in $N(C[p])$ and $N(C[p'])$). Consequently, $\mathsf{ht}(N(C[p])/v) = \mathsf{ht}(M(C[p]) \Psi / v) \leq \mathsf{ht}(M(C[p])/v) + 1 \leq \mathsf{lubdiff}(M) + 1$. If $v \notin V(M(C[p']))$, then the parent $\hat{v}$ of v is a difference node of $M(C[p])$ and $M(C[p'])$ (in fact, the label of $\hat{v}$ in $M(C[p'])$ is $\langle q, p' \rangle$ for some $q \in Q_1$, and so $\hat{v}$ has different labels in $M(C[p])$ and $M(C[p'])$ because $p \neq p'$). Hence, in this case, $\mathsf{ht}(N(C[p])/v) \leq \mathsf{ht}(N(C[p])/\hat{v}) \leq \mathsf{ht}(M(C[p])/\hat{v}) + 1 \leq \mathsf{lubdiff}(M) + 1$. This proves that $\mathsf{lubdiff}(N) \leq \mathsf{lubdiff}(M) + 1$.

Now let v be a difference node of $M(C[p])$ and $M(C[p'])$ that is not a leaf of $M(C[p])$; note that $\mathsf{lab}(N(C[p]), v) = \mathsf{lab}(M(C[p]), v) \in \Delta$. If v is also a difference node of $N(C[p])$ and $N(C[p'])$, then $\mathsf{ht}(M(C[p])/v) \leq \mathsf{ht}(M(C[p]) \Psi / v) =$

$\mathsf{ht}(N(C[p])/v) \le \mathsf{lubdiff}(N)$. Now assume that v is not a difference node of $N(C[p])$ and $N(C[p'])$. Then, by Lemma 6, the label of v in $M(C[p'])$ is $\langle q, p' \rangle$ for some $q \in Q_1$ and $\mathsf{lab}(N(C[p]), v) = \mathsf{lab}(N(C[p']), v) = d_q$. This implies, again by Lemma 6, that the children of v are difference nodes of $N(C[p])$ and $N(C[p'])$. Let vi be a child of v for which $\mathsf{ht}(M(C[p])/vi)$ is maximal. Then we have $\mathsf{ht}(M(C[p])/v) = \mathsf{ht}(M(C[p])/vi) + 1 \le \mathsf{ht}(N(C[p])/vi) + 1 \le \mathsf{lubdiff}(N) + 1$. This proves that $\mathsf{lubdiff}(M) \le \mathsf{lubdiff}(N) + 1$. $\qquad\square$

Note that it follows from the inequalities for $\mathsf{lubdiff}(\mathsf{can}(M))$ that if $h(M)$ is a difference bound for M, then $h(M) + \mathsf{sumfix}(M)$ is a difference bound for $\mathsf{can}(M)$. In fact (similar to the argument after Lemma 11), if $\mathsf{diff}(\mathsf{can}(M))$ is finite, then $\mathsf{diff}(M)$ is finite because $\mathsf{lubdiff}(M) \le \mathsf{lubdiff}(\mathsf{can}(M)) + \mathsf{sumfix}(M)$, hence $\mathsf{lubdiff}(M) \le h(M)$ and hence $\mathsf{lubdiff}(\mathsf{can}(M)) \le \mathsf{lubdiff}(M) + \mathsf{sumfix}(M) \le h(M) + \mathsf{sumfix}(M)$.

Note also that the transformation in the above proof does not preserve the ultralinear property (as can be seen in the next example).

Example 16. In this example we denote by Y the nonempty subsets of $\{a, b\}$, i.e., $Y = \{\{a\}, \{b\}, \{a, b\}\}$. Let $\Sigma = \{\sigma^{(2)}, a^{(0)}, b^{(0)}\}$ and $\Delta = \{\sigma_y^{(2)} \mid y \in Y\} \cup \{a^{(0)}, b^{(0)}\}$. We consider an la-uniform dtla M such that $M(a) = a$, $M(b) = b$, and $M(\sigma(s_1, s_2)) = \sigma_y(M(s_1), M(s_2))$ for $s_1, s_2 \in \mathcal{T}_\Sigma$, where y is the set of labels of the leaves of $\sigma(s_1, s_2)$. Its set of look-ahead states is $P = \{p_y \mid y \in Y\}$ and δ is defined in the obvious way: $\delta(a) = \{a\}$, $\delta(b) = \{b\}$, and $\delta(\sigma, p_y, p_z) = p_{y \cup z}$ for $y, z \in Y$. Its set of states is $Q = \{q_y \mid y \in Y\}$ with $\rho(q_y) = p_y$, its axioms are $A(p_y) = q_y(x_0)$ for every $y \in Y$, and its set R consists of the rules $q_{\{a\}}(a) \to a$, $q_{\{b\}}(b) \to b$, and

$$q_{y \cup z}(\sigma(x_1 : p_y, x_2 : p_z)) \to \sigma_{y \cup z}(q_y(x_1), q_z(x_2))$$

for $y, z \in Y$.

The dtla M_1 that is obtained from M by identifying all its states into one state q, is equivalent to M; it has $\mathsf{rlabs}_{M_1}(q) = \Delta$, but it is not la-uniform. However, M is not earliest: in fact, $\mathsf{rlabs}_M(q_{\{a\}}) = \{\sigma_{\{a\}}, a\}$ and $\mathsf{rlabs}_M(q_{\{b\}}) = \{\sigma_{\{b\}}, b\}$, but $\mathsf{rlabs}_M(q_{\{a,b\}}) = \{\sigma_{\{a,b\}}\}$. Let N be the dtla obtained from M by applying the transformation in the proof of Theorem 15 once. Then $Q_1 = \{q_{\{a,b\}}\}$. We will write the states $\langle q_{\{a,b\}}, 1 \rangle$ and $\langle q_{\{a,b\}}, 2 \rangle$ as $q_{1\{a,b\}}$ and $q_{2\{a,b\}}$, respectively. So, N has states $q_{\{a\}}$, $q_{\{b\}}$, $q_{1\{a,b\}}$, and $q_{2\{a,b\}}$. Its axioms are $A_N(p_y) = q_y(x_0)$ for $y = \{a\}$ or $y = \{b\}$ (just as in M), and $A_N(p_y) = \sigma_y(q_{1y}(x_0), q_{2y}(x_0))$ for $y = \{a, b\}$. For $y = \{a\}$ or $y = \{b\}$, its set R_N of rules contains the rule

$$q_y(\sigma(x_1 : p_y, x_2 : p_y)) \to \sigma_y(q_y(x_1), q_y(x_2))$$

plus the rules $q_{\{a\}}(a) \to a$ and $q_{\{b\}}(b) \to b$ (just as in M). Moreover, for $y = \{a, b\}$, R_N contains the rules

$$q_{1y}(\sigma(x_1 : p_y, x_2 : p_z)) \to \sigma_y(q_{1y}(x_1), q_{2y}(x_1))$$
$$q_{2y}(\sigma(x_1 : p_z, x_2 : p_y)) \to \sigma_y(q_{1y}(x_2), q_{2y}(x_2))$$

for all $z \in Y$, plus the rules

$$q_{1y}(\sigma(x_1 : p_w, x_2 : p_z)) \to q_w(x_1)$$
$$q_{2y}(\sigma(x_1 : p_z, x_2 : p_w)) \to q_w(x_2)$$

20

for all $w, z \in Y$ with $w \neq y$ and $w \cup z = y$. For N we have $\mathsf{rlabs}_N(q_{\{a\}}) = \{\sigma_{\{a\}}, a\}$ and $\mathsf{rlabs}_N(q_{\{b\}}) = \{\sigma_{\{b\}}, b\}$ as for M, and we have $\mathsf{rlabs}_N(q_{1\{a,b\}}) = \mathsf{rlabs}_N(q_{2\{a,b\}}) = \Delta$. Hence N is earliest. Obviously, N is also canonical, and so $N = \mathsf{can}(M)$.

Clearly, N is not ultralinear, because condition (2) of the definition of ultralinearity cannot be satisfied for the rule $q_{1y}(\sigma(x_1 : p_y, x_2 : p_z)) \rightarrow \sigma_y(q_{1y}(x_1), q_{2y}(x_1))$. Since M is linear, this shows that ultralinearity is not preserved. After the definition of canonicalness we observed (without proof) that the dtla $N = \mathsf{can}(M)$ is unique, modulo a renaming of states, and so the nonpreservation of ultralinearity in the proof of Theorem 15 is, in fact, unavoidable. $\square$

6 Difference Tuples and the Algorithm

In this section we introduce the notion of difference tuple, generalizing the notion of difference trees by considering all look-ahead states of a dtla simultaneously. Based on that notion, we present an algorithm which, for a given total dtla M and a difference bound for M, decides whether M is equivalent to a dtop, and if so, constructs such a dtop. By the results of the previous section, we may assume M to be canonical.

6.1 Difference Tuples

Let M be a total 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 proper dtla, i.e., 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 all these trees does *not* depend on the look-ahead. In contrast, the subtrees of the above trees 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

$$\mathsf{diftup}(t_1, \ldots, t_n) := \{(t_1/v, \ldots, t_n/v) \mid v \in V_\perp(\sqcap\{t_1, \ldots, t_n\})\},$$

which is a set of n-tuples in $\mathcal{T}_\Delta(Q \times P)^n$. We will say that $(t_1/v, \ldots, t_n/v)$ is the difference tuple of $t_1, \ldots, t_n$ at such a node v. 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 $\mathsf{pref}(M, C) \in \mathcal{P}_\Delta$ as

$$\mathsf{pref}(M, C) := \sqcap\{M(C[p]) \mid p \in P\} = \sqcap\{M(C[\hat{p}_1]), \ldots, M(C[\hat{p}_n])\}.$$

Note that $\mathsf{pref}(M, C)$ is a Δ-pattern because a node with label $\langle q, \hat{p}_i \rangle$ in $M(C[\hat{p}_i])$ cannot have the same label in $M(C[\hat{p}_j])$ for $i \neq j$. Note also that

$$\mathsf{diftup}(M) = \{(M(C[\hat{p}_1])/v, \ldots, M(C[\hat{p}_n])/v) \mid C \in \mathcal{C}_\Sigma, \ v \in V_\perp(\mathsf{pref}(M, C))\}.$$

We define the number $\mathsf{lubdiftup}(M) \in \mathbb{N} \cup \{\infty\}$ to be the least upper bound of the heights of the components of all difference tuples of M, i.e.,

$$\mathsf{lubdiftup}(M) = \mathsf{lub}\{\mathsf{ht}(\bar{t}) \mid \bar{t} \in \mathsf{diftup}(M)\}.$$

Difference tuples are introduced for the following reason, cf. Lemma 4. We wish to decide whether M is equivalent to a dtop. If there exists a dtop N that is equivalent to M, then we expect intuitively for any $s \in \mathcal{T}_\Sigma$ and $C \in \mathcal{C}_\Sigma$, that $N(C[s]) = t[q_{1N}(s), \ldots, q_{rN}(s)]$ where $t = \mathsf{pref}(M, C) = \sqcap\{M(C[\hat{p}_1]), \ldots, M(C[\hat{p}_n])\}$ and $r = |V_\perp(t)|$. Thus, 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 for $i \in [r]$, then we expect the difference tuple $(M(C[\hat{p}_1])/v_i, \ldots, M(C[\hat{p}_n])/v_i)$ of these output trees at 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 will be proved in Lemma 24, under the condition that M is canonical and N is earliest. If N is also canonical, then its states are in one-to-one correspondence with the difference tuples of M, as will be proved in Lemma 26.

Before giving some examples, we show that $\mathsf{lubdiftup}(M)$ equals $\mathsf{lubdiff}(M)$, defined in Section 4. This implies that $\mathsf{diftup}(M)$ is finite if and only if $\mathsf{diff}(M)$ is finite.

Lemma 17. *For every total dtpla M, $\mathsf{lubdiff}(M) = \mathsf{lubdiftup}(M)$.*

Proof. ($\leq$) We show that every difference tree is a subtree of a component of a difference tuple. Consider a difference tree $M(C[p])/v$ with $C \in \mathcal{C}_\Sigma$, $p \in P$, and v a difference node of $M(C[p])$ and $M(C[p'])$ where $p' \in P$, i.e., $v \in V_\perp(M(C[p]) \sqcap M(C[p']))$. Since $\mathsf{pref}(M, C) \sqsubseteq M(C[p]) \sqcap M(C[p'])$, there is an ancestor $\hat{v}$ of v such that $\hat{v} \in V_\perp(\mathsf{pref}(M, C))$. Thus, $M(C[p])/\hat{v}$ is a component of a difference tuple, and $M(C[p])/v$ is one of its subtrees.

($\geq$) We show that every component of a difference tuple is a difference tree. Consider $M(C[p])/v$ with $C \in \mathcal{C}_\Sigma$, $p \in P$, and $v \in V_\perp(\mathsf{pref}(M, C))$. By Lemma 1, each proper ancestor of v has the same label in all $M(C[\bar{p}])$, $\bar{p} \in P$, but v does not have the same label in all $M(C[\bar{p}])$. Thus, there exists $p' \in P$ such that v has different labels in $M(C[p])$ and $M(C[p'])$. Then v is a difference node of $M(C[p])$ and $M(C[p'])$ by Lemma 6. $\square$

We now give examples of $\mathsf{diftup}(M)$ for several total dtplas M. In the remainder of this subsection, we will use the dtla of Example 8 as a running example.

Example 18. For the dtla M of Example 7, 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 8, also with the order $P = \{p_a, p_b\}$, we obtain that $\mathsf{diftup}(M)$ consists of all pairs

$$\begin{aligned}
&(\langle q^a_{i+50}, p_a\rangle, \ \langle q^b_i, p_b\rangle) \ \text{for } i \in [30], \\
&(z(\langle q^a_{50-|z|}, p_a\rangle), \ b) \quad \text{for } z \in \{\sigma, \tau\}^* \text{ with } |z| < 50, \text{and} \\
&(z(a), \ b) \qquad\qquad\qquad \text{for } z \in \{\sigma, \tau\}^* \text{ with } |z| = 50.
\end{aligned}$$

For the dtla M of Example 14 (which is the la-uniform version of the dtla of Example 9) it is not difficult to see that $\text{diff}(M) = \{a, b\} \cup \{\langle q_{yz}, p_{yz} \rangle \mid y, z \in \{a, b\}\}$, and that the set $\text{diftup}(M)$ consists of the three 4-tuples (a, a, b, b), (a, b, a, b), and $(\langle q_{aa}, p_{aa} \rangle, \langle q_{ab}, p_{ab} \rangle, \langle q_{ba}, p_{ba} \rangle, \langle q_{bb}, p_{bb} \rangle)$, where we have taken the order $P = \{p_{aa}, p_{ab}, p_{ba}, p_{bb}\}$.

For the dtla M of Example 10, $\text{diftup}(M) = \{(e, o), (o, e)\}$.

In the above examples, the components of the difference tuples are exactly the difference trees. As another example, let $\Sigma = \Delta = \{\sigma^{(1)}, a^{(0)}, b^{(0)}, c^{(0)}\}$ and consider the dtla M with $P = \{p_a, p_b, p_c\}$, $\delta(y) = p_y$ and $\delta(\sigma, p_y) = p_y$ for $y \in \{a, b, c\}$, $Q = \emptyset$, $A(p_a) = a$, $A(p_b) = \sigma(b)$, and $A(p_c) = \sigma(\sigma(c))$. Thus, for every $n \in \mathbb{N}$, M translates $\sigma^n a$ into a, $\sigma^n b$ into σb, and $\sigma^n c$ into $\sigma\sigma c$. Since $a \sqcap \sigma b = \bot$, $a \sqcap \sigma\sigma c = \bot$, and $\sigma b \sqcap \sigma\sigma c = \sigma\bot$, we obtain that $\text{diff}(M) = \{a, \sigma b, \sigma\sigma c, b, \sigma c\}$. Since $a \sqcap \sigma b \sqcap \sigma\sigma c = \bot$, we obtain that $\text{diftup}(M) = \{(a, \sigma b, \sigma\sigma c)\}$. Thus, b and σc are difference trees that are not components of a difference tuple (but are subtrees of such components). Note that there is a dtop with one state that is equivalent to M. □

In the next lemmas N is a total dtop, equivalent to M. We assume that the unique look-ahead state of N is $\bot$. So, N° translates input trees in $\mathcal{T}_\Sigma(\{\bot\})$, in particular Σ-contexts, into output trees in $\mathcal{T}_\Delta(Q_N \times \{\bot\})$; for a Σ-context C we of course write C instead of $C[\bot]$. The unique axiom $A_N(\bot)$ is denoted by A_N, a rule $q(a(x_1 : \bot, \ldots, x_k : \bot)) \to \zeta$ is written $q(a(x_1, \ldots, x_k)) \to \zeta$, and ζ is denoted $\text{rhs}_N(q, a)$. For a tree $t \in \mathcal{T}_\Delta(Q_N \times \{\bot\})$ we define the pattern $t\Phi \in \mathcal{P}_\Delta$ by $t\Phi = t[\langle q, \bot \rangle \leftarrow \bot \mid q \in Q_N]$; similarly, for $t \in \mathcal{T}_\Delta(Q_N(X))$ we define $t\Phi = t[q(x_i) \leftarrow \bot \mid q \in Q_N, i \in \mathbb{N}]$.

Let M be a canonical dtpla and N a dtop such that $\llbracket M \rrbracket = \llbracket N \rrbracket$. We first show 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 $p \in P_M$,

$$M(C[p]) = N(C)[\langle q, \bot \rangle \leftarrow \varphi(q, p) \mid q \in Q_N]. \tag{2}$$

Note that $\varphi(q, p)$ must be 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. This amount is stored in each state q of N, for every look-ahead state p of M.

The next lemma provides an obvious equivalent formulation of Equation (2), using the substitution Φ defined above.

Lemma 19. *For every $C \in \mathcal{C}_\Sigma$ and $p \in P_M$, Equation (2) is equivalent to the following two conditions:*

(a) $N(C)\Phi \sqsubseteq M(C[p])$, and
(b) for every $v \in V_\bot(N(C)\Phi)$ and $q \in Q_N$,
    if $N(C)/v = \langle q, \bot \rangle$, then $\varphi(q, p) = M(C[p])/v$.

Example 20. Consider M and N as in Example 8. As observed after Example 14, the dtpla M is canonical. It is easy to check that N is also canonical. We will show that there is an aheadness mapping φ from N to M. To this aim, let $w \in \{\sigma, \tau\}^*$ and consider the context $C = w(\bot)$. In Example 8 we have computed $M(C[p_a])$ and $M(C[p_b])$.

Let us now compute $N(C)$. For $|w| < 30$ we obtain that $N(C) = w(\langle q_{30-|w|}, \bot\rangle)$. Since in this case $M(C[p_a]) = w(\langle q^a_{80-|w|}, p_a\rangle)$ and $M(C[p_b]) = w(\langle q^b_{30-|w|}, p_b\rangle)$, Equation (2) requires for $i = 30 - [w]$, and hence for every $i \in [30]$, that

$$\varphi(q_i, p_a) = \langle q^a_{i+50}, p_a\rangle \text{ and } \varphi(q_i, p_b) = \langle q^b_i, p_b\rangle.$$

Note that in this case M is not properly ahead of N. For $w = w_1 z$ with $|w_1| = 30$ and $|z| < 50$, we get that $N(C) = w_1(\langle q_z, \bot\rangle)$. Since $M(C[p_a]) = w_1 z(\langle q^a_{80-|w_1 z|}, p_a\rangle)$ and $M(C[p_b]) = w_1(b)$, the mapping φ should satisfy

$$\varphi(q_z, p_a) = z(\langle q^a_{50-|z|}, p_a\rangle) \text{ and } \varphi(q_z, p_b) = b, \text{ for } |z| < 50.$$

Finally, if $w = w_1 z w_2$ with $|w_1| = 30$ and $|z| = 50$, then $N(C) = w_1(\langle q_z, \bot\rangle)$. Since $M(C[p_a]) = w_1 z(a)$ and $M(C[p_b]) = w_1(b)$, we get that

$$\varphi(q_z, p_a) = z(a) \text{ and } \varphi(q_z, p_b) = b, \text{ for } |z| = 50.$$

Clearly, the above requirements define φ uniquely, and hence φ is an aheadness mapping from N to M (and, in fact, the unique one). □

Lemma 21. *Let M be a canonical dtpla and N a dtop such that $[\![M]\!] = [\![N]\!]$. Then there is a unique aheadness mapping 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.

    Claim 1.   Let $p \in P_M$ and let C be a Σ-context.
For every $d \in \Delta$, $V_d(N(C)) \subseteq V_d(M(C[p]))$. Equivalently, $N(C)\Phi \sqsubseteq M(C[p])$.

    Proof. We show that every node v of $N(C)$ with label $d \in \Delta$ is also a node of $M(C[p]$, with the same label. The proof is by induction on the length of v, as follows. Since v's proper ancestors are in $V_\Delta(N(C))$, the induction hypothesis implies that v is a node of $M(C[p])$. Consider an arbitrary $s \in [\![p]\!]_M$. By Lemma 4, v has label d in $N(C[s])$. Since $[\![M]\!] = [\![N]\!]$, $M(C[s]) = N(C[s])$ and so v has label d in $M(C[s])$. Suppose that v does not have label d in $M(C[p])$. Then, again by Lemma 4, v must have 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, $\rho_M(q) = p$ by Lemma 12(3) and hence $[\![p]\!]_M = \mathsf{dom}([\![q]\!]_M)$ by Lemma 12(1). This proves the claim.

    Next we show that the amount in which M is ahead of N, is independent of C.

    Claim 2. Let C_1, C_2 be Σ-contexts, $v_1, v_2 \in \mathbb{N}^*_+$, $q \in Q_N$, and $p \in P_M$.
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 4, and so $M(C_1[s])/v_1 = M(C_2[s])/v_2$. Hence, again by Lemma 4, $t_1\Psi_s = t_2\Psi_s$ 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 $q_{1M}(s)$ 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 $q_{1M}(s) = q_{2M}(s)$ for all $s \in [\![p]\!]_M$. Since, as observed in the proof of Claim 1, $[\![p]\!]_M$ is the domain of both $[\![q_1]\!]_M$ and $[\![q_2]\!]_M$ by Lemma 12, we obtain that $[\![q_1]\!]_M = [\![q_2]\!]_M$, contradicting the fact that M is canonical. This proves the claim.

An aheadness mapping from N to M can now be defined as follows. Let $q \in Q_N$ and $p \in P_M$. By Lemma 5, 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 follows from Claim 1 that if $N(C) = t[\langle q_1, \bot \rangle, \ldots, \langle q_r, \bot \rangle]$ with $t \in \mathcal{P}_\Delta$ and $r = |V_\bot(t)|$, then $M(C[p]) = t[M(C[p])/v_1, \ldots, M(C[p])/v_r]$ where v_i is the ith occurrence of $\bot$ in t, and hence $M(C[p]) = t[\varphi(q_1, p), \ldots, \varphi(q_r, p)]$. Thus, φ satisfies Equation (2): the requirement for an aheadness mapping. Obviously, if φ' is an aheadness mapping from N to M, and C is a Σ-context such that $N(C)$ has a node v labeled $\langle q, \bot \rangle$, then $M(C[p])/v = \varphi'(q, p)$ for every $p \in P_M$, by Lemma 19(b). Thus $\varphi' = \varphi$, which shows that φ is the unique aheadness mapping from N to M. $\qquad\square$

Let M and N be as in Lemma 21, with aheadness mapping φ. Let $\bar{s}$ be an input tree and v a node of $\bar{s}$ with $\bar{s}/v = s$. When N arrives at v in state q, we expect that, when processing s, it will first output $\varphi(q, \delta_M(s))$ and then produce the output of M on s. This is formalized in the following lemma.

Lemma 22. *Let M be a canonical dtpla and N a dtop such that $[\![M]\!] = [\![N]\!]$. Let φ be the aheadness mapping from N to M. 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. By Lemma 5, there are C, v such that $N(C)/v = \langle q, \bot \rangle$. It follows from Lemma 19(b) that $M(C[p])/v = \varphi(q, p)$. Since M and N are equivalent, $N(C[s]) = M(C[s])$. Applying Lemma 4 twice, we obtain $q_N(s) = N(C[s])/v = M(C[s])/v = (M(C[p])/v)[\langle \bar{q}, p \rangle \leftarrow \bar{q}_M(s) \mid \bar{q} \in Q_M]$, which proves the equation. $\qquad\square$

Example 23. We continue Example 20. Consider a state q_z of N with $|z| < 50$, and let $s = w_1 w_2(a)$ with $|zw_1| = 50$. Then $q_{zN}(s) = zw_1(a)$. Now $\varphi(q_z, p_a) = z(\langle q_{50-|z|}^a, p_a \rangle) = z(\langle q_{|w_1|}^a, p_a \rangle)$ by Example 20, and $q_{|w_1|M}^a(s) = w_1(a)$. $\qquad\square$

In the next lemma we prove our basic 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.

Lemma 24. *Let M be a canonical dtpla and N an earliest dtop such that $[\![M]\!] = [\![N]\!]$. Let $C \in \mathcal{C}_\Sigma$ and let φ be the aheadness mapping from N to M. Then*

(1) $N(C)\Phi = \mathsf{pref}(M, C)$, and
(2) for every $v \in V_\bot(N(C)\Phi)$, $q \in Q_N$, and $p \in P_M$,
 if $N(C)/v = \langle q, \bot \rangle$, then $\varphi(q, p) = M(C[p])/v$.

Proof. By Lemma 19(b) it suffices to prove statement (1). By Lemma 19(a), $N(C)\Phi \sqsubseteq M(C[p])$ for every $p \in P_M$ and so $N(C)\Phi \sqsubseteq \mathsf{pref}(M,C)$ by the definition of greatest lower bound. To show equality, we prove for every node v of $N(C)\Phi$ with label $\bot$ that v has label $\bot$ in $\mathsf{pref}(M,C)$ too. Let $N(C)/v = \langle q, \bot \rangle$ for $q \in Q_N$. Then, by Lemma 19(b), $M(C[p])/v = \varphi(q,p)$ for every $p \in P_M$. Suppose now that v has label $d \in \Delta$ in $\mathsf{pref}(M,C)$. Then v has label d in every $M(C[p])$, and so the root symbol of $\varphi(q,p)$ is d for every $p \in P_M$. It then follows from Lemma 22 that $q_N(s)$ has root label d for every $s \in \mathcal{T}_\Sigma$, contradicting the fact that N is earliest. Hence v has label $\bot$ in $\mathsf{pref}(M,C)$. □

Example 25. For M and N of Example 8, the statements of Lemma 24 are easy to check from Examples 8 and 20. As an example, let $C = w_1 z(\bot)$ with $|w_1| = 30$ and $|z| < 50$. Then $N(C) = w_1(\langle q_z, \bot \rangle)$, $M(C[p_a]) = w_1 z(\langle q^a_{80-|w_1 z|}, p_a \rangle)$ and $M(C[p_b]) = w_1(b)$. Thus $N(C)\Phi = w_1(\bot) = M(C[p_a]) \sqcap M(C[p_b])$, and for the node v with $N(C)/v = \langle q_z, \bot \rangle$ we have, using the definition of φ in Example 20, that $\varphi(q_z, p_a) = z(\langle q^a_{50-|z|}, p_a \rangle) = M(C[p_a])/v$ and $\varphi(q_z, p_b) = b = M(C[p_b])/v$. □

If a dtpla M is equivalent to a dtop, then it is equivalent to a canonical dtop by Theorem 15. By [9, Theorem 15], two (reduced) canonical dtops are equivalent if and only if they are the same (modulo a renaming of their states). Thus, if a dtpla 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 the dtop $\mathsf{td}(M)$ can be constructed from M and $\mathsf{diftup}(M)$.

We start by showing that $Q_{\mathsf{td}(M)}$ can be identified with $\mathsf{diftup}(M)$, assuming N to be canonical too.

Lemma 26. *Let M, N be equivalent canonical dtlas such that M is a dtpla and N is a dtop. Let $\varphi : Q_N \times P_M \to T_\Delta(Q_M \times P_M)$ be the aheadness mapping from N to M, and let for $q \in Q_N$*

$$\psi(q) = (\varphi(q, \hat{p}_1), \ldots, \varphi(q, \hat{p}_n))$$

where $P_M = \{\hat{p}_1, \ldots, \hat{p}_n\}$. Then ψ is a bijection between Q_N and $\mathsf{diftup}(M)$.

Proof. The proof is in three parts.

(i) For all $q \in Q_N$, $\psi(q) \in \mathsf{diftup}(M)$.

Proof. Let $C \in \mathcal{C}_\Sigma$ and $v \in \mathbb{N}^*_+$ be such that $N(C)/v = \langle q, \bot \rangle$, by Lemma 5. By Lemma 24, $v \in V_\bot(\mathsf{pref}(M,C))$ and $M(C[\hat{p}_i])/v = \varphi(q, \hat{p}_i)$ for every $i \in [n]$. That shows that $\psi(q) \in \mathsf{diftup}(M)$.

(ii) For every $(t_1, \ldots, t_n) \in \mathsf{diftup}(M)$ there exists $q \in Q_N$ such that $\psi(q) = (t_1, \ldots, t_n)$.

Proof. If $(t_1, \ldots, t_n) \in \mathsf{diftup}(M)$ then there are $C \in \mathcal{C}_\Sigma$ and $v \in \mathbb{N}^*_+$ such that $\mathsf{pref}(M,C)/v = \bot$ and $M(C[\hat{p}_i])/v = t_i$ for $i \in [n]$. By Lemma 24, $N(C)/v = \langle q, \bot \rangle$ 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 $i \in [n]$.

(iii) For all $q_1, q_2 \in Q_N$, if $\psi(q_1) = \psi(q_2)$ then $q_1 = q_2$.

Proof. Let $\psi(q_1) = \psi(q_2)$. By Lemma 22, $q_{1N}(s) = q_{2N}(s)$ for all $s \in \mathcal{T}_\Sigma$. In other words, $[\![q_1]\!]_N = [\![q_2]\!]_N$ and hence $q_1 = q_2$ because N is canonical. □

Thus, by renaming the states of N according to ψ, we obtain a canonical dtop N', equivalent to M, such that $Q_{N'} = \mathsf{diftup}(M)$. Moreover, the aheadness mapping φ' from N' to M satisfies $\varphi'((t_1, \ldots, t_n), \hat{p}_i) = t_i$ for every $(t_1, \ldots, t_n) \in \mathsf{diftup}(M)$ and $i \in [n]$, in other words, φ' consists of the projections of difference tuples on their components.

Example 27. For M and N of Example 8, we obtain from Example 20 that

$$\psi(q_i) = (\langle q_{i+50}^a, p_a \rangle, \langle q_i^b, p_b \rangle) \text{ for } i \in [30],$$

$$\psi(q_z) = (z(\langle q_{50-|z|}^a, p_a \rangle), b) \quad \text{for } z \in \{\sigma, \tau\}^* \text{ with } |z| < 50, \text{and}$$

$$\psi(q_z) = (z(a), b) \qquad\qquad \text{for } z \in \{\sigma, \tau\}^* \text{ with } |z| = 50.$$

Thus, by Example 18, the mapping $\psi : Q_N \to \mathsf{diftup}(M)$ is indeed a bijection. □

From Lemma 26 we obtain the following corollary, which, by Example 10, does not hold in the other direction.

Corollary 28. *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. By Lemma 13 we may assume that M is la-uniform. Let $\mathsf{can}(M)$ be a canonical dtpla equivalent to M, which exists by Theorem 15. If M, and hence $\mathsf{can}(M)$, is equivalent to a (total) dtop, then $\mathsf{can}(M)$ is equivalent to a canonical dtop, by Theorem 15. By Lemmas 21 and 26, $\mathsf{diftup}(\mathsf{can}(M))$ is finite, and so $\mathsf{diff}(\mathsf{can}(M))$ is finite by Lemma 17. Hence $\mathsf{diff}(M)$ is finite by Theorem 15, because $\mathsf{lubdiff}(M) \leq \mathsf{lubdiff}(\mathsf{can}(M)) + \mathsf{sumfix}(M)$. □

Next we show, for a canonical dtpla M, how to compute the axiom $A_{\mathsf{td}(M)}$ of $\mathsf{td}(M)$, representing the states of $\mathsf{td}(M)$ by difference tuples. For that we need some terminology.

For an la-uniform dtla M and a tree $t \in \mathcal{T}_\Delta(Q(X))$ we define $t\Omega \in \mathcal{T}_\Delta(Q \times P)$ by $t\Omega = t[q(x_i) \leftarrow \langle q, \rho(q) \rangle \mid q \in Q, i \in \mathbb{N}]$. In particular, for a total dtop N, $t\Omega = t[q(x_i) \leftarrow \langle q, \bot \rangle \mid q \in Q_N, i \in \mathbb{N}]$. Recall also the definition of the pattern $t\Phi \in \mathcal{P}_\Delta$ for $t \in \mathcal{T}_\Delta(Q_N(X))$ or $t \in \mathcal{T}_\Delta(Q_N \times \{\bot\})$ after Example 18: each subtree in $Q_N(X)$ or $Q_N \times \{\bot\}$ is replaced by $\bot$.

Intuitively, if M, N, φ, and ψ are as in Lemma 26, then we expect the pattern $A_N\Phi$ to be the least common prefix of all axioms $A_M(p)$ of M, or, equivalently of all trees $A_M(p)\Omega$ (with $p \in P_M$). Moreover, for every node $v \in V_\bot(A_N\Phi)$ with $A_N/v = q(x_0)$, we expect the state q of N to correspond (via ψ) to the difference tuple of these trees at the node v, i.e., the difference tuple $(A_M(\hat{p}_1)\Omega/v, \ldots, A_M(\hat{p}_n)\Omega/v)$. This intuition is proved in the next lemma.

Lemma 29. *Let M, N be equivalent canonical dtlas such that M is a dtpla and N is a dtop. Let φ be the aheadness mapping from N to M. Then*

(1) $A_N\Phi = \sqcap\{A_M(p)\Omega \mid p \in P_M\}$, and
(2) for every $v \in V_\bot(A_N\Phi)$ and $q \in Q_N$,
*   if $A_N/v = q(x_0)$, then $\varphi(q, p) = A_M(p)\Omega/v$ for all $p \in P_M$.*

Proof. By the semantics of $N°$, $N(\bot) = A_N \Omega$ and by the semantics of $M°$, $M(p) = A_M(p)\Omega$ for every $p \in P_M$. Hence by Lemma 24(1), 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 $N(\bot)/v = A_N\Omega/v = \langle q, \bot \rangle$ and so by Lemma 24(2), with $C = \bot$, $\varphi(q, p) = M(p)/v = A_M(p)\Omega/v$ for every $p \in P_M$. □

Example 30. For M and N of Example 8, we have $A_M(p_a) = q^a_{80}(x_0)$, $A_M(p_b) = q^b_{30}(x_0)$, and $A_N = q_{30}(x_0)$. Thus $A_N\Phi = \bot$ and $A_M(p_a)\Omega \sqcap A_M(p_b)\Omega = \langle q^a_{80}, p_a \rangle \sqcap \langle q^b_{30}, p_b \rangle = \bot$. Moreover, for $v = \varepsilon$, we have $\varphi(q_{30}, p_a) = \langle q^a_{80}, p_a \rangle$ and $\varphi(q_{30}, p_b) = \langle q^b_{30}, p_b \rangle$, cf. Example 27.

Suppose now that we change the axioms of M into $A_M(p_a) = \sigma(q^a_{80}(x_0))$ and $A_M(p_b) = \sigma(q^b_{30}(x_0))$. Obviously, an equivalent dtop is obtained by changing the axiom of N into $A_N = \sigma(q_{30}(x_0))$. It should be clear that the aheadness mapping does not change. Now $A_N\Phi = \sigma(\bot) = A_M(p_a)\Omega \sqcap A_M(p_b)\Omega$, and for $v = 1$ we have the same statement as above.

Next, suppose that we change the axioms of M into $A_M(p_a) = \tau(\sigma(q^a_{80}(x_0)))$ and $A_M(p_b) = \tau(\tau(q^b_{30}(x_0)))$. For this M, an equivalent canonical dtop N exists, of which we will not give the details. It should be clear that its axiom is $A_N = \tau(q_0(x_0))$ for some state q_0, and that for a context $C = w(\bot)$ with $|w| \geq 80$ it computes $N(C) = \langle q_{z_1,z_2}, \bot \rangle$, where z_1 and z_2 are the first 80 and 30 symbols of w, respectively; for the states q_{z_1,z_2}, it has rules $q_{z_1,z_2}(a) \to \sigma z_1(a)$ and $q_{z_1,z_2}(b) \to \tau z_2(b)$. Now $A_N\Phi = \tau(\bot) = A_M(p_a)\Omega \sqcap A_M(p_b)\Omega$ and, after computing the new aheadness mapping φ from N to M, it can be seen that for $v = 1$, $\varphi(q_0, p_a) = \sigma(\langle q^a_{80}, p_a \rangle)$ and $\varphi(q_0, p_b) = \tau(\langle q^b_{30}, p_b \rangle)$. □

Finally we show how to compute the rules of $\mathsf{td}(M)$. Again, we need some terminology. In view of the next subsection, we consider an arbitrary set S of "states" instead of the set of states Q_N of an equivalent canonical dtop N (which, of course, motivates the terminology).

Let M be a canonical dtla, S be a set and $\varphi : S \times P_M \to \mathcal{T}_\Delta(Q_M \times P_M)$ be 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 $q \in S$ and $p \in P_M$. Then we define for every $q \in S$, $a \in \Sigma^{(k)}$, 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 $p = \delta_M(a, p_1, \ldots, p_k)$. Note that by the above condition on $\varphi(q, p)$, the right-hand side of the equation is always defined and is a tree in $\mathcal{T}_\Delta(Q_M(X_k))$; moreover, if it has a subtree $q'(x_i)$, then $\rho_M(q') = p_i$. Thus, $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k) \in \mathcal{T}_\Delta(Q_M(X_k))$ and $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega \in \mathcal{T}_\Delta(\{\langle \bar{q}, p \rangle \mid \bar{q} \in Q_M, \rho_M(\bar{q}) = p\})$.

Now let M, N, φ (with $S = Q_N$) and ψ be as in Lemma 26, and consider $q \in Q_N$ and $a \in \Sigma^{(k)}$. Intuitively, in view of Lemma 22, we expect the pattern $\mathsf{rhs}_N(q, a)\Phi$ to be the least common prefix of all "right-hand sides" $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega$ with $p_1, \ldots, p_k \in P_M$. Moreover, in view of Lemma 24, we expect for every node $v \in V_\bot(\mathsf{rhs}_N(q, a)\Phi)$ with $\mathsf{rhs}_N(q, a)/v = q'(x_i)$, that $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v$ is a tree in $\mathcal{T}_\Delta(Q_M(\{x_i\}))$ that does not depend on p_j for $j \neq i$, and we expect the state q' of N to correspond (via ψ) to the difference tuple of the trees $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega$ at the node v, for varying p_i. More precisely, for every $p \in P_M$, the set of all trees

28

$\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)/v$ with $p_i = p$ is a singleton $\{t_p\}$ with $t_p \in \mathcal{T}_\Delta(Q_M(\{x_i\}))$, and q' corresponds to the difference tuple of the trees $t_p\Omega$ at v, i.e., the difference tuple $(t_{\hat{p}_1}\Omega/v,\ldots,t_{\hat{p}_n}\Omega/v)$. These intuitions are proved in the next lemma. Examples are given after the proof.

Lemma 31. *Let M, N be equivalent canonical dtlas such that M is a dtpla and N is a dtop. Let φ be the aheadness mapping from N to M. Let $q \in Q_N$ and $a \in \Sigma^{(k)}$. Then*

(1) $\mathsf{rhs}_N(q,a)\Phi = \sqcap\{\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Omega \mid p_1,\ldots,p_k \in P_M\}$, *and*
(2) for every $v \in V_\perp(\mathsf{rhs}_N(q,a)\Phi)$, $q' \in Q_N$, and $i \in [k]$,
    if $\mathsf{rhs}_N(q,a)/v = q'(x_i)$, *then*
    (a) $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)/v \in \mathcal{T}_\Delta(Q_M(\{x_i\}))$ *for all $p_1,\ldots,p_k \in P_M$, and*
    (b) $\varphi(q',p_i) = \mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Omega/v$ *for all $p_1,\ldots,p_k \in P_M$.*

Remark 32. Before proving this lemma, we observe that items (a) and (b) determine the number i uniquely (and hence also the state q'). Indeed, suppose that (a) and (b) also hold for some q'' and some $i' \neq i$. Then it follows from (a) that every tree $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)/v$ is in $\mathcal{T}_\Delta$ (and hence equals $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Omega/v$). Moreover, it follows from (a) and (b) together that $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)/v$ does not depend on p_j for $j \neq i$ *and* does not depend on p_j for $j \neq i'$. In other words, all $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)/v$ are the same, which contradicts item (1) and the fact that $v \in V_\perp(\mathsf{rhs}_N(q,a)\Phi)$ (by Lemma 1).

Proof. (1) We first show that $\mathsf{rhs}_N(q,a)\Phi \sqsubseteq \mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)$ for all $p_1,\ldots,p_k \in P_M$. Consider arbitrary trees $s_1,\ldots,s_k \in \mathcal{T}_\Sigma$ with $s_i \in [\![p_i]\!]_M$ for $i \in [k]$, and let $s = a(s_1,\ldots,s_k)$. By Lemma 2(4),

$$q_N(s) = \mathsf{rhs}_N(q,a)\Psi_N$$

where $\Psi_N = [\bar{q}(x_i) \leftarrow \bar{q}_N(s_i) \mid \bar{q} \in Q_N, i \in [k]]$. On the other hand, by Lemma 22 and Lemma 2(4),

$$q_N(s) = \varphi(q,p)[\langle\bar{q},p\rangle \leftarrow \mathsf{rhs}_M(\bar{q},a,p_1,\ldots,p_k) \mid \bar{q} \in Q_M]\Psi_M$$

where $p = \delta_M(a,p_1,\ldots,p_k)$ and $\Psi_M = [\bar{q}(x_i) \leftarrow \bar{q}_M(s_i) \mid \bar{q} \in Q_M, i \in [k]]$. Thus

$$\mathsf{rhs}_N(q,a)\Psi_N = \mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Psi_M. \tag{3}$$

We have to prove that every node of $\mathsf{rhs}_N(q,a)$ with label $d \in \Delta$ is also a node of $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)$, with the same label. We prove this by induction on the length of the nodes of $\mathsf{rhs}_N(q,a)$, similar to the proof of Claim 1 of Lemma 21. Let v be a node of $\mathsf{rhs}_N(q,a)$ with label $d \in \Delta$. By induction, v is a node of $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)$. Now suppose that v does not have label d in $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)$. Then Equation (3) implies that $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)/v = \bar{q}(x_i)$ and that $\bar{q}_M(s_i)$ has root label d. Since that holds for arbitrary $s_i \in [\![p_i]\!]_M$, it contradicts the fact that M is earliest. Note that since M is la-uniform, $[\![p_i]\!]_M$ is the domain of $[\![\bar{q}]\!]_M$.

Now let $\Pi = \{\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Omega \mid p_1,\ldots,p_k \in P_M\}$. By the above, $\mathsf{rhs}_N(q,a)\Phi \sqsubseteq \sqcap\Pi$. It remains to prove for every node v of $\mathsf{rhs}_N(q,a)\Phi$ with label $\perp$

that v has label $\perp$ in $\sqcap \Pi$ too. Let $\mathsf{rhs}_N(q,a)/v = \bar{q}(x_i)$. Assume now that v has label $d \in \Delta$ in $\sqcap \Pi$. Then v has label d in every tree $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)$. Equation (3) then implies that $\bar{q}_N(s_i)$ has root label d for every $s_i \in [\![p_i]\!]_M$. Since this holds for every $p_i \in P_M$, $\bar{q}_N(s)$ has root label d for every $s \in \mathcal{T}_\Sigma$, which contradicts the fact that N is earliest.

(2) Let $v \in V_\perp(\mathsf{rhs}_N(q,a)\Phi)$ with $\mathsf{rhs}_N(q,a)/v = q'(x_i)$, and let $p_1,\ldots,p_k \in P_M$. Before proving (a) and (b) for $p_1,\ldots,p_k$, we prove the following claim.

Claim. Let $s_j \in [\![p_j]\!]_M$ for every $j \in [k], j \neq i$. Then

$$\varphi(q',p_i) = \mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Psi_{iM}/v$$

where Ψ_{iM} abbreviates $[\bar{q}(x_j) \leftarrow \bar{q}_M(s_j) \mid \bar{q} \in Q_M, j \in [k], j \neq i]\Omega$.

Proof of Claim. We first observe that the effect of Ψ_{iM} is to replace every $\bar{q}(x_j)$ with $j \neq i$ by a tree in $\mathcal{T}_\Delta$ (viz. the tree $\bar{q}_M(s_j)$) and, through Ω, every $\bar{q}(x_i)$ by $\langle \bar{q}, \rho_M(\bar{q}) \rangle$.

Let Ψ_{iN} be defined similar to Ψ_{iM}, i.e., Ψ_{iN} abbreviates $[\bar{q}(x_j) \leftarrow \bar{q}_N(s_j) \mid \bar{q} \in Q_N, j \in [k], j \neq i]\Omega$. Let $C \in \mathcal{C}_\Sigma$ and $u \in \mathbb{N}_+^*$ be such that $N(C)/u = \langle q, \perp \rangle$, by Lemma 5, and consider the context $C_i = C[a(s_1,\ldots,s_{i-1},\perp,s_{i+1},\ldots,s_k)]$. By Lemma 4 and Lemma 2(4), $N(C_i) = N(C)[\langle \bar{q}, \perp \rangle \leftarrow \mathsf{rhs}_N(\bar{q},a) \mid \bar{q} \in Q_N]\Psi_{iN}$. Hence

$$N(C_i)/u = \mathsf{rhs}_N(q,a)\Psi_{iN}. \tag{4}$$

Similarly, $M(C_i[p_i]) = M(C[p])[\langle \bar{q}, p \rangle \leftarrow \mathsf{rhs}_M(\bar{q},a,p_1,\ldots,p_k) \mid \bar{q} \in Q_M]\Psi_{iM}$, where $p = \delta_M(a,p_1,\ldots,p_k)$. Hence, since $M(C[p])/u = \varphi(q,p)$ by Lemma 24(2), we obtain that

$$M(C_i[p_i])/u = \mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Psi_{iM}. \tag{5}$$

Since $\mathsf{rhs}_N(q,a)/v = q'(x_i)$, Equation (4) implies that $N(C_i)/uv = \langle q', \perp \rangle$, and so $\varphi(q',p_i) = M(C_i[p_i])/uv$ by Lemma 24(2). Then Equation (5) gives the required result. This proves the Claim.

(a) We choose trees $s_j \in [\![p_j]\!]_M$ for $j \in [k], j \neq i$. By the Claim, we have that $\varphi(q',p_i) = \mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Psi_{iM}/v$. Suppose that there exists $w \in \mathbb{N}_+^*$ such that $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)/vw = \tilde{q}(x_j)$ with $j \neq i$. Let $s'_j \in [\![p_j]\!]_M$ be such that $\tilde{q}_M(s'_j) \neq \tilde{q}_M(s_j)$; it exists because M is earliest (and so $\tilde{q}_M(s'_j)$ and $\tilde{q}_M(s_j)$ can even have different root symbols). Then, again by the Claim, we have that $\varphi(q',p_i) = \mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Psi'_{iM}/v$ where Ψ'_{iM} is obtained from Ψ_{iM} by changing s_j into s'_j. But the trees $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Psi_{iM}/v$ and $\mathsf{rhs}_{M,\varphi}(q,a,p_1,\ldots,p_k)\Psi'_{iM}/v$ are different: their subtrees at node w are $\tilde{q}_M(s_j)$ and $\tilde{q}_M(s'_j)$, respectively. That is a contradiction.

(b) follows immediately from (a) and the Claim. $\square$

We observe that for $k \in \{0,1\}$, the statement of Lemma 31 can be simplified. For $k = 0$, it just states that $\mathsf{rhs}_N(q,a) = \mathsf{rhs}_{M,\varphi}(q,a)$. For $k = 1$, item (2) of the lemma just states that for every $v \in V_\perp(\mathsf{rhs}_N(q,a)\Phi)$ and $q' \in Q_N$, if $\mathsf{rhs}_N(q,a)/v = q'(x_1)$, then $\varphi(q',p) = \mathsf{rhs}_{M,\varphi}(q,a,p)\Omega/v$ for all $p \in P_M$.

Example 33. We consider again M and N of Example 8. Since the rank of every symbol in Σ is 0 or 1, the above observation applies.

Let us first check Lemma 31 for $q_1 \in Q_N$ and for $\gamma \in \{\sigma, \tau\}$. To do this we have to compute $\mathsf{rhs}_{M,\varphi}(q_1, \gamma, p)$ for $p \in \{p_a, p_b\}$. Since $\varphi(q_1, p_a) = \langle q_{51}^a, p_a \rangle$, we obtain that $\mathsf{rhs}_{M,\varphi}(q_1, \gamma, p_a) = \mathsf{rhs}_M(q_{51}^a, \gamma, p_a) = \gamma(q_{50}^a(x_1))$. Since $\varphi(q_1, p_b) = \langle q_1^b, p_b \rangle$, we obtain that $\mathsf{rhs}_{M,\varphi}(q_1, \gamma, p_b) = \mathsf{rhs}_M(q_1^b, \gamma, p_b) = \gamma(b)$. Consequently $\mathsf{rhs}_{M,\varphi}(q_1, \gamma, p_a)\Omega \sqcap \mathsf{rhs}_{M,\varphi}(q_1, \gamma, p_b)\Omega = \gamma(\bot)$. Since $\mathsf{rhs}_N(q_1, \gamma) = \gamma(q_\varepsilon(x_1))$, we have that $\mathsf{rhs}_N(q_1, \gamma)\Phi = \gamma(\bot)$, which shows item (1) of Lemma 31. To show item (2), we have to check for $v = 1$ that $\varphi(q_\varepsilon, p) = \mathsf{rhs}_{M,\varphi}(q_1, \gamma, p)\Omega/v$ for $p \in \{p_a, p_b\}$. This is true because $\varphi(q_\varepsilon, p_a) = \langle q_{50}^a, p_a \rangle$ and $\varphi(q_\varepsilon, p_b) = b$.

Next we check Lemma 31 for $q_z \in Q_N$ with $|z| < 50$ and for $\gamma \in \{\sigma, \tau\}$, in the same way. We first compute $\mathsf{rhs}_{M,\varphi}(q_z, \gamma, p)$ for $p \in \{p_a, p_b\}$. Since $\varphi(q_z, p_a) = z(\langle q_{50-|z|}^a, p_a \rangle)$, we obtain that $\mathsf{rhs}_{M,\varphi}(q_z, \gamma, p_a) = z(\mathsf{rhs}_M(q_{50-|z|}^a, \gamma, p_a))$, which equals $z\gamma(q_{49-|z|}^a(x_1))$ for $|z| < 49$ and $z\gamma(a)$ for $|z| = 49$. And since $\varphi(q_z, p_b) = b$, we get that $\mathsf{rhs}_{M,\varphi}(q_z, \gamma, p_b) = b$. Thus $\mathsf{rhs}_{M,\varphi}(q_z, \gamma, p_a)\Omega \sqcap \mathsf{rhs}_{M,\varphi}(q_z, \gamma, p_b)\Omega = \bot$. Since $\mathsf{rhs}_N(q_z, \gamma) = q_{z\gamma}(x_1)$, we have that $\mathsf{rhs}_N(q_z, \gamma)\Phi = \bot$, which shows item (1). To show item (2), we have to check that $\varphi(q_{z\gamma}, p) = \mathsf{rhs}_{M,\varphi}(q_z, \gamma, p)\Omega$ for $p \in \{p_a, p_b\}$. For $p = p_a$ that follows from the fact that $\varphi(q_{z\gamma}, p_a)$ equals $z\gamma(\langle q_{50-|z\gamma|}^a, p_a \rangle)$ for $|z\gamma| < 50$ and $z\gamma(a)$ for $|z\gamma| = 50$. For $p = p_b$, from $\varphi(q_{z\gamma}, p_b) = b$.

We now check Lemma 31 for $q_z \in Q_N$ with $|z| \leq 50$ and for $a \in \Sigma^{(0)}$. For $|z| < 50$, we have $\mathsf{rhs}_{M,\varphi}(q_z, a) = z(\mathsf{rhs}_M(q_{50-|z|}^a, a)) = z(a) = \mathsf{rhs}_N(q_z, a)$. For $|z| = 50$, since $\varphi(q_z, p_a) = z(a)$, we have $\mathsf{rhs}_{M,\varphi}(q_z, a) = z(a) = \mathsf{rhs}_N(q_z, a)$.

To illustrate Lemma 31 also for the case where $k \geq 2$, we extend M with the binary input symbols ℓ and r (standing for 'left' and 'right'). We extend δ_M such that $\delta_M(\ell, p, p') = p$ and $\delta_M(r, p', p) = p$ for all $p, p' \in \{p_a, p_b\}$, and we extend R_M with the rules $q_i^y(\ell(x_1 : p_y, x_2 : p')) \to q_i^y(x_1)$ and $q_i^y(r(x_1 : p', x_2 : p_y)) \to q_i^y(x_2)$ for all $(y, i) \in (\{a\} \times [80]) \cup (\{b\} \times [30])$ and $p' \in \{p_a, p_b\}$. Intuitively, the new symbols ℓ and r are signposts that determine a unique path through the input tree, and M processes the unary symbols on that path as before. To obtain a dtop equivalent to M, we extend N with the rules $q(\ell(x_1, x_2)) \to q(x_1)$ and $q(r(x_1, x_2)) \to q(x_2)$ for every $q \in Q_N$. It is easy to see that this extension does not change the aheadness mapping φ from N to M.

Let us now check Lemma 31 for $q_z \in Q_N$ with $|z| < 50$ and for the new symbol r. For $p' \in \{p_a, p_b\}$, we obtain $\mathsf{rhs}_{M,\varphi}(q_z, r, p', p_a) = z(\mathsf{rhs}_M(q_{50-|z|}^a, r, p', p_a)) = z(q_{50-|z|}^a(x_2))$ and $\mathsf{rhs}_{M,\varphi}(q_z, r, p', p_b) = b$. Thus their largest common prefix is $\bot$. Since $\mathsf{rhs}_N(q_z, r) = q_z(x_2)$, this shows item (1). For $q' = q_z$ and $i = 2$, item (2)(a) clearly holds because the trees $z(q_{50-|z|}^a(x_2))$ and b do not contain the variable x_1. Item (2)(b) also holds because $\varphi(q_z, p_a) = z(\langle q_{50-|z|}^a, p_a \rangle) = \mathsf{rhs}_{M,\varphi}(q_z, r, p', p_a)\Omega$ and $\varphi(q_z, p_b) = b = \mathsf{rhs}_{M,\varphi}(q_z, r, p', p_b)\Omega$ for every $p' \in \{p_a, p_b\}$.

Finally, we consider q_z and r with $|z| = 50$. We now have $\mathsf{rhs}_{M,\varphi}(q_z, r, p', p_a) = z(a)$ and $\mathsf{rhs}_{M,\varphi}(q_z, r, p', p_b) = b$ for every $p' \in \{p_a, p_b\}$, and we have $\varphi(q_z, p_a) = z(a)$ and $\varphi(q_z, p_b) = b$. From this it is easy to see that items (1) and (2) hold, as in the previous paragraph. With respect to item (2)(a) we note that in this case the trees $\mathsf{rhs}_{M,\varphi}(q_z, \ell, p_a, p')$ and $\mathsf{rhs}_{M,\varphi}(q_z, \ell, p_b, p')$ do not contain the variable x_2 either. $\square$

The three Lemmas 26, 29, and 31 show that every canonical dtpla M that is equivalent to a dtop, is equivalent to a unique canonical dtop $\mathsf{td}(M)$, modulo a renaming of states. That will be stated formally in Corollary 39 in the next subsection.

6.2 The Algorithm

Let $M = (Q, \Sigma, \Delta, R, A, P, \delta)$ be a canonical dtpla and let $P = \{\hat{p}_1, \ldots, \hat{p}_n\}$. Our aim in this subsection is to construct the canonical dtop $N = \mathsf{td}(M)$ equivalent to M, if it exists, based on the characterizations in Lemmas 26, 29, and 31. By Lemma 26, we may assume that the states of N are difference tuples of M, and that the aheadness mapping from N to M projects these tuples on their components. This leads to the following definitions.

We define the set S_M of *potential states* to consist of all n-tuples $(t_1, \ldots, t_n) \in \mathcal{T}_\Delta(Q \times P)^n$ such that $t_i \in \mathcal{T}_\Delta(\{\langle q, \hat{p}_i \rangle \mid q \in Q, \rho(q) = \hat{p}_i\})$ for every $i \in [n]$. Note that $\mathsf{diftup}(M) \subseteq S_M$. Moreover, we define the *projection mapping* $\varphi_M : S_M \times P \to \mathcal{T}_\Delta(Q \times P)$ such that, for every $(t_1, \ldots, t_n) \in S_M$ and $i \in [n]$, $\varphi((t_1, \ldots, t_n), \hat{p}_i) = t_i$. In what follows we will denote S_M and φ_M just by S and φ, to simplify the notation; it will always be clear to which dtpla they belong.

The next two definitions are directly based on Lemmas 29 and 31, respectively. For $t \in \mathcal{T}_\Delta(S(X))$ we define the pattern $t\Phi = t[q(x_i) \leftarrow \bot \mid q \in S, i \in \mathbb{N}]$.

Definition 34. *The* potential axiom $A_{M,\varphi}$ *is the unique tree in* $\mathcal{T}_\Delta(S(\{x_0\}))$ *such that*

(1) $A_{M,\varphi}\Phi = \sqcap\{A(p)\Omega \mid p \in P\}$, *and*
(2) *for every* $v \in V_\bot(A_{M,\varphi}\Phi)$ *and* $q \in S$,
    if $A_{M,\varphi}/v = q(x_0)$, *then* $\varphi(q, p) = A(p)\Omega/v$ *for all* $p \in P$.

Note that $A_{M,\varphi}$ exists because $A(p)\Omega \in \mathcal{T}_\Delta(\{\langle \bar{q}, p \rangle \mid \bar{q} \in Q, \rho(\bar{q}) = p\})$ for every $p \in P$. Obviously, $A_{M,\varphi}$ can be computed from M.

For the next definition, we recall that $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)$ was defined before Lemma 31. It is a tree in $\mathcal{T}_\Delta(Q(X_k))$, more precisely, in $\mathcal{T}_\Delta(\{\bar{q}(x_i) \mid \bar{q} \in Q, \rho(\bar{q}) = p_i\})$.

Definition 35. *Let* $q \in S$ *and* $a \in \Sigma^{(k)}$. *The* potential right-hand side $\mathsf{rhs}_{M,\varphi}(q, a)$ *is the tree in* $\mathcal{T}_\Delta(S(X_k))$ *such that*

(1) $\mathsf{rhs}_{M,\varphi}(q, a)\Phi = \sqcap\{\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega \mid p_1, \ldots, p_k \in P\}$, *and*
(2) *for every* $v \in V_\bot(\mathsf{rhs}_{M,\varphi}(q, a)\Phi)$, $q' \in S$, *and* $i \in [k]$,
    if $\mathsf{rhs}_{M,\varphi}(q, a)/v = q'(x_i)$, *then*
    (a) $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v \in \mathcal{T}_\Delta(Q(\{x_i\}))$ *for all* $p_1, \ldots, p_k \in P$, *and*
    (b) $\varphi(q', p_i) = \mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega/v$ *for all* $p_1, \ldots, p_k \in P$.

For $k = 0$ this definition (re)defines $\mathsf{rhs}_{M,\varphi}(q, a)$ consistently to be equal to the tree $\mathsf{rhs}_{M,\varphi}(q, a)$ as defined before Lemma 31.

We also observe here that for $k \geq 1$ (more precisely, for $k \geq 2$), $\mathsf{rhs}_{M,\varphi}(q, a)$ may not exist. However, if it exists then it is unique, which can be shown as follows (cf. Remark 32). First, by item (1), $\mathsf{rhs}_{M,\varphi}(q, a)\Phi$ always exists and is unique. Second, let $v \in V_\bot(\mathsf{rhs}_{M,\varphi}(q, a)\Phi)$ and assume that items (2)(a) and (2)(b) hold for q' and i, and also

for q'' and j. Suppose that $i \neq j$. Then (a) implies that every $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v$ is in $\mathcal{T}_\Delta$ (and hence equals $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega/v$). It now follows from (b) that they are all the same tree in $\mathcal{T}_\Delta$. Indeed, for all $p_1, \ldots, p_k, p'_1, \ldots, p'_k \in P$ we have $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v = \mathsf{rhs}_{M,\varphi}(q, a, p'_1, \ldots, p'_k)/v$ because both $\varphi(q', p_i)$ and $\varphi(q'', p'_j)$ are equal to $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_{j-1}, p'_j, p_{j+1}, \ldots, p_k)/v$. This contradicts item (1) and the fact that $v \in V_\perp(\mathsf{rhs}_{M,\varphi}(q, a)\Phi)$, by Lemma 1. Hence $i = j$. It now follows from item (b) that $\varphi(q', p) = \varphi(q'', p)$ for every $p \in P$, and so $q' = q''$.

In the rest of this subsection we will denote the tree $\sqcap\{\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega \mid p_1, \ldots, p_k \in P\}$ by $\mathsf{rhs}_{M,\varphi}(q, a)\Phi$, even if $\mathsf{rhs}_{M,\varphi}(q, a)$ does not exist.

Lemma 36. *For $q \in S$ and $a \in \Sigma^{(k)}$, $\mathsf{rhs}_{M,\varphi}(q, a)$ exists if and only if for every $v \in V_\perp(\mathsf{rhs}_{M,\varphi}(q, a)\Phi)$ there exists $i \in [k]$ such that*

(a) $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v \in \mathcal{T}_\Delta(Q(\{x_i\}))$ *for all $p_1, \ldots, p_k \in P$, and*
(b) *for all $p_1, \ldots, p_k, p'_1, \ldots, p'_k \in P$, if $p_i = p'_i$ then $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega/v = \mathsf{rhs}_{M,\varphi}(q, a, p'_1, \ldots, p'_k)\Omega/v$.*

Proof. The only-if direction is obvious. For the if-direction, let $v \in V_\perp(\mathsf{rhs}_{M,\varphi}(q, a)\Phi)$ and let $i \in [k]$ satisfy items (a) and (b). Then we define $\mathsf{rhs}_{M,\varphi}(q, a)/v = q'(x_i)$ where, for every $p \in P$, $\varphi(q', p) = \mathsf{rhs}_{M,\varphi}(q, a, p, \ldots, p)\Omega/v$. Note that $q' \in S$ because $\mathsf{rhs}_{M,\varphi}(q, a, p, \ldots, p)\Omega$ is in $\mathcal{T}_\Delta(\{\langle \bar{q}, p \rangle \mid \bar{q} \in Q, \rho(\bar{q}) = p\})$. This defines $\mathsf{rhs}_{M,\varphi}(q, a)$ such that it satisfies the requirements of Definition 35. $\square$

This lemma shows that $\mathsf{rhs}_{M,\varphi}(q, a)$ always exists for $k = 1$ (and thus, it always exists for $k \in \{0, 1\}$).

We now show that $\mathsf{rhs}_{M,\varphi}(q, a)$ can be computed from M, q, and a, as follows. First one computes $\mathsf{rhs}_{M,\varphi}(q, a)\Phi$ by Definition 35(1). Then one determines whether or not $\mathsf{rhs}_{M,\varphi}(q, a)$ exists by Lemma 36. Thus, one computes for every node $v \in V_\perp(\mathsf{rhs}_{M,\varphi}(q, a)\Phi)$ a number $i = i_v \in [k]$ that satisfies (a) and (b) of Lemma 36 (provided such numbers exist). If this is successful, then, according to the proof of Lemma 36, one defines, for every such v, $\mathsf{rhs}_{M,\varphi}(q, a)/v = q'(x_{i_v})$ where $\varphi(q', p) = \mathsf{rhs}_{M,\varphi}(q, a, p, \ldots, p)\Omega/v$ for all $p \in P$.

Definition 37. *Let M be a canonical dtpla. A dtop N is* associated *with M if*

  - *N is complete,*
  - *N has input and output alphabets Σ and Δ,*
  - *$Q_N \subseteq S$,*
  - *$A_N = A_{M,\varphi}$, and*
  - *$\mathsf{rhs}_N(q, a) = \mathsf{rhs}_{M,\varphi}(q, a)$ for every $q \in Q_N$ and $a \in \Sigma^{(k)}$.*

For $h \in \mathbb{N}$, N is associated *with M and h if, moreover, $\mathsf{ht}(q) \leq h$ for every $q \in Q_N$.*

The results of Section 6.1 can now be summarized as follows. Recall from Section 4 that a number $h(M) \in \mathbb{N}$ is a *difference bound* for M if the following holds: if $\mathsf{diff}(M)$ is finite, then $\mathsf{lubdiff}(M) \leq h(M)$.

Corollary 38. *Let M be a canonical dtpla, let N be a canonical dtop equivalent to M, and let $h(M)$ be a difference bound for M. Then, modulo a renaming of states, $Q_N = \mathsf{diftup}(M)$ and N is associated with M and $h(M)$.*

Proof. Let $\bar{N}$ be a canonical dtop equivalent to M. As observed after Lemma 26, by renaming the states of $\bar{N}$ to difference tuples of M, according to the mapping ψ of Lemma 26, we obtain a canonical dtop N such that $Q_N = \mathsf{diftup}(M) \subseteq S$, with the projection mapping φ as aheadness mapping (or more precisely, the restriction of φ to Q_N). Since N is la-uniform, it is complete. The 4th and 5th condition of Definition 37 hold by Lemmas 29 and 31, respectively. Since $h(M)$ is a difference bound for M, the height of every $q \in Q_N$ is $\leq h(M)$ by Lemma 17. $\qquad\square$

This corollary has two corollaries. The first shows that, as already observed in Section 6.1, if M is equivalent to a dtop, then it is equivalent to a unique (reduced) canonical dtop $\mathsf{td}(M)$. The second shows that, to find a dtop equivalent to M, it suffices to consider all dtops associated with M and $h(M)$, where $h(M)$ is a difference bound for M.

Corollary 39. *A total dtpla M is equivalent to at most one canonical dtop, modulo a renaming of states.*

Proof. By Lemma 13 and Theorem 15 we may assume that M is canonical. If N and N' are canonical dtops equivalent to M, renamed according to Lemma 26, then $Q_N = Q_{N'}$ and N, N' are associated with M and $h(M)$, by Corollary 38. The 4th and 5th condition of Definition 37 then imply that $N = N'$. $\qquad\square$

Corollary 40. *Let M be a canonical dtpla and let $h(M)$ be a difference bound for M. Then M is equivalent to a dtop if and only if there exists a dtop N, associated with M and $h(M)$, such that M is equivalent to N.*

Proof. If M is equivalent to a dtop, then it is equivalent to a canonical dtop by Theorem 15. Hence, by Corollary 38, it is equivalent to a dtop that is associated with M and $h(M)$. $\qquad\square$

The last corollary already allows us to conclude our first main result, for the case that M is canonical.

Lemma 41. *It is decidable for a given canonical dtpla M and a given difference bound $h(M)$ for M whether there exists a dtop N equivalent to M, and if so, such a dtop N can be constructed.*

Proof. By Corollary 40, it suffices to check for every dtop that is associated with M and $h(M)$, whether it is equivalent to M (and equivalence is decidable by [13]; see also [9, Corollary 19]). As already observed, $A_{M,\varphi}$ and $\mathsf{rhs}_{M,\varphi}(q, a)$ can be computed from M. Thus, for every subset Q' of $\{q \in S \mid \mathsf{ht}(q) \leq h(M)\}$ (of which there are only finitely many), we can compute the unique dtop N associated with M (and $h(M)$) such that $Q_N = Q'$ (if it exists), and decide its equivalence to M. Note that such an N may not exist, either because some $\mathsf{rhs}_{M,\varphi}(q, a)$ does not exist (which can be determined by Lemma 36) or because some potential state (in S) that occurs in $A_{M,\varphi}$ or some $\mathsf{rhs}_{M,\varphi}(q, a)$, does not belong to Q'. $\qquad\square$

However, it should be clear that the algorithm in the above proof is inelegant and not even feasible for small examples. Unfortunately, we do not know whether it is decidable if $\mathsf{diftup}(M)$ is finite, and whether it can be computed when it is finite. In that case it would suffice to consider $Q' = \mathsf{diftup}(M)$, and we would not even need the difference bound $h(M)$. Nevertheless, we will show in the remainder of this subsection that, with a given difference bound $h(M)$, there is a straighforward algorithm to compute the (unique) dtop associated with M and $h(M)$, if it exists. Thus, there is no need to search the space of subsets of $\{q \in S \mid \mathsf{ht}(q) \le h(M)\}$. Moreover, equivalence tests are not needed, which we now prove.

Lemma 42. *Let M be a canonical dtpla and let N be a dtop that is associated with M. Then N is equivalent to M.*

Proof. We first show, by structural induction on $s \in \mathcal{T}_\Sigma$, that if $\delta_M(s) = p$, then $q_N(s) = \varphi(q,p)[\langle \tilde{q}, p \rangle \leftarrow \tilde{q}_M(s) \mid \tilde{q} \in Q_M]$ for all $q \in Q_N$, cf. Lemma 22. Let $s = a(s_1, \ldots, s_k)$ with $\delta_M(s_i) = p_i$ for $i \in [k]$ and $\delta_M(a, p_1, \ldots, p_k) = p$. From Lemma 2(4) we obtain that $q_N(s) = \mathsf{rhs}_N(q,a)[\bar{q}(x_i) \leftarrow \bar{q}_N(s_i) \mid \bar{q} \in Q_N, i \in [k]]$. By induction, $\bar{q}_N(s_i) = \varphi(\bar{q}, p_i)[\langle \tilde{q}, p_i \rangle \leftarrow \tilde{q}_M(s_i) \mid \tilde{q} \in Q_M]$. Now consider any node v such that $\mathsf{rhs}_N(q,a)/v = \bar{q}(x_i)$. By Definition 35(2) and the fact that $\mathsf{rhs}_N(q,a) = \mathsf{rhs}_{M,\varphi}(q,a)$, we have $\varphi(\bar{q}, p_i) = \mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Omega/v$ where $\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v$ is in $\mathcal{T}_\Delta(Q_M(\{x_i\}))$. Consequently

$$\bar{q}_N(s_i) = (\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v)[\tilde{q}(x_i) \leftarrow \tilde{q}_M(s_i) \mid \tilde{q} \in Q_M]$$
$$= (\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v)\Psi_M$$

where $\Psi_M = [\tilde{q}(x_j) \leftarrow \tilde{q}_M(s_j) \mid \tilde{q} \in Q_M, j \in [k]]$. Hence $q_N(s) = t\Psi_M$ with

$$t = \mathsf{rhs}_N(q,a)\Phi[\mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v_1, \ldots, \mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)/v_r]$$

where $v_1, \ldots, v_r$ are the occurrences of $\perp$ in $\mathsf{rhs}_N(q,a)\Phi$, from left to right. By Definition 35(1), $\mathsf{rhs}_N(q,a)\Phi \sqsubseteq \mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)$ from which it follows that $t = \mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)$. So, $q_N(s) = \mathsf{rhs}_{M,\varphi}(q, a, p_1, \ldots, p_k)\Psi_M = \varphi(q,p)[\langle \tilde{q}, p \rangle \leftarrow \mathsf{rhs}_M(\tilde{q}, a, p_1, \ldots, p_k) \mid \tilde{q} \in Q_M]\Psi_M = \varphi(q,p)[\langle \tilde{q}, p \rangle \leftarrow \tilde{q}_M(s) \mid \tilde{q} \in Q_M]$ where the last equality is by Lemma 2(4).

We now show with a similar proof that $N(s) = M(s)$ for all $s \in \mathcal{T}_\Sigma$, i.e., that N is equivalent to M. By Lemma 2(3), $N(s) = A_N[q(x_0) \leftarrow q_N(s) \mid q \in Q_N]$ and $M(s) = A_M(p)[\tilde{q}(x_0) \leftarrow \tilde{q}_M(s) \mid \tilde{q} \in Q_M]$ where $p = \delta_M(s)$. By the above, $q_N(s) = \varphi(q,p)[\langle \tilde{q}, p \rangle \leftarrow \tilde{q}_M(s) \mid \tilde{q} \in Q_M]$. Now consider any node v such that $A_N/v = q(x_0)$. By Definition 34(2) and the fact that $A_N = A_{M,\varphi}$, we have $\varphi(q,p) = A_M(p)\Omega/v$ and consequently $q_N(s) = (A_M(p)/v)[\tilde{q}(x_0) \leftarrow \tilde{q}_M(s) \mid \tilde{q} \in Q_M]$. Hence $N(s) = t[\tilde{q}(x_0) \leftarrow \tilde{q}_M(s) \mid \tilde{q} \in Q_M]$ with $t = A_N\Phi[A_M(p)/v_1, \ldots, A_M(p)/v_r]$ where $v_1, \ldots, v_r$ are the occurrences of $\perp$ in $A_N\Phi$, from left to right. By Definition 34(1), $A_N\Phi \sqsubseteq A_M(p)$ and so $t = A_M(p)$ and $N(s) = A_M(p)[\tilde{q}(x_0) \leftarrow \tilde{q}_M(s) \mid \tilde{q} \in Q_M] = M(s)$. $\qquad\square$

Thus, by Corollary 40, M is equivalent to a dtop if and only if there exists a dtop that is associated with M and $h(M)$, where $h(M)$ is a difference bound for M. We now show that such a (reduced) dtop is unique.

Lemma 43. *Let M be a canonical dtpla. There exists at most one dtop that is associated with M.*

Proof. Obviously, it suffices to prove that if N and N' are both associated with M, then $Q_N \subseteq Q_{N'}$. We show that $q \in Q_{N'}$ for every $q \in Q_N$. This trivially follows from the reachability of q in N, as follows. If q occurs in the axiom A_N, then it occurs in $A_{N'}$ (because A_N and $A_{N'}$ both equal $A_{M,\varphi}$) and hence it is in $Q_{N'}$. If $\bar{q}$ occurs in $\mathrm{rhs}_N(q, a)$ and we already know that $q \in Q_{N'}$, then $\bar{q}$ occurs in $\mathrm{rhs}_{N'}(q, a)$ (because it equals $\mathrm{rhs}_N(q, a)$) and hence $\bar{q} \in Q_{N'}$. $\qquad\square$

Of course, by Corollary 38, this dtop is the unique canonical dtop $\mathrm{td}(M)$ equivalent to M. The following algorithm finds $\mathrm{td}(M)$ by constructing the unique dtop associated with M (and difference bound $h(M)$).

Algorithm 44. We now present our main algorithm that, for a given canonical dtpla M and a given difference bound for M, decides whether there exists a dtop N equivalent to M, and if so, constructs such a dtop N. Moreover, the constructed dtop N is canonical.

Input:

a canonical dtpla $M = (Q, \Sigma, \Delta, R, A, P, \delta)$ and a difference bound $h(M)$ for M.

Output:

either the unique canonical dtop N equivalent to M, or the answer 'no' if M is not equivalent to a dtop.

Let Q_N, rhs_N, A_N be variables that determine a dtop $N = (Q_N, \Sigma, \Delta, R_N, A_N)$. The variable rhs_N is a $Q_N \times \Sigma$ array with $\mathrm{rhs}_N(q, a) \in \mathcal{T}_\Delta(Q_N(X_k)) \cup \{\bot\}$ for every $q \in Q_N$ and $a \in \Sigma^{(k)}$. Then R_N consists of all rules $q(a(x_1, \ldots, x_k)) \to \mathrm{rhs}_N(q, a)$ such that $\mathrm{rhs}_N(q, a) \neq \bot$.

The instruction 'abort(no)' causes the algorithm to halt and output 'no'.

Initialization:

$A_N := A_{M,\varphi}$;
$Q_N := \{q \in S \mid q \text{ occurs in } A_N\}$;
if $\{q \in Q \mid \mathrm{ht}(q) > h(M)\} \neq \emptyset$ **then** abort(no) ;
$\mathrm{rhs}_N(q, a) := \bot$ for every $(q, a) \in Q_N \times \Sigma$;

Loop:

    while N is not complete **do**
        choose $(q, a) \in Q_N \times \Sigma$ such that $\mathrm{rhs}_N(q, a) = \bot$;
        if $\mathrm{rhs}_{M,\varphi}(q, a)$ does not exist **then** abort(no) ;
        $\mathrm{rhs}_N(q, a) := \mathrm{rhs}_{M,\varphi}(q, a)$;
        $Q_N := Q_N \cup \{q \in S \mid q \text{ occurs in } \mathrm{rhs}_N(q, a)\}$;
        if $\{q \in Q \mid \mathrm{ht}(q) > h(M)\} \neq \emptyset$ **then** abort(no)

Finalization:

output N.

The next theorem shows the correctness of the algorithm. Thus it can be viewed as an improvement of the proof of Lemma 41.

Theorem 45. *Algorithm 44 is correct.*

Proof. We first observe that the algorithm always halts. If it does not abort, then Q_N must remain stable after some time because Q_N is then always a subset of the finite set $\{q \in S \mid \mathsf{ht}(q) \leq h(M)\}$, and then N will be complete after at most $|Q_N \times \Sigma|$ repetitions of the while loop.

We now show that if the algorithm outputs N, then N is associated with M, and hence N is equivalent to M by Lemma 42. In fact, it is easy to verify that the following statements are invariants of the while loop:

- N has input and output alphabets Σ and Δ,
- $Q_N \subseteq S$,
- $A_N = A_{M,\varphi}$,
- $\mathsf{rhs}_N(q, a) = \perp$ or $\mathsf{rhs}_N(q, a) = \mathsf{rhs}_{M,\varphi}(q, a)$ for every $q \in Q_N$ and $a \in \Sigma^{(k)}$,

If N is complete, then these statements show that N is associated with M. This proves that if M is not equivalent to a dtop, then the algorithm aborts and outputs 'no'.

Next we show that if N_0 is a dtop that is associated with M and $h(M)$ (and hence is *the* dtop associated with M by Lemma 43), then the algorithm outputs N_0. To see this, we claim that the algorithm does not abort and that the statement "$Q_N \subseteq Q_{N_0}$, $R_N \subseteq R_{N_0}$, and $A_N = A_{N_0}$" is an invariant of the while loop. Clearly, after lines 1 and 2 of the Initialization, $A_N = A_{N_0}$ and hence $Q_N \subseteq Q_{N_0}$, and so line 3 does not abort because $\mathsf{ht}(q) \leq h(M)$ for every $q \in Q_{N_0}$. Since $R_N = \emptyset$ after line 4, the invariant holds after Initialization. A similar argument shows the invariance. If $(q, a) \in Q_N \times \Sigma$, then q is in Q_{N_0} and hence $\mathsf{rhs}_{N_0}(q, a) = \mathsf{rhs}_{M,\varphi}(q, a)$. So, line 2 of the while body does not abort, line 3 makes $\mathsf{rhs}_N(q, a)$ equal to $\mathsf{rhs}_{N_0}(q, a)$, line 4 updates Q_N with the states in $\mathsf{rhs}_{N_0}(q, a)$, and hence line 5 does not abort. This shows that the algorithm does not abort and hence outputs the dtop N. By the previous paragraph, N is associated with M, and hence $N = N_0$ by Lemma 43.

If M is equivalent to a dtop, then (as in the proof of Corollary 40) it is equivalent to a canonical dtop by Theorem 15, and hence, by Corollary 38, it is equivalent to a canonical dtop N_0 that is associated with M and $h(M)$. By the previous paragraph, the algorithm outputs N_0, which, by Corollary 39, is the unique canonical dtop equivalent to M. □

One might wonder whether the tests on height in Algorithm 44 are necessary, i.e., whether a difference bound for M is needed at all. It is easy to see, by a proof very similar to the one of Theorem 45, that, without the two lines that test on height, Algorithm 44 outputs the unique canonical dtop equivalent to M, if M is equivalent to a dtop, and otherwise either outputs 'no' or does not halt. Thus, although it is not guaranteed to halt, it is still a useful algorithm. The existence of such an algorithm is obvious from the fact that equivalence of dtlas is decidable: one can just enumerate all dtops N and test the equivalence of M and N. Obviously, our Algorithm 44 (without the tests on height) is more direct.

In the remainder of this subsection we consider some examples of Algorithm 44, first three examples in which it returns the answer 'no', and then an example in which it successfully returns a dtop equivalent to the given dtla.

The first example shows that, without the tests on height, Algorithm 44 may indeed not halt. In fact, in such a case it can be viewed as computing an *infinite* dtop equivalent to M. Of course, with the tests on height, Algorithm 44 outputs 'no'.

Example 46. Consider the dtla M of Example 7 with $M(\sigma^n a) = a$ and $M(\sigma^n b) = \sigma^n b$ for every $n \in \mathbb{N}$. It has $P_M = \{p_a, p_b\}$ with $\delta(a) = p_a$, $\delta(b) = p_b$, $\delta(\sigma, p_a) = p_a$, and $\delta(\sigma, p_b) = p_b$. It has $Q_M = \{q\}$, its two axioms are $A_M(p_a) = a$ and $A_M(p_b) = q(x_0)$, and its rules are $q(\sigma(x_1 : p_b)) \to \sigma(q(x_1))$ and $q(b) \to b$. In Example 18 we have seen that $\mathsf{diftup}(M) = \{(a, \sigma^n \langle q, p_b \rangle) \mid n \in \mathbb{N}\}$. It is easy to see that M is canonical.

We now run Algorithm 44 with input M, without the tests on height. Note that it cannot abort, because all symbols in Σ are nullary or unary (see the remark after Lemma 36). By Definition 34, $A_N \Phi = a \sqcap \langle q, p_b \rangle = \bot$ 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 item (1) of Definition 35, $\mathsf{rhs}_N(q_n, b) = \mathsf{rhs}_{M,\varphi}(q_n, b) = \varphi(q_n, p_b)[\langle \bar{q}, p_b \rangle \leftarrow \mathsf{rhs}_M(\bar{q}, b) \mid \bar{q} \in Q_M] = \varphi(q_n, p_b)[\langle q, p_b \rangle \leftarrow b] = \sigma^n b$. 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 that 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 (1) of Definition 35, $\mathsf{rhs}_N(q_n, \sigma)\Phi = a \sqcap \sigma^{n+1} \langle q, p_b \rangle = \bot$. Hence, $\mathsf{rhs}_N(q_n, \sigma) = q(x_1)$ for some state q (because $\sigma \in \Sigma^{(1)}$). By (2)(b) of Definition 35, $\varphi(q, p_y) = \mathsf{rhs}_{M,\varphi}(q_n, \sigma, p_y)\Omega$ for $y \in \{a, b\}$, and so $\varphi(q, p_a) = a$ and $\varphi(q, p_b) = \sigma^{n+1} \langle q, p_b \rangle$. In other words, $q = q_{n+1}$ and N has the rule $q_n(\sigma(x_1)) \to q_{n+1}(x_1)$.

This shows that the algorithm 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, this infinite dtop N is equivalent to M.

The dtla M is ultralinear and b-erasing (in fact, linear and nonerasing). We will see in the proof of Theorem 85 that it has difference bound $h(M) = 1 + 4 \cdot \mathsf{maxrhs}(M) \cdot (|Q| + 2)^2 \cdot |P|^2 = 1 + 4 \cdot 2 \cdot 3^2 \cdot 2^2 = 289$. So, with this difference bound given, Algorithm 44 (*with* the tests on height) will halt with answer 'no' when constructing q_{290}. $\qquad\square$

In the next two examples we show that Algorithm 44 can abort because some potential right-hand side $\mathsf{rhs}_{M,\varphi}(q, a)$ does not exist. According to Lemma 36 there can be two reasons for such a failure, either due to condition (a) or to condition (b). We start with (a).

Example 47. Let $\Sigma = \{\sigma^{(2)}, a^{(0)}\}$ and $\Delta = \{\sigma^{(2)}, a^{(1)}, e^{(0)}\}$, and consider the following canonical dtla M that translates every subtree $\sigma(a, s)$ and $\sigma(s, a)$ into $a(s)$. More precisely, $M(a) = e$, $M(\sigma(a, s)) = M(\sigma(s, a)) = a(M(s))$ for every $s \in \mathcal{T}_\Sigma$, and $M(\sigma(s_1, s_2)) = \sigma(M(s_1), M(s_2))$ for all $s_i \in \mathcal{T}_\Sigma$ with $\mathsf{lab}(s_i, \varepsilon) = \sigma$. Its look-ahead

automaton has two states p_a and p_σ with $\delta(a) = p_a$ and $\delta(\sigma, p_y, p_z) = p_\sigma$ for all $y, z \in \{a, \sigma\}$. It has one state q with $\rho(q) = p_\sigma$, its two axioms are $A_M(p_a) = e$ and $A_M(p_\sigma) = q(x_0)$, and its four rules are

$$q(\sigma(x_1:p_a, x_2:p_\sigma)) \to a(q(x_2)), \qquad q(\sigma(x_1:p_\sigma, x_2:p_\sigma)) \to \sigma(q(x_1), q(x_2)),$$
$$q(\sigma(x_1:p_a, x_2:p_a)) \to a(e), \qquad q(\sigma(x_1:p_\sigma, x_2:p_a)) \to a(q(x_1)).$$

We try to construct a dtop N by Algorithm 44 (assuming a large difference bound). By Definition 34, $A_N = q_0(x_0)$ with $q_0 = (e, \langle q, p_\sigma \rangle)$ where $P_M = \{p_a, p_\sigma\}$. By (1) of Definition 35, $\mathsf{rhs}_N(q_0, a) = \mathsf{rhs}_{M,\varphi}(q_0, a) = e$ and so N has the rule $q_0(a) \to e$. To compute $\mathsf{rhs}_N(q_0, \sigma)$ by Definition 35, we observe that for $y, z \in \{a, \sigma\}$, we have

$$\mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_y, p_z) = \mathsf{rhs}_M(q, \sigma, p_y, p_z). \tag{6}$$

Hence $\mathsf{rhs}_N(q_0, \sigma)\Phi = \mathsf{rhs}_{M,\varphi}(q_0, \sigma)\Phi = \perp$ by (1) of Definition 35, and so N may have a rule of the form $q_0(\sigma(x_1, x_2)) \to q'(x_i)$. However, since both x_1 and x_2 occur in the right-hand sides of the rules of M, condition (a) of Lemma 36 is not satisfied for $v = \varepsilon$, for any $i \in \{1, 2\}$. Thus, $\mathsf{rhs}_N(q_0, \sigma) = \mathsf{rhs}_{M,\varphi}(q_0, \sigma)$ does not exist and the algorithm aborts with answer 'no'. □

The next example illustrates the failure of condition (b) of Lemma 36.

Example 48. Let Σ and Δ be as in the previous example, except that σ has rank 1 in Δ (instead of rank 2). We now consider a canonical dtla M that translates every tree $\sigma(s_1, \sigma(s_2, \ldots, \sigma(s_n, a) \cdots))$ into the tree $r_1 r_2 \cdots r_n e$, where $r_i \in \{a, \sigma\}$ is the root symbol of s_i for $i \in [n]$. It is the same as the dtla of the previous example, except that it has the following four rules

$$q(\sigma(x_1:p_a, x_2:p_\sigma)) \to a(q(x_2)), \qquad q(\sigma(x_1:p_\sigma, x_2:p_\sigma)) \to \sigma(q(x_2)),$$
$$q(\sigma(x_1:p_a, x_2:p_a)) \to a(e), \qquad q(\sigma(x_1:p_\sigma, x_2:p_a)) \to \sigma(e).$$

The construction of the dtop N by Algorithm 44 is of course the same as in the previous example, including Equation (6), and again N may have a rule of the form $q_0(\sigma(x_1, x_2)) \to q'(x_i)$. This time only x_2 occurs in the right-hand sides of the rules of M, and so condition (a) of Lemma 36 is satisfied for $v = \varepsilon$ and $i = 2$ (and not for $i = 1$, because x_2 actually occurs). However, for $i = 2$ (and $v = \varepsilon$), condition (b) of Lemma 36 is not satisfied because $\mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_a, p_\sigma)\Omega = a(\langle q, p_\sigma \rangle) \neq \sigma(\langle q, p_\sigma \rangle) = \mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_\sigma, p_\sigma)\Omega$. Thus, the algorithm aborts with answer 'no'.

Note that also $\mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_a, p_a)\Omega = a(e) \neq \sigma(e) = \mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_\sigma, p_a)\Omega$. Thus, the algorithm would also abort if, in the rules, $a(q(x_2))$ would be changed into $\sigma(q(x_2))$ (or if $a(e)$ would be changed into $\sigma(e)$). □

Finally, we give an example in which Algorithm 44 successfully returns a dtop N that is equivalent to M.

Example 49. Consider the la-uniform version of the dtla M of Example 9, as presented in Example 14. As observed after Example 14, it is easy to see that M is canonical. Recall that its translation is such that $M(aa) = aa$, $M(ab) = ab$, $M(ba) = ba$,

$M(bb) = bb$, and $M(\sigma(s_1, s_2)) = \sigma(M(s_1), M(s_2), \#(y, z))$ 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 $P_M = \{p_{aa}, p_{ab}, p_{ba}, p_{bb}\}$ with $\delta(yz) = p_{yz}$ and $\delta(\sigma, p_{wx}, p_{yz}) = p_{wz}$ for all $w, x, y, z \in \{a, b\}$. It has $Q_M = \{q_{yz} \mid y, z \in \{a, b\}\}$, its axioms are $A_M(p_{yz}) = q_{yz}(x_0)$, and its rules are $q_{yz}(yz) \rightarrow yz$ and

$$q_{wz}(\sigma(x_1 : p_{wx}, x_2 : p_{yz})) \rightarrow \sigma(q_{wx}(x_1), q_{yz}(x_2), \#(w, z))$$

for all $w, x, y, z \in \{a, b\}$. In Example 18 we have seen that 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 now run Algorithm 44 on M, constructing a dtop N. Since lubdiff$(M) = 0$, the construction is the same for every difference bound $h(M)$. By Definition 34, $A_N = q_0(x_0)$ with $\varphi(q_0, p_{yz}) = \langle q_{yz}, p_{yz}\rangle$ for $y, z \in \{a, b\}$. So, q_0 is the 4-tuple $(\langle q_{aa}, p_{aa}\rangle, \langle q_{ab}, p_{ab}\rangle, \langle q_{ba}, p_{ba}\rangle, \langle q_{bb}, p_{bb}\rangle)$. From (1) of Definition 35 we conclude that $\mathsf{rhs}_N(q_0, yz)\Phi = \mathsf{rhs}_{M,\varphi}(q_0, yz) = \varphi(q_0, p_{yz})[\langle q_{yz}, p_{yz}\rangle \leftarrow \mathsf{rhs}_M(q_{yz}, yz)] = \mathsf{rhs}_M(q_{yz}, yz) = yz$, and hence N has the rules $q_0(yz) \rightarrow yz$ for all $y, z \in \{a, b\}$. To compute $\mathsf{rhs}_N(q_0, \sigma)$, we first observe that for every $w, x, y, z \in \{a, b\}$,

$$\begin{aligned}
\mathsf{rhs}_{M,\varphi}&(q_0, \sigma, p_{wx}, p_{yz}) \\
&= \varphi(q_0, p_{wz})[\langle q_{wz}, p_{wz}\rangle \leftarrow \mathsf{rhs}_M(q_{wz}, \sigma, p_{wx}, p_{yz})] \\
&= \mathsf{rhs}_M(q_{wz}, \sigma, p_{wx}, p_{yz}) \\
&= \sigma(q_{wx}(x_1), q_{yz}(x_2), \#(w, z)).
\end{aligned} \tag{7}$$

So, by (1) of Definition 35, $\mathsf{rhs}_N(q_0, \sigma)\Phi = \sigma(\bot, \bot, \#(\bot, \bot))$. Thus, N may have a rule of the form

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

We first (try to) determine $\mathsf{rhs}_N(q_0, \sigma)/v = q_1(x_{i_1})$ with $v = (3, 1)$. By Equation (7), $\mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_{wx}, p_{yz})/v = w$ for all look-ahead states. So, (2)(a) of Definition 35 is satisfied for both $i_1 = 1$ and $i_1 = 2$ (and hence does not determine i_1). Clearly, since $\mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_{wx}, p_{yz})\Omega/v$ does not depend on p_{yz}, condition (b) of Lemma 36 is satisfied for $i_1 = 1$ (and not for $i_1 = 2$). So $i_1 = 1$, and $\varphi(q_1, p_{wx}) = w$ for all $w, x \in \{a, b\}$ by (2)(b) of Definition 35. Thus, $q_1 = (a, a, b, b)$. Similarly we obtain for $v = (3, 2)$ that $i_2 = 2$ and $\varphi(q_2, p_{yz}) = z$, and hence $q_2 = (a, b, a, b)$.

Next we (try to) determine $\mathsf{rhs}_N(q_0, \sigma)/v = q_3(x_{i_3})$ with $v = 1$. Since Equation (7) shows that $\mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_{wx}, p_{yz})/v = q_{wx}(x_1)$, (2)(a) of Definition 35 is satisfied for $i_3 = 1$, but not for $i_3 = 2$. Hence $i_3 = 1$. Since $\mathsf{rhs}_{M,\varphi}(q_0, \sigma, p_{wx}, p_{yz})\Omega/v = \langle q_{wx}, p_{wx}\rangle$ does not depend on p_{yz}, condition (b) of Lemma 36 is satisfied for $i_3 = 1$. Hence $\varphi(q_1, p_{wx}) = \langle q_{wx}, p_{wx}\rangle$ by (2)(b) of Definition 35, and so $q_3 = q_0$. Similarly we obtain for $v = 2$ that $i_4 = 2$ and $q_4 = q_0$. Consequently, N has the rule

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

We now (try to) determine the rules for $q_2 = (a, b, a, b)$. 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 $q_2(yz) \rightarrow z$ and it may

have a rule of the form $q_2(\sigma(x_1, x_2)) \to q(x_i)$. Let $v = \varepsilon$. Clearly, (2)(a) of Definition 35 is satisfied for $i \in [2]$ because $z \in \mathcal{T}_\Delta$. Also, condition (b) of Lemma 36 is satisfied for $i = 2$ because $\mathsf{rhs}_{M,\varphi}(q_2, \sigma, p_{wx}, p_{yz})$ does not depend on p_{wx}. So $i = 2$, and $\varphi(q, p_{yz}) = z$ by (2)(b) of Definition 35, 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 $q_1(\sigma(x_1, x_2)) \to q_1(x_1)$.

Thus, since N is now complete, Algorithm 44 halts and outputs the dtop N that has axiom $A_N = q_0(x_0)$ and the rules

$$
\begin{array}{ll}
q_0(yz) \to yz, & q_0(\sigma(x_1, x_2)) \to \sigma(q_0(x_1), q_0(x_2), \#(q_1(x_1), q_2(x_2))), \\
q_1(yz) \to y, & q_1(\sigma(x_1, x_2)) \to q_1(x_1), \\
q_2(yz) \to z, & q_2(\sigma(x_1, x_2)) \to q_2(x_2),
\end{array}
$$

for $y, z \in \{a, b\}$. It should be clear that N is indeed equivalent to M (cf. the end of Example 9).

We finally note that the original dtla M of Example 9 is ultralinear and b-erasing (even linear and nonerasing). The difference bound in the proof of Theorem 85 for this M is $1 + 4 \cdot 2 \cdot 3^2 \cdot 4^2 = 1153$. By Lemma 13, the la-uniform version of M has the same difference bound. Obviously, this is a rather large bound, in view of the fact that $\mathsf{lubdiff}(M) = 0$. □

It is left to the reader to show that for the dtla M of Example 8, Algorithm 44 computes the dtop N of Example 8, which was used as a running example in Section 6.1.

6.3 The First Main Result

The next theorem is our first main result, for arbitrary total dtlas.

Theorem 50. *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 Lemma 13, we may additionally assume that M is la-uniform. By Theorem 15, an equivalent canonical dtpla $\mathsf{can}(M)$ can be constructed such that, as shown after the theorem, $h(M) + \mathsf{sumfix}(M)$ is a (computable) difference bound for $\mathsf{can}(M)$. We now apply Lemma 41 to $\mathsf{can}(M)$ and its difference bound $h(\mathsf{can}(M)) = h(M) + \mathsf{sumfix}(M)$. □

As an algorithm for this problem, we can use the constructions in the proofs of Lemma 13 and Theorem 15, followed by Algorithm 44. Thus, in case of success, the constructed dtop N is the unique canonical dtop equivalent to M, which we previously denoted by $\mathsf{td}(M)$.

The algorithm is useful for the designer of a dtla, cf. Examples 7–10 in Section 4. If you have designed a dtla M to satisfy a specification of its translation $[\![M]\!]$, then you usually also know how to specify the output $M(C[p])$, for every context C and every look-ahead state p. From this it is probably straightforward for you to obtain a

specification of $\text{diff}(M)$. If $\text{diff}(M)$ is infinite (which you probably also can see), then M is not equivalent to a dtop by Corollary 28. If it is finite, then you can determine $\text{lubdiff}(M)$ (or an upper bound for it), and hence you have determined a difference bound for M. So now you can use the algorithm of Theorem 50 to find out whether M is equivalent to a dtop, and if so, to construct such a dtop. On the other hand, if you are *not* able to determine a difference bound for M, then, as discussed after Theorem 45, you can still use the algorithm of Theorem 50, without the tests on height in Algorithm 44. If M is equivalent to a dtop, then the algorithm will construct such a dtop; otherwise, the algorithm outputs 'no' or may not halt (as was shown in Example 46).

From Theorem 50 we immediately obtain the following result.

Corollary 51. *Let $\mathcal{U}$ be a class of total dtlas with the following property.*

(H) There is a computable mapping $h : \mathcal{U} \to \mathbb{N}$ such that, for every $M \in \mathcal{U}$,
    $h(M)$ is a difference bound for M.

Then it is decidable for a given dtla $M \in \mathcal{U}$ whether there exists a dtop N such that $[\![M]\!] = [\![N]\!]$, and if so, such a dtop N can be constructed.

Proof. For a dtla M in $\mathcal{U}$, compute the difference bound $h(M)$ and run the algorithm of Theorem 50. $\qquad\square$

Let $\mathcal{U}$ be the class of total ultralinear b-erasing dtlas. Our goal in Sections 7–9 is to prove that $\mathcal{U}$ has Property (H), i.e., to compute a difference bound for the dtlas in the class $\mathcal{U}$.

7 Links and Origins

In this section we define two basic concepts for dtlas, and discuss some of their properties. Thus, this section can be viewed as a sequel to Section 3.

Convention. In this section and the next two sections we assume that M is a dtla that is **initialized** with initial states $q_{0,p}$ for $p \in P$ and **la-uniform** with la-map $\rho : Q \to P$, cf. Lemmas 11 and 13. Since M is la-uniform, all its initial states $q_{0,p}$ are distinct.

If $M(s) = t$, then every node v of the output tree $t \in \mathcal{T}_\Delta$ is produced (together with its label) at a certain node u of the input tree $s \in \mathcal{T}_\Sigma$ by the application of a rule $q(a(x_1{:}p_1, \ldots, x_k{:}p_k)) \to \zeta$ at u. The node v is an instantiation of a node z of ζ with label in Δ. The triple (q, u, z) will be called the "origin" of v, cf. [28].

In the special case that $z = \varepsilon$, there exist pairs (q', u') for which there is a reachable sentential form ξ for s such that $\xi/v = q'(u')$. In such a case v is already a node of ξ, but does not yet have a label in Δ. In fact, (q, u) is such a pair, but there can be more due to the presence of erasing rules: if (q', u') is such a pair and $\text{rhs}(q', s, u') = q''(x_i)$, then $(q'', u'i)$ is also such a pair. A pair (q', u') will be called a "link" to v (cf. [21]). For convenience it will be denoted as a triple $(q', u', \#)$, and the origin (q, u, z) of v will also be called a link to v.

We now formally define links. For $s \in \mathcal{T}_\Sigma$ and all $v \in \mathbb{N}_+^*$, the sets $\text{link}_s(v)$ of triples (q, u, z) with $q \in Q$, $u \in V(s)$, and $z \in \mathbb{N}_+^* \cup \{\#\}$, are defined recursively to be the smallest sets such that:

(1) If $\delta(s) = p$, then $(q_{0,p}, \varepsilon, \#) \in \mathsf{link}_s(\varepsilon)$.
(2) If $(q, u, \#) \in \mathsf{link}_s(v)$ and $\mathsf{rhs}(q, s, u) = \zeta$, then
    (a) $(q, u, z) \in \mathsf{link}_s(vz)$ for every $z \in V_\Delta(\zeta)$, and
    (b) $(\bar{q}, ui, \#) \in \mathsf{link}_s(vz)$ for every $z \in V(\zeta)$ with $\zeta/z = \bar{q}(x_i)$.

A triple (q, u, z) in $\mathsf{link}_s(v)$ is called a *link* from u to v; if $z \in \mathbb{N}_+^*$, then it is also called an *origin* of v. The above intuition about links and origins is proved in the next two lemmas. It will follow that $\mathsf{link}_s(v)$ contains exactly one origin.

Lemma 52. *Let $s \in \mathcal{T}_\Sigma$, $q \in Q$, $u \in V(s)$, and $z, v \in \mathbb{N}_+^*$. Then*

(1) $(q, u, \#) \in \mathsf{link}_s(v)$ if and only if there is a sentential form ξ of M such that $q_{0,\delta(s)}(\varepsilon) \Rightarrow_s^* \xi$ and $\xi/v = q(u)$,
(2) $(q, u, z) \in \mathsf{link}_s(v)$ if and only if there are $\hat{v} \in \mathbb{N}_+^*$ and a sentential form ξ of M such that $v = \hat{v}z$, $q_{0,\delta(s)}(\varepsilon) \Rightarrow_s^* \xi$, $\xi/\hat{v} = q(u)$, and $z \in V_\Delta(\mathsf{rhs}(q, s, u))$,
(3) $\mathsf{link}_s(v) \neq \emptyset$ if and only if $v \in V(M(s))$, and
(4) if $(q, u, \#) \in \mathsf{link}_s(v)$ or $(q, u, z) \in \mathsf{link}_s(v)$, then $\rho(q) = \delta(s/u)$.

Proof. Since the definition of link_s closely follows the semantics of M, (1) and (2) are easy to prove by recursion induction on the definition of link_s in one direction, and by induction on the length of the computation $q_{0,\delta(s)}(\varepsilon) \Rightarrow_s^* \xi$ in the other direction.

Clearly, the right-hand side of (1) implies that $v \in V(M(s))$, by Lemma 2(2) applied to $\xi \Rightarrow_s^* M(s)$, and the right-hand side of (2) also implies that $v \in V(M(s))$ because there is a computation step $\xi \Rightarrow_s \xi' = \xi[\hat{v} \leftarrow \mathsf{rhs}(q, s, u)[x_i \leftarrow ui \mid i \in \mathbb{N}_+]]$ with $v \in V(\xi')$. Thus, if $\mathsf{link}_s(v) \neq \emptyset$ then $v \in V(M(s))$. Now let $v \in V(M(s))$ and consider, in a computation of M that translates s into $M(s)$, the first sentential form ξ such that $v \in V_{\Delta \cup Q}(\xi)$. If $v \in V_Q(\xi)$, then $\mathsf{link}_s(v)$ contains a triple $(q, u, \#)$ by (1). If $v \in V_\Delta(\xi)$, then v is produced in the computation step $\xi' \Rightarrow_s \xi$ where ξ' is the previous sentential form, and then $\mathsf{link}_s(v)$ contains a triple (q, u, z) by (2) applied to ξ'. Thus, $\mathsf{link}_s(v) \neq \emptyset$.

(4) follows from (1) and (2) by Lemma 12(4). $\qquad\qquad\square$

Lemma 53. *For every $s \in \mathcal{T}_\Sigma$ and $v \in V(M(s))$, either $\mathsf{link}_s(v) = \{(q, u, z)\}$ with $z \in \mathbb{N}_+^*$ and $z \neq \varepsilon$, or $\mathsf{link}_s(v) = \{(q_1, u_1, \#), \ldots, (q_n, u_n, \#), (q_n, u_n, \varepsilon)\}$ with $n \geq 1$ and for every $j \in [n-1]$ there exists $i_j \in \mathbb{N}_+$ such that $\mathsf{rhs}(q_j, s, u_j) = q_{j+1}(x_{i_j})$ and $u_{j+1} = u_j i_j$.*

Proof. Let us say that a mapping λ from $\mathbb{N}_+^*$ to the finite subsets of $Q \times V(s) \times (\mathbb{N}^* \cup \{\#\})$ is an *approximation* of link_s if it is obtained by a finite number of applications of requirements (1) and (2) of the definition of link_s, starting with $\lambda(v) = \emptyset$ for all $v \in \mathbb{N}_+^*$ (and so $\lambda(v) \subseteq \mathsf{link}_s(v)$ for all v). It is straightforward to show by induction on the number of such applications that for every $v \in \mathbb{N}_+^*$, either $\lambda(v) = \emptyset$, or $\lambda(v) = \{(q, u, z)\}$ with $z \in \mathbb{N}_+^* - \{\varepsilon\}$, or $\lambda(v) = \{(q_1, u_1, \#), \ldots, (q_n, u_n, \#), (q_n, u_n, \varepsilon)\}$ with the condition stated in the lemma, or $\lambda(v) = \{(q_1, u_1, \#), \ldots, (q_n, u_n, \#)\}$ with that same condition. In the last case, Lemma 52(4) implies that $\mathsf{rhs}(q_n, s, u_n)$ is defined, and hence $\lambda(v)$ is properly included in $\mathsf{link}_s(v)$ by the definition of link_s. The first case does not occur when $v \in V(M(s))$, by Lemma 52(3). Since link_s is itself an approximation of link_s, this proves the lemma. $\qquad\square$

Lemma 53 shows that every node v of $M(s)$ has exactly one origin. Thus, for $s \in \mathcal{T}_\Sigma$ and $v \in V(M(s))$, we define $\mathrm{or}_s(v) \in Q \times V(s) \times \mathbb{N}_+^*$, called *the* origin of v, by $\mathrm{or}_s(v) = (q, u, z)$ if $(q, u, z) \in \mathrm{link}_s(v)$. We denote the node u of s also by $\mathrm{orn}_s(v)$ and call it the *origin node* of v.

In the remainder of this section we state some elementary properties of links and origins.

Lemma 54. *Let $s \in \mathcal{T}_\Sigma$, $v \in V(M(s))$, $q \in Q$, $u \in V(s)$, and $z \in \mathbb{N}_+^*$.*

(1) If $(q, u, \#) \in \mathrm{link}_s(v)$, then $M(s)/v = q_M(s/u)$.
(2) If $\mathrm{or}_s(v) = (q, u, z)$ and $\zeta = \mathrm{rhs}(q, s, u)$, then $z \in V_\Delta(\zeta)$ and

$$M(s)/v = q_M(s/u)/z = \zeta/z[\bar{q}(x_i) \leftarrow \bar{q}_M(s/ui) \mid \bar{q} \in Q,\ i \in \mathbb{N}_+].$$

Proof. If $(q, u, \#) \in \mathrm{link}_s(v)$ then, by Lemma 52(1), there is a reachable sentential form ξ for s such that $\xi/v = q(u)$. Since $\xi \Rightarrow_s^* M(s)$, we obtain that $\xi/v \Rightarrow_s^* M(s)/v$ and so $M(s)/v = q_M(s/u)$ by Lemma 2(2).

If $\mathrm{or}_s(v) = (q, u, z)$ then, by Lemma 52(2), there are $\hat{v} \in \mathbb{N}^*$ and a reachable sentential form ξ for s such that $v = \hat{v}z$, $\xi/\hat{v} = q(u)$, and $z \in V_\Delta(\zeta)$. Hence $(q, u, \#) \in \mathrm{link}_s(\hat{v})$ by Lemma 52(1), and so $M(s)/\hat{v} = q_M(s/u)$ by the previous paragraph. Thus $M(s)/v = q_M(s/u)/z$. Since $q_M(s/u) = \zeta[\bar{q}(x_i) \leftarrow \bar{q}_M(s/ui) \mid \bar{q} \in Q,\ i \in \mathbb{N}_+]$ by Lemma 2(4), this shows the result. $\square$

The following four lemmas state relationships between the link sets and the ancestor relation.

Recall that (since M is initialized) $\mathrm{maxrhs}(M)$ is the maximal height of the right-hand sides of rules of M.

Lemma 55. *Let $s \in \mathcal{T}_\Sigma$ and let $v_i \in V(M(s))$, $q_i \in Q$, $u_i \in V(s)$, and $z_i \in \mathbb{N}_+^* \cup \{\#\}$ for $i \in [2]$. If $(q_1, u_1, z_1) \in \mathrm{link}_s(v_1)$ and $(q_2, u_2, z_2) \in \mathrm{link}_s(v_2)$, and v_1 is a proper ancestor of v_2, then*

(1) u_1 is an ancestor of u_2, and
(2) if $u_1 = u_2$ and $z_1, z_2 \in \mathbb{N}_+^*$, then $|v_2| - |v_1| \leq \mathrm{maxrhs}(M)$.

Proof. It is immediate from the definition of link_s that if v_1 is a proper ancestor of v_2, then u_1 is an ancestor of u_2. Moreover, if $u_1 = u_2 = u$ and $z_1, z_2 \in \mathbb{N}_+^*$, then $q_1 = q_2 = q$ and $z_1, z_2 \in V_\Delta(\mathrm{rhs}(q, s, u))$, and there exists $v \in \mathbb{N}_+^*$ such that $v_1 = vz_1$ and $v_2 = vz_2$; hence $|v_2| - |v_1| = |z_2| - |z_1| \leq \mathrm{ht}(\mathrm{rhs}(q, s, u)) \leq \mathrm{maxrhs}(M)$. $\square$

The next lemma is an immediate corollary of Lemma 55.

Lemma 56. *Let $s \in \mathcal{T}_\Sigma$ and $v, \hat{v} \in V(M(s))$.*

(1) If $\hat{v}$ is an ancestor of v, then $\mathrm{orn}_s(\hat{v})$ is an ancestor of $\mathrm{orn}_s(v)$.
(2) If $\hat{v}$ is an ancestor of v and $\mathrm{orn}_s(\hat{v}) = \mathrm{orn}_s(v)$, then $|v| - |\hat{v}| \leq \mathrm{maxrhs}(M)$.

Lemma 57. *Let $s \in \mathcal{T}_\Sigma$ and $v \in V(M(s))$. If $\mathrm{or}_s(v) = (q, u, z)$, then there exists $\hat{v} \in \mathbb{N}_+^*$ such that $v = \hat{v}z$ and $(q, u, \#) \in \mathrm{link}_s(\hat{v})$.*

Proof. Immediate by (2) and (1) of Lemma 52. □

Lemma 58. *Let $s \in \mathcal{T}_\Sigma$ and $v \in V(M(s))$. If $(q, u, z) \in \mathsf{link}_s(v)$ and $\hat{u}$ is an ancestor of u in s, then there exist an ancestor $\hat{v}$ of v and a state $q' \in Q$ such that $(q', \hat{u}, \#) \in \mathsf{link}_s(\hat{v})$.*

Proof. Straightforward by recursion induction on the definition of link_s, as follows. For $(q_{0,\delta(s)}, \varepsilon, \#) \in \mathsf{link}_s(\varepsilon)$ we have $\hat{u} = \varepsilon$ and we take $\hat{v} = \varepsilon$ and $q' = q_{0,\delta(s)}$. Assume now that the statement holds for $(q, u, \#) \in \mathsf{link}_s(v)$. For $(q, u, z) \in \mathsf{link}_s(vz)$ and an ancestor $\hat{u}$ of u we obtain by induction that $(q', \hat{u}, \#) \in \mathsf{link}_s(\hat{v})$ for an ancestor $\hat{v}$ of v, which is also an ancestor of vz. For $(\bar{q}, ui, \#) \in \mathsf{link}_s(vz)$ and an ancestor $\hat{u}$ of u (and hence of ui) the previous argument also holds; for the ancestor $\hat{u} = ui$ of ui we take $\hat{v} = vz$ and $q' = \bar{q}$. □

Intuitively, a fact such as $(q, u, \#) \in \mathsf{link}_s(v)$ does not depend on the whole of s, but only on the proper ancestors of u and their children. This is proved in the next lemma.
Let $s, s' \in \mathcal{T}_\Sigma$ and $u \in V(s) \cap V(s')$. We will say that u is *similar in s and s'* if

(1) $\mathsf{lab}(s, u) = \mathsf{lab}(s', u)$, and
(2) $\delta(s/ui) = \delta(s'/ui)$ for every child ui of u.

This implies that $\mathsf{rhs}(q, s, u) = \mathsf{rhs}(q, s', u)$ for every $q \in Q$.

Lemma 59. *Let $s, s' \in \mathcal{T}_\Sigma$ be such that $\delta(s) = \delta(s')$, and let $u \in V(s) \cap V(s')$ be such that every proper ancestor of u is similar in s and s'. Let $q \in Q$ and $v, z \in \mathbb{N}_+^*$.*

(1) If $(q, u, \#) \in \mathsf{link}_s(v)$ then $(q, u, \#) \in \mathsf{link}_{s'}(v)$.
(2) Let, moreover, u be similar in s and s'.
   If $(q, u, z) \in \mathsf{link}_s(v)$ then $(q, u, z) \in \mathsf{link}_{s'}(v)$.

Proof. We prove (1) by induction on $|u|$. By the definition of link_s, if $(q, \varepsilon, \#) \in \mathsf{link}_s(v)$, then $q = q_{0,\delta(s)}$ and $v = \varepsilon$, and so $(q, \varepsilon, \#) \in \mathsf{link}_{s'}(v)$ by the definition of $\mathsf{link}_{s'}$. For the induction step we consider, by the definition of link_s, that $(\bar{q}, ui, \#) \in \mathsf{link}_s(vz)$ where $(q, u, \#) \in \mathsf{link}_s(v)$, $\mathsf{rhs}(q, s, u) = \zeta$, and $\zeta/z = \bar{q}(x_i)$. By induction, $(q, u, \#) \in \mathsf{link}_{s'}(v)$. Since u is similar in s and s', $\mathsf{rhs}(q, s', u) = \zeta$. Hence $(\bar{q}, ui, \#) \in \mathsf{link}_{s'}(vz)$ by the definition of $\mathsf{link}_{s'}$.

To prove (2) we consider, by the definition of link_s, that $(q, u, z) \in \mathsf{link}_s(vz)$ where $(q, u, \#) \in \mathsf{link}_s(v)$, $\mathsf{rhs}(q, s, u) = \zeta$, and $z \in V_\Delta(\zeta)$. By (1), $(q, u, \#) \in \mathsf{link}_{s'}(v)$, and since u is similar in s and s' by assumption, $\mathsf{rhs}(q, s', u) = \zeta$. Hence $(q, u, z) \in \mathsf{link}_{s'}(vz)$ by the definition of $\mathsf{link}_{s'}$. □

Note that in this lemma, by reasons of symmetry, the implications are actually equivalences.

In the next section we wish to change (in fact, pump) parts of the input tree s in such a way that a given node v of $M(s)$ is preserved, together with the labels of all its ancestors (cf. Lemma 6). Intuitively, this can be done as long as we do not change the following in s: the origin node u of v, the labels of all ancestors of u, and for each ancestor of u the look-ahead states at its children. This is proved in the next lemma.

Lemma 60. *Let $s \in \mathcal{T}_\Sigma$ and $v \in V(M(s))$, and let $\mathsf{orn}_s(v) = u \in V(s)$. Moreover, let $s' \in \mathcal{T}_\Sigma$ be such that $u \in V(s')$ and every ancestor of u (including u itself) is similar in s and s'. Then $v \in V(M(s'))$ and $\mathsf{lab}(M(s'), \hat{v}) = \mathsf{lab}(M(s), \hat{v})$ for every ancestor $\hat{v}$ of v (including v itself).*

Proof. Since, by Lemma 56(1), $\mathsf{orn}_s(\hat{v})$ is an ancestor of u, it suffices to prove the lemma for $\hat{v} = v$. Let $\mathsf{or}_s(v) = (q, u, z)$. Since the root ε is similar in s and s', $\delta(s) = \delta(s')$. It now follows from Lemma 59(2) that $v \in V(M(s'))$ and $\mathsf{or}_{s'}(v) = (q, u, z) = \mathsf{or}_s(v)$. Using this, and the fact that $\mathsf{rhs}(q, s', u) = \mathsf{rhs}(q, s, u)$ (because u is similar in s and s'), we obtain from Lemma 54(2) that $\mathsf{lab}(M(s'), v) = \mathsf{lab}(M(s), v)$. $\square$

8   Auxiliary Bounds

If M is an initialized la-uniform dtla, then so is M°, cf. Lemma 12(3). In this section and the next, the lemmas of the previous section will be applied to M° instead of M.

   We recall that $\mathsf{maxrhs}(M)$ is the maximal height of the right-hand sides of rules of M.

   In view of Theorem 50 and Corollary 51, we wish to compute a difference bound for M, i.e., an upper bound for $\mathsf{lubdiff}(M)$ when $\mathsf{diff}(M)$ is finite. Thus, we are looking for an upper bound on the height of all difference trees of M, i.e., all trees $M(C[p])/v$ where C is a Σ-context, $p \in P$, and v is a difference node of $M(C[p])$ and $M(C[p'])$ where $p' \in P$. Let u and u' be the respective origin nodes of v, i.e., $u = \mathsf{orn}_{C[p]}(v)$ and $u' = \mathsf{orn}_{C[p']}(v)$.

   We first compute an upper bound for the case where u is *not* a proper ancestor of u'. The idea is that if the height of $M(C[p])/v$ is larger than that upper bound, then we can pump the subtree $C[p]/uj$ at one of u's children uj (without changing the look-ahead state at uj in both $C[p]$ and $C[p']$), thus turning C into $\bar{C}$, in such a way that $M(\bar{C}[p])/v$ becomes arbitrarily large. Since the pumping does not change the labels of the ancestors of u and u', nor the look-ahead states at the children of those ancestors, v is still a difference node of $M(\bar{C}[p])$ and $M(\bar{C}[p'])$ by Lemmas 6 and 60.

   To express an upper bound for the height of $M(C[p])/v$, we use an auxiliary bound defined as follows.

Definition 61. *A number $h_\mathrm{o}(M) \in \mathbb{N}$ is an* output bound *for M if it has the following two properties, for every $q \in Q$ and $p, p', p_1, p'_1 \in P$:*

(1) if the set $\{q_M(s) \mid s \in \mathcal{T}_\Sigma, \delta(s) = p_1\}$ is finite, then $\mathsf{ht}(t) \leq h_\mathrm{o}(M)$ for every tree t in this set, and
(2) if the set $\{q_M(C[p]) \mid C \in \mathcal{C}_\Sigma, \delta(C[p]) = p_1, \delta(C[p']) = p'_1\}$ is finite, then $\mathsf{ht}(t) \leq h_\mathrm{o}(M)$ for every tree t in this set.

Note that since M is la-uniform, it suffices to consider the case where $p_1 = \rho(q)$. Condition (1) of this definition is needed for the case where the subtree $C[p]/uj$ discussed above is in $\mathcal{T}_\Sigma$ (i.e., does not contain p), whereas Condition (2) is needed when $C[p]/uj$ is of the form $D[p]$ for a context $D \in \mathcal{C}_\Sigma$ (i.e., contains p). The restrictions on

δ in the sets of Conditions (1) and (2) are needed to ensure that the pumping of $C[p]/uj$ preserves the look-ahead state of uj in $C[p]$ and $C[p']$.

The minimal output bound $h_o(M)$ can be computed from M, because every set mentioned in Definition 61 is the image of a regular tree language by a dtla translation (as will be shown in the proof of Theorem 67), and because it is decidable whether or not such an image is finite, and if so, the elements of that image can be computed (see [5, Theorem 4.5] and note that every dtla translation can be realized by a macro tree transducer). Also, for completeness sake, we will prove by a pumping argument that $\mathsf{maxrhs}(M) \cdot |Q| \cdot (|P| + 2)$ is an output bound for M, in Lemma 66 and Theorem 67.

We now show that the upper bound discussed above is $\mathsf{maxrhs}(M) + h_o(M)$. For later use (in the proof of Lemma 64) we prove a slightly more general result; the case discussed above is obtained by taking $\bar{v} = v$.

Lemma 62. *Let $h_o(M)$ be an output bound for M. Let $C \in \mathcal{C}_\Sigma$ and $p, p' \in P$, let v be a difference node of $M(C[p])$ and $M(C[p'])$, and let $\bar{v}$ be a descendant of v in $M(C[p])$ such that $\mathsf{orn}_{C[p]}(\bar{v})$ is not a proper ancestor of $\mathsf{orn}_{C[p']}(v)$. If $\mathsf{diff}(M)$ is finite, then*

$$\mathsf{ht}(M(C[p])/\bar{v}) \leq \mathsf{maxrhs}(M) + h_o(M).$$

Proof. Assume that $\mathsf{ht}(M(C[p])/\bar{v}) > \mathsf{maxrhs}(M) + h_o(M)$. Let $u = \mathsf{orn}_{C[p]}(v)$ and $u' = \mathsf{orn}_{C[p']}(v)$, and let $(q, \bar{u}, z) = \mathsf{or}_{C[p]}(\bar{v})$. By Lemma 56(1), u is an ancestor of $\bar{u}$. By Lemma 54(2), $M(C[p])/\bar{v} = \zeta/z[\bar{q}(x_i) \leftarrow \bar{q}_M(C[p]/\bar{u}i) \mid \bar{q} \in Q, i \in \mathbb{N}_+]$ where $\zeta = \mathsf{rhs}(q, C[p], \bar{u})$. Since $\mathsf{ht}(M(C[p])/\bar{v}) > \mathsf{maxrhs}(M) + h_o(M)$, there exist $y \in \mathbb{N}_+^*$, $q' \in Q$, and $j \in \mathbb{N}$ such that $\zeta/zy = q'(x_j)$ and $\mathsf{ht}(q'_M(C[p]/\bar{u}j)) > h_o(M)$. Note that $M(C[p])/\bar{v}y = q'_M(C[p]/\bar{u}j)$. Note also that, by the definition of link, $(q', \bar{u}j, \#) \in \mathsf{link}_{C[p]}(\bar{v}y)$. The situation is shown in Figure 1 (but note that it is also possible that u is an ancestor of u', as in Figure 2).
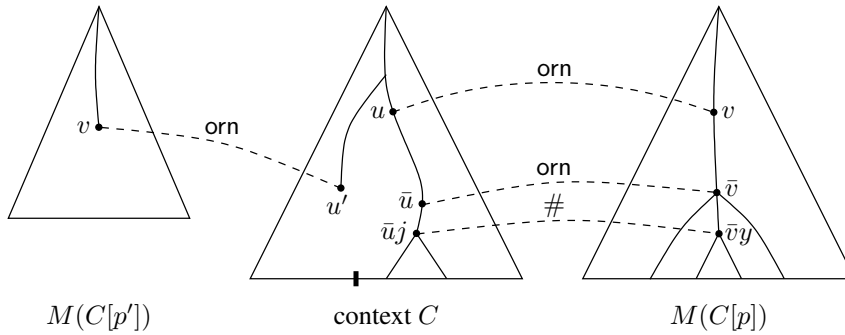
Fig. 1. $\bar{u} = \mathsf{orn}_{C[p]}(\bar{v})$ is not a proper ancestor of $u' = \mathsf{orn}_{C[p']}(v)$

There are now two cases: p does or does not occur in $C[p]/\bar{u}j$. Let $p_j = \delta(C[p]/\bar{u}j)$.

Case 1: p does not occur in $C[p]/\bar{u}j$, i.e., $C[p]/\bar{u}j = C/\bar{u}j \in \mathcal{T}_\Sigma$. In Figure 1, the vertical thick line represents the node of C with label $\bot$; so, it shows Case 1. By

Definition 61(1) of $h_\mathrm{o}(M)$, the set $\{q'_M(s) \mid s \in \mathcal{T}_\Sigma, \delta(s) = p_j\}$ is infinite. Hence there exist trees $s_n \in \mathcal{T}_\Sigma$ such that $\delta(s_n) = p_j$ and $\mathsf{ht}(q'_M(s_n)) > n$ for every $n \in \mathbb{N}$. Let $C_n = C[\bar{u}j \leftarrow s_n]$, the context C in which the subtree at $\bar{u}j$ is replaced by s_n. Since $\delta(s_n) = \delta(C[p]/\bar{u}j) = \delta(C[p']/\bar{u}j)$ and $\bar{u}$ is not a proper ancestor of u or u', it follows from Lemma 60 that $\mathsf{lab}(M(C_n[p]), \hat{v}) = \mathsf{lab}(M(C[p]), \hat{v})$ for every ancestor $\hat{v}$ of v, and similarly for p'. Hence, by Lemma 6, v is a difference node of $M(C_n[p])$ and $M(C_n[p'])$, and so $M(C_n[p])/v$ is in $\mathrm{diff}(M)$. Also, since $(q', \bar{u}j, \#) \in \mathsf{link}_{C[p]}(\bar{v}y)$, we obtain from Lemma 59(1) that $(q', \bar{u}j, \#) \in \mathsf{link}_{C_n[p]}(\bar{v}y)$. Consequently, by Lemma 54(1), $M(C_n[p])/\bar{v}y = q'_M(C_n[p]/\bar{u}j) = q'_M(s_n)$, which implies that $\mathsf{ht}(M(C_n[p])/v) \geq \mathsf{ht}(M(C_n[p])/\bar{v}y) > n$. Hence $\mathrm{diff}(M)$ is infinite.

Case 2: p occurs in $C[p]/\bar{u}j$, i.e., $C[p]/\bar{u}j = D[p]$ where $D = C/\bar{u}j \in \mathcal{C}_\Sigma$. Let $p'_j = \delta(C[p']/\bar{u}j)$. By Definition 61(2) of $h_\mathrm{o}(M)$, the set $\{q'_M(D[p]) \mid D \in \mathcal{C}_\Sigma, \delta(D[p]) = p_j, \delta(D[p']) = p'_j\}$ is infinite. Hence there exist contexts D_n such that $\delta(D_n[p]) = p_j$, $\delta(D_n[p']) = p'_j$, and $\mathsf{ht}(q'_M(D_n[p])) > n$. Let $C_n = C[\bar{u}j \leftarrow D_n]$. It can now be shown in the same way as in Case 1 that $M(C_n[p])/v$ is in $\mathrm{diff}(M)$ and $\mathsf{ht}(M(C_n[p])/v) > n$, and hence $\mathrm{diff}(M)$ is infinite. Note that the condition $\delta(D_n[p']) = p'_j$ is needed to ensure that Lemma 60 is applicable to u'. $\qquad\square$

We now consider the case where u *is* a proper ancestor of u', and again wish to obtain an upper bound for the height of $M(C[p])/v$. It follows from Lemma 62 that $\mathsf{ht}(M(C[p])/\bar{v}) \leq \mathsf{maxrhs}(M) + h_\mathrm{o}(M)$ for every descendant $\bar{v}$ of v of which the origin node $\bar{u}$ is not a proper ancestor of u'. But what about a descendant vw of v of which the origin node x *is* a proper ancestor of u'? (see Figure 2 with $w = w_1$). Then, by the previous observation, it suffices to obtain an upper bound on $|w|$. Now we observe that, roughly speaking, when M arrives at node x of $C[p]$ it generates node vw of $M(C[p])$, which is a descendant of v, but when M arrives at node x of $C[p']$ it generates a node $\hat{v}$ of $M(C[p'])$ that is an ancestor of v. Hence, it suffices to have an upper bound on $|vw| - |\hat{v}|$, which measures how much the translation of $M(C[p])$ is ahead of the translation of $M(C[p'])$ when M arrives at x. Note that by Lemma 6, $\mathsf{lab}(M(C[p]), \tilde{v}) = \mathsf{lab}(M(C[p']), \tilde{v})$ for every proper ancestor $\tilde{v}$ of $\hat{v}$.

Thus, to express an upper bound for the height of $M(C[p])/v$, we use an auxiliary bound defined as follows.

Definition 63. *A number $h_\mathrm{a}(M) \in \mathbb{N}$ is an* ancestral bound *for M if $\mathrm{diff}(M)$ is infinite or the following holds for every $C \in \mathcal{C}_\Sigma$, $p, p' \in P$, $x \in V(C)$, $y \in V(M(C[p]))$, $y' \in V(M(C[p']))$, and $q, q' \in Q$:*

if

(1) $(q, x, \#) \in \mathsf{link}_{C[p]}(y)$ and $(q', x, \#) \in \mathsf{link}_{C[p']}(y')$,
(2) y' is a prefix of y, hence an ancestor of y in $M(C[p])$, and
(3) $\mathsf{lab}(M(C[p]), \hat{y}') = \mathsf{lab}(M(C[p']), \hat{y}')$ for every proper ancestor $\hat{y}'$ of y',

*then $|y| - |y'| \leq h_\mathrm{a}(M)$.*

We will show in the next section (Lemma 77) that every dtla M has an ancestral bound $h_\mathrm{a}(M)$. Unfortunately, we do not know whether an ancestral bound can be computed from M. We will also show in the next section (Theorem 84) that it can be computed in the restricted case where M is ultralinear and b-erasing.

48

Lemma 64. *Let $h_o(M)$ be an output bound and $h_a(M)$ an ancestral bound for M. Let $C \in \mathcal{C}_\Sigma$ and $p, p' \in P$, and let v be a difference node of $M(C[p])$ and $M(C[p'])$ such that $\mathrm{orn}_{C[p]}(v)$ is a proper ancestor of $\mathrm{orn}_{C[p']}(v)$. If $\mathrm{diff}(M)$ is finite, then*

$$\mathrm{ht}(M(C[p])/v) \leq 2 \cdot \mathrm{maxrhs}(M) + h_o(M) + h_a(M) + 1.$$

Proof. Let $\mathrm{orn}_{C[p]}(v) = u$ and $\mathrm{orn}_{C[p']}(v) = u'$. Consider an arbitrary leaf w of $M(C[p])/v$. We have to show that $|w| \leq 2 \cdot \mathrm{maxrhs}(M) + h_o(M) + h_a(M) + 1$. The proof is illustrated in Figure 2.
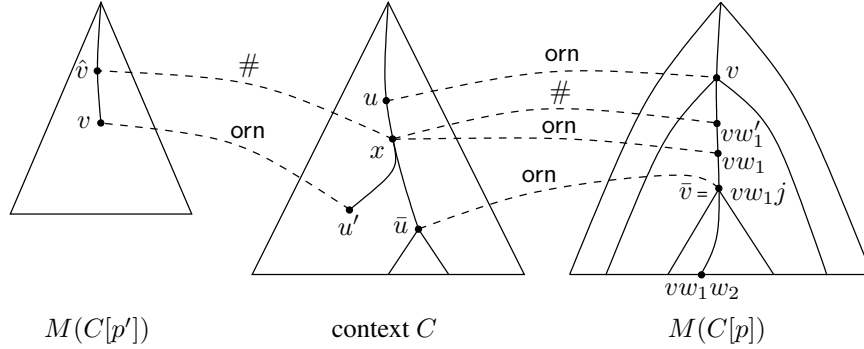
Fig. 2. $u = \mathrm{orn}_{C[p]}(v)$ is a proper ancestor of $u' = \mathrm{orn}_{C[p']}(v)$

Let $w = w_1 w_2$ where w_1 is the longest ancestor of w such that $\mathrm{orn}_{C[p]}(vw_1)$ is a proper ancestor of u'. If $w_2 = jw_2'$ with $j \in \mathbb{N}_+$, then $\mathrm{orn}_{C[p]}(vw_1 j)$ is not a proper ancestor of u', and so $|w_2| = |w_2'| + 1 \leq \mathrm{maxrhs}(M) + h_o(M) + 1$ by Lemma 62 (with $\bar{v} = vw_1 j$). It now suffices to prove that $|w_1| \leq \mathrm{maxrhs}(M) + h_a(M)$.

Let $\mathrm{or}_{C[p]}(vw_1) = (q, x, z)$, and note that x is a descendant of u by Lemma 56(1) (and a proper ancestor of u' by definition of w_1). If $x = u$, then $|w_1| \leq \mathrm{maxrhs}(M)$ by Lemma 56(2). Otherwise, by Lemma 57 (applied to vw_1) and Lemma 55(1), $w_1 = w_1' z$ such that $(q, x, \#) \in \mathrm{link}_{C[p]}(vw_1')$. Since $|z| \leq \mathrm{maxrhs}(M)$, it now remains to show that $|w_1'| \leq h_a(M)$.

By Lemma 58, since x is an ancestor of u', there is an ancestor $\hat{v}$ of v such that $(q', x, \#) \in \mathrm{link}_{C[p']}(\hat{v})$ for some $q' \in Q$. We now apply Definition 63 of $h_a(M)$ to the nodes $y = vw_1'$ of $M(C[p])$ and $y' = \hat{v}$ of $M(C[p'])$, and obtain that $|w_1'| \leq |vw_1'| - |\hat{v}| \leq h_a(M)$. Note that condition (3) in Definition 63 is satisfied by Lemma 6, because $\hat{v}$ is an ancestor of the difference node v of $M(C[p])$ and $M(C[p'])$. $\square$

Theorem 65. *If $h_o(M)$ is an output bound and $h_a(M)$ an ancestral bound for M, then $h(M) = 2 \cdot \mathrm{maxrhs}(M) + h_o(M) + h_a(M) + 1$ is a difference bound for M.*

Proof. Immediate from Lemmas 62 (for $\bar{v} = v$) and 64. $\square$

We end this section by proving that every dtla M has a computable output bound (Theorem 67). The reader who believes that this follows from [5, Theorem 4.5] can

skip the rest of this section. However, the proof of the next lemma can also serve as an introduction to the pumping technique used in Section 9 for computing an ancestral bound for M.

Lemma 66. *Let $p \in P$ such that the set $\{M(s) \mid s \in \llbracket p \rrbracket\}$ is finite. Then $\mathsf{ht}(M(s)) \leq \mathsf{maxrhs}(M) \cdot |Q|$ for every $s \in \llbracket p \rrbracket$.*

Proof. The (obvious) idea for the proof is that we consider an output path from the root to a leaf v in $M(s)$ that is longer than $\mathsf{maxrhs}(M) \cdot |Q|$. Then we find two ancestors $\hat{u}$ and $\bar{u}$ of the origin node u of v where the computation of M on s is in a cycle (i.e., arrives in the same state at $\hat{u}$ and $\bar{u}$) and produces a nonempty part of the output path. And then we pump the part of s between $\hat{u}$ and $\bar{u}$, thus pumping the path to v in $M(s)$ and obtaining arbitrarily long output paths. For a formal proof it is convenient to build a graph G_M^{o} of which each path corresponds to a pair (u, v) and a state q such that there is a sentential form ξ with $q_{0,p} \Rightarrow_s^* \xi$ and $\xi/v = q(u)$, i.e., such that $(q, u, \#) \in \mathsf{link}_s(v)$, see Lemma 52(1). The pumping of s then corresponds to the repetition of a cycle in G_M^{o}. For the notion of a directed edge-labeled graph see Section 2.

We construct a directed edge-labeled *dependency graph* G_M^{o} with node set Q and an edge with label (j, z) from q to q' if there is a rule $q(a(x_1 \colon p_{i,1}, \ldots, x_k \colon p_{i,k})) \to \zeta$ in R such that $z \in V(\zeta)$ and $\zeta/z = q'(x_j)$. Note that $|z| \leq \mathsf{maxrhs}(M)$. Each path $e_1 \cdots e_n$ in G_M^{o} (where each e_i is an edge) has a label in $\mathbb{N}_+^* \times \mathbb{N}_+^*$, obtained by the component-wise concatenation of the labels of $e_1, \ldots, e_n$.

Claim. There is a path π from $q_{0,p}$ to q with label (u, v) in G_M^{o} if and only if there is a tree $s \in \llbracket p \rrbracket$ such that $(q, u, \#) \in \mathsf{link}_s(v)$.

Proof of Claim. ($\Rightarrow$) We proceed by induction on the length of π. If π is empty, then $q = q_{0,p}$ and $(u, v) = (\varepsilon, \varepsilon)$, and so $(q, u, \#) \in \mathsf{link}_s(v)$ for every $s \in \llbracket p \rrbracket$ (by the definition of link). Now let $\pi' = \pi e$ where π is a path from $q_{0,p}$ to q with label (u, v), and e is an edge from q to q' with label (j, z). Thus, π' is a path from $q_{0,p}$ to q' with label (uj, vz). By the definition of G_M^{o}, the edge e is obtained from a rule $q(a(x_1 \colon p_1, \ldots, x_k \colon p_k)) \to \zeta$ such that $z \in V(\zeta)$ and $\zeta/z = q'(x_j)$. By induction, $(q, u, \#) \in \mathsf{link}_s(v)$. Choose $s_m \in \llbracket p_m \rrbracket$ for $m \in [k]$, and let $s' = s[u \leftarrow a(s_1, \ldots, s_k)]$. By Lemma 52(4), we have $\delta(s/u) = \rho(q)$ and so $\delta(s/u) = \delta(a(s_1, \ldots, s_k))$ because M is la-uniform. Hence $\delta(s) = \delta(s')$ and every proper ancestor of u is similar in s and s'. Consequently $(q, u, \#) \in \mathsf{link}_{s'}(v)$ by Lemma 59(1). Since $\mathsf{rhs}(q, s', u) = \zeta$, we obtain that $(q', uj, \#) \in \mathsf{link}_{s'}(vz)$ from the definition of link.

($\Leftarrow$) We proceed by induction on the length of u. If $u = \varepsilon$, then $q = q_{0,p}$ and $v = \varepsilon$ by the definition of link, and so the empty path satisfies the conditions. Now let $u' = uj$ with $u \in \mathbb{N}_+^*$ and $j \in \mathbb{N}_+$, and let $s \in \llbracket p \rrbracket$ such that $(q', u', \#) \in \mathsf{link}_s(v')$. By the definition of link there is a rule $q(a(x_1 \colon p_1, \ldots, x_k \colon p_k)) \to \zeta$ such that $a = \mathsf{lab}(s, u)$ and $p_m = \delta(s, um)$ for every $m \in [k]$; moreover, there exist $v \in \mathbb{N}_+^*$ and $z \in V(\zeta)$ such that $v' = vz$, $(q, u, \#) \in \mathsf{link}_s(v)$, and $\zeta/z = q'(x_j)$. By induction, there is a path π from $q_{0,p}$ to q with label (u, v). By the definition of G_M^{o} there is an edge e from q to q' with label (j, z). Hence πe is a path from $q_{0,p}$ to q' with label $(uj, vz) = (u', v')$. This ends the proof of the Claim.

Suppose that π_0 is a path from $q_{0,p}$ to q with label (u_0, v_0), and that π is a cycle from q to itself with label (u, v) such that $v \neq \varepsilon$. Consider the "pumped" path $\pi_0 \pi^k$

from $q_{0,p}$ to q, for any $k \geq 1$. By the Claim, there exists a tree $s_k \in [\![p]\!]$ such that $(q, u_0 u^k, \#) \in \mathsf{link}_{s_k}(v_0 v^k)$. Since $v_0 v^k \in V(M(s_k))$ and $|v_0 v^k| \geq k$, we obtain that $\mathsf{ht}(M(s_k)) \geq k$. That contradicts the fact that $\{M(s) \mid s \in [\![p]\!]\}$ is finite.

Now consider a tree $s \in [\![p]\!]$ and a leaf v of $M(s)$. Let $\mathsf{or}_s(v) = (q, u, z)$. Then, by Lemma 57, there exists an ancestor $\hat{v}$ of v such that $(q, u, \#) \in \mathsf{link}_s(\hat{v})$ and $v = \hat{v}z$. By the Claim, there is a path π from $q_{0,p}$ to q with label $(u, \hat{v})$. As shown above, if π contains a cycle, then its output label is empty. Hence there is a path π_1 without cycles from $q_{0,p}$ to q with label $(u_1, \hat{v})$. Then π_1 has at most $|Q| - 1$ edges. Since the second component of the label of every edge has length at most $\mathsf{maxrhs}(M)$, we obtain that $|\hat{v}| \leq \mathsf{maxrhs}(M) \cdot (|Q| - 1)$. Hence $|v| = |\hat{v}| + |z| \leq |\hat{v}| + \mathsf{maxrhs}(M) \leq \mathsf{maxrhs}(M) \cdot |Q|$. $\qquad \square$

Theorem 67. *The number* $\mathsf{maxrhs}(M) \cdot |Q| \cdot (|P| + 2)$ *is an output bound for* M.

Proof. For a set $\{q_M(s) \mid s \in \mathcal{T}_\Sigma, \delta(s) = p_1\}$ as in Definition 61(1), with $\rho(q) = p_1$, we construct the dtla M_1 from M by changing its p_1-axiom into $q(x_0)$. Application of Lemma 66 to M_1 and p_1 gives the upper bound $\mathsf{maxrhs}(M) \cdot |Q|$ on the height of the trees in the given set.

For a set $\{q_M(C[p]) \mid C \in \mathcal{C}_\Sigma, \delta(C[p]) = p_1, \delta(C[p']) = p_1'\}$ as in Definition 61(2), with $\rho(q) = p_1$, we first change M into M_1 as before. It remains to consider the set $\{M_1(C[p]) \mid C \in \mathcal{C}_\Sigma, \delta(C[p']) = p_1'\}$. In order to apply Lemma 66, we modify the dtla M_1°. We use a simple product construction to obtain from M_1° an equivalent dtla M_2 that "recognizes" input trees of the form $C[p]$ and computes $\delta(C[p'])$ by additional look-ahead. The dtla M_2 has the same input alphabet $\Sigma \cup P$ and output alphabet $\Delta \cup (Q \times P)$ as M° and M_1°. It has the same set Q of states as M and M_1, and the set of look-ahead states $P \times (P \cup \{1, 0\})$. Its transition function $\bar{\delta}$ is constructed in such a way that $\bar{\delta}(s) = (\bar{p}, \bar{p}')$ for $s \in \mathcal{T}_\Sigma(P)$, where $\bar{p}$ and $\bar{p}'$ are given as follows: $\bar{p} = \delta(s)$, $\bar{p}' = 1$ if $s \in \mathcal{T}_\Sigma$, $\bar{p}' = 0$ if s contains more than one occurrence of p or at least one occurrence of an element of $P - \{p\}$, and $\bar{p}' = \delta(C[p'])$ if $s = C[p]$ for some $C \in \mathcal{C}_\Sigma$. We leave the easy construction of $\bar{\delta}$ to the reader. The rules of M_2 are defined by: $\mathsf{rhs}_{M_2}(\bar{q}, a, (\bar{p}_1, \bar{p}_1'), \ldots, (\bar{p}_k, \bar{p}_k')) = \mathsf{rhs}_{M_1^\circ}(\bar{q}, a, \bar{p}_1, \ldots, \bar{p}_k)$, and the $(\bar{p}, \bar{p}')$-axiom of M_2 is the $\bar{p}$-axiom of M_1. Thus, M_2 ignores the additional look-ahead and hence is equivalent to M_1°. Note that M_2 is initialized, but need not be la-uniform. We finally turn M_2 into an equivalent initialized la-uniform dtla M_3 by an obvious variant of the construction in the proof of Lemma 13, such that M_3 has state set $Q_3 = Q \times (P \cup \{1, 0\})$ and la-map ρ_3 with $\rho_3(\langle \bar{q}, \bar{p}' \rangle) = \rho(\bar{q})$. We also leave this construction to the reader. Note that $\mathsf{maxrhs}(M_3) = \mathsf{maxrhs}(M)$. Application of Lemma 66 to M_3 and (p_1, p_1') gives the upper bound $\mathsf{maxrhs}(M_3) \cdot |Q_3| = \mathsf{maxrhs}(M) \cdot |Q| \cdot (|P| + 2)$ on the height of the trees in the given set. $\qquad \square$

9 A Dependency Graph for Output Branches

Convention. In this section we assume additionally that the initalized la-uniform dtla M is **ultralinear** with mapping $\mu : Q \to \mathbb{N}$ and **b-erasing** with graph E_M, see Section 3. These properties will be used (only) in the proof of Lemma 81.

For this M we will compute an ancestral bound $h_a(M)$, as defined in Definition 63. In view of this definition (and Lemma 6), it is technically convenient to combine a node v of an output tree with the sequence of labels of the proper ancestors of v, as follows.

For the output alphabet Δ, we define the *branch alphabet* $\Delta_\mathcal{B}$ by:

$$\Delta_\mathcal{B} = \{(d, j) \mid d \in \Delta, \mathsf{rk}(d) \geq 1, j \in [\mathsf{rk}(d)]\}.$$

A string in $\Delta_\mathcal{B}^*$ is called a *branch*. For a branch $v \in \Delta_\mathcal{B}^*$ we define $\mathsf{nod}(v) \in \mathbb{N}_+^*$ to be the sequence of numbers obtained from v by changing every (d, j) into j. For a tree $t = d(t_1, \ldots, t_k) \in \mathcal{T}_\Delta(Q \times P)$ with $k \in \mathbb{N}$, $d \in (\Delta \cup (Q \times P))^{(k)}$, and $t_1, \ldots, t_k \in \mathcal{T}_\Delta(Q \times P)$, we define the set $B(t) \subseteq \Delta_\mathcal{B}^*$ of *branches of* t inductively as follows:

$$B(d(t_1, \ldots, t_k)) = \{\varepsilon\} \cup \{(d, j)\, v \mid j \in [k], v \in B(t_j)\}.$$

The mapping nod, restricted to $B(t)$, is a bijection from $B(t)$ to $V(t)$. Intuitively, a branch v contains the node $\mathsf{nod}(v)$ and the labels of its proper ancestors (from the root to the node). For example, if $v = (a, 2)(b, 1)(b, 3)$ is a branch of t, then it corresponds to the node $\mathsf{nod}(v) = (2, 1, 3)$ of t, and moreover, the root of t has label a, node 2 has label b, and node $(2, 1)$ has label b too. For a branch $v \in B(t)$, we define $t/v = t/\mathsf{nod}(v)$ and $\mathsf{lab}(t, v) = \mathsf{lab}(t, \mathsf{nod}(v))$.

We will need the following lemma on branches. Roughly, it says that if $M(C[p])$ has a branch that is longer than any branch of $M(C[p'])$, then a prefix of that branch corresponds to a difference node of $M(C[p])$ and $M(C[p'])$. In this section, we rename p, p' into p_1, p_2.

Lemma 68. *Let $C \in \mathcal{C}_\Sigma$ and $p_1, p_2 \in P$. Let v be a branch of both $M(C[p_1])$ and $M(C[p_2])$, and let $w \in \Delta_\mathcal{B}^*$ be such that*

(1) vw is a branch of $M(C[p_1])$, and
(2) $|w| > \mathsf{ht}(M(C[p_2])/v)$.

Then there is a prefix w' of w such that $\mathsf{nod}(vw')$ is a difference node of $M(C[p_1])$ and $M(C[p_2])$.

Proof. Since $|w| > \mathsf{ht}(M(C[p_2])/v)$, vw is not a branch of $M(C[p_2])$. Let w' be the longest prefix of w such that vw' is a branch of $M(C[p_2])$. Since vw' is also a branch of $M(C[p_1])$, all proper ancestors of $\mathsf{nod}(vw')$ have the same label in $M(C[p_1])$ and $M(C[p_2])$. By Lemma 6, it remains to show that $\mathsf{nod}(vw')$ has different labels in $M(C[p_1])$ and $M(C[p_2])$. Since $w' \neq w$, there exist $k \geq 1$, $d \in \Delta^{(k)}$, and $j \in [k]$ such that $w'(d, j)$ is a prefix of w. Then $d = \mathsf{lab}(M(C[p_1]), vw')$. Suppose that also $d = \mathsf{lab}(M(C[p_2]), vw')$. Then $vw'(d, j)$ is a branch of $M(C[p_2])$, contradicting the choice of w'. $\qquad\square$

Again in view of Definition 63, we extend the definition of link to branches, and call it blink; we are, however, only interested in triples $(q, u, \#)$. For $s \in \mathcal{T}_\Sigma(P)$ and all $v \in \Delta_\mathcal{B}^*$, the sets $\mathsf{blink}_s(v)$ of triples $(q, u, \#)$ with $q \in Q$ and $u \in V(s)$, are defined inductively as follows:

(1) If $\delta(s) = p$, then $(q_{0,p}, \varepsilon, \#) \in \mathsf{blink}_s(\varepsilon)$.
(2) If $(q, u, \#) \in \mathsf{blink}_s(v)$ and $\mathsf{rhs}(q, s, u) = \zeta$,
    then $(\bar{q}, uj, \#) \in \mathsf{blink}_s(vz)$ for every $z \in B(\zeta)$ with $\zeta/z = \bar{q}(x_j)$.

Lemma 69. *For every $s \in \mathcal{T}_\Sigma(P)$, $v \in \Delta_\mathcal{B}^*$, $q \in Q$, and $u \in V(s)$, the following three statements are equivalent:*

(1) $(q, u, \#) \in \mathsf{blink}_s(v)$;
(2) $(q, u, \#) \in \mathsf{link}_s(\mathsf{nod}(v))$ *and* $v \in B(M(s))$;
(3) *there is a sentential form ξ of M° such that $q_{0,\delta(s)}(\varepsilon) \Rightarrow_s^* \xi$, $v \in B(\xi)$, and $\xi/v = q(u)$.*

Proof. Since the definition of blink_s closely follows the semantics of M°, it is straight-forward to show the equivalence of (1) and (3). This proves the equivalence of (1) and (2), by Lemmas 52(1) and 2(2). $\qquad\square$

With these definitions, we can reformulate Definition 63 as follows.

Lemma 70. *A number $h_\mathrm{a}(M) \in \mathbb{N}$ is an ancestral bound for M if and only if $\mathsf{diff}(M)$ is infinite or the following holds for every $C \in \mathcal{C}_\Sigma$, $p_1, p_2 \in P$, $u \in V(C)$, $v_1 \in B(M(C[p_1]))$, $v_2 \in B(M(C[p_2]))$, and $q_1, q_2 \in Q$:*
if $(q_1, u, \#) \in \mathsf{blink}_{C[p_1]}(v_1)$, $(q_2, u, \#) \in \mathsf{blink}_{C[p_2]}(v_2)$, and v_2 is a prefix of v_1, then $|v_1| - |v_2| \leq h_\mathrm{a}(M)$.

Thus, to determine an ancestral bound for M, we are interested in a node u of a Σ-context C and in two computations of M on the path from the root of C to u, one with input $C[p_1]$ and the other with input $C[p_2]$. The idea is, roughly, to find two ancestors $\hat{u}$ and $\bar{u}$ of u in C where each of these computations is in a cycle (i.e., arrives in the same state at $\hat{u}$ and $\bar{u}$), and then to pump the part of C between $\hat{u}$ and $\bar{u}$, leading to an infinite $\mathsf{diff}(M)$. For this it is technically helpful to build a graph G_M of which each path captures two such computations: those corresponding to $(q_1, u, \#) \in \mathsf{blink}_{C[p_1]}(v_1)$ and $(q_2, u, \#) \in \mathsf{blink}_{C[p_2]}(v_2)$ according to Lemma 69(3). The pumping of C then corresponds to the repetition of a cycle in the graph G_M. In this graph, we have to distinguish between the case that u is on or off the spine of C, where the spine of a context C is the path from the root to the unique occurrence of $\bot$ (as in the proof of Lemma 62). Formally, for $C \in \mathcal{C}_\Sigma$, the *spine* of C is the set $\mathsf{spi}(C) = \{u \in V(C) \mid C/u \in \mathcal{C}_\Sigma\}$.

For an introduction to the pumping technique using a graph, the reader is advised to study the proof of Lemma 66, where each path of the graph captures just one computation of M, corresponding to $(q, u, \#) \in \mathsf{link}_s(v)$. For the notion of a directed edge-labeled graph see Section 2.

For the dtla M, we define a finite directed edge-labeled *dependency graph* G_M whose nodes are all 3-tuples (q_1, q_2, b) with $q_1, q_2 \in Q$ and $b \in \{0, 1\}$, and whose edge labels are in $R \times R \times \mathbb{N}_+ \times \Delta_\mathcal{B}^* \times \Delta_\mathcal{B}^*$, more precisely, in the finite set of all 5-tuples (r_1, r_2, j, z_1, z_2) such that $r_1, r_2 \in R$, $j \in [\mathsf{rk}(a)]$ for some $a \in \Sigma$, and $z_i \in \Delta_\mathcal{B}^*$ is a branch of the right-hand side of r_i for $i = 1, 2$. The boolean b is called the *type* of the node; intuitively, $b = 1$ stands for "on the spine", and $b = 0$ for "off the spine". The edges of the dependency graph G_M are defined as follows, together with its "entry nodes".

(Ge) For every $\bar{p}_1, \bar{p}_2 \in P$, $(q_{0,\bar{p}_1}, q_{0,\bar{p}_2}, 1)$ is an *entry node* of G_M.

(G1) Let $k \geq 1$, $a \in \Sigma^{(k)}$, and $\ell, j \in [k]$. Let $(q_1, q_2, 1)$ be a node of G_M and, for $i = 1, 2$, let $r_i = q_i(a(x_1 : \bar{p}_{i,1}, \ldots, x_k : \bar{p}_{i,k})) \to \zeta_i$ be a rule of M, let $z_i \in \Delta_{\mathcal{B}}^*$, and let $q_i' \in Q$, such that
  (a) $\bar{p}_{1,m} = \bar{p}_{2,m}$ for all $m \in [k] - \{\ell\}$,
  (b) $z_i \in B(\zeta_i)$ and $\zeta_i/z_i = q_i'(x_j)$, for $i = 1, 2$.
  Then there is an edge labeled (r_1, r_2, j, z_1, z_2) from $(q_1, q_2, 1)$ to (q_1', q_2', b'), where $b' = 1$ if and only if $\ell = j$.

(G0) Let $k \geq 1$, $a \in \Sigma^{(k)}$, and $j \in [k]$. Let $(q_1, q_2, 0)$ be a node of G_M and, for $i = 1, 2$, let $r_i = q_i(a(x_1 : \bar{p}_{i,1}, \ldots, x_k : \bar{p}_{i,k})) \to \zeta_i$ be a rule of M, let $z_i \in \Delta_{\mathcal{B}}^*$, and let $q_i' \in Q$, such that
  (a) $\bar{p}_{1,m} = \bar{p}_{2,m}$ for all $m \in [k]$,
  (b) $z_i \in B(\zeta_i)$ and $\zeta_i/z_i = q_i'(x_j)$, for $i = 1, 2$.
  Then there is an edge labeled (r_1, r_2, j, z_1, z_2) from $(q_1, q_2, 0)$ to $(q_1', q_2', 0)$.

Note that there are no edges from a node of type 0 to a node of type 1. Note also that if (q_1, q_2, b) is a node of G_M, then so is (q_2, q_1, b); moreover, if there is an edge from (q_1, q_2, b) to (q_1', q_2', b') with label (r_1, r_2, j, z_1, z_2), then there is an edge from (q_2, q_1, b) to (q_2', q_1', b') with label (r_2, r_1, j, z_2, z_1).

A path $\pi = e_1 \cdots e_n$ in G_M (with $n \geq 0$ and $e_i \in E$ for every $i \in [n]$) has a *label* $(u, v_1, v_2) \in \mathbb{N}_+^* \times \Delta_{\mathcal{B}}^* \times \Delta_{\mathcal{B}}^*$ obtained by the component-wise concatenation of the last three components of the labels of $e_1, \ldots, e_n$. The *output label* of π is the pair of branches (v_1, v_2), denoted by $\mathrm{out}(\pi)$. We say that π is an *entry path* if it starts at an entry node, and that it is a (q_1, q_2, b)-*path* if it is an entry path that ends at the node (q_1, q_2, b). We will only be interested in the entry paths of G_M.

Let us now consider the intuition behind items (Ge), (G1), and (G2), continuing the discussion before the definition of G_M. An entry path of G_M, more precisely a (q_1, q_2, b)-path with label (u, v_1, v_2), is meant to capture two computations of M, corresponding to $(q_1, u, \#) \in \mathrm{blink}_{C[p_1]}(v_1)$ and $(q_2, u, \#) \in \mathrm{blink}_{C[p_2]}(v_2)$ according to Lemma 69(3), and moreover, the boolean b indicates whether or not u is on the spine of C. This is formally stated in Lemma 75. Since these two computations start at the root of C in the initial states $q_{0,\bar{p}_1}$ and $q_{0,\bar{p}_2}$ with $\bar{p}_i = \delta(C[p_i])$, and since the root of C is on its spine, this explains item (Ge) for the empty path at that entry node and for $u = v_1 = v_2 = \varepsilon$, see also condition (1) of the definition of blink. Items (G1) and (G0) extend a (q_1, q_2, b)-path with label (u, v_1, v_2) with one more edge, assuming that u is on/off the spine, respectively. The computations of M have arrived at node u of C, and now move to its child uj according to condition (2) of the definition of blink. To do this, M applies the rules r_1 with right-hand side $\zeta_1 = \mathrm{rhs}(q_1, C[p_1], u)$ and r_2 with right-hand side $\zeta_2 = \mathrm{rhs}(q_2, C[p_2], u)$ in $C[p_1]$ and $C[p_2]$, respectively, and it picks branches z_i in ζ_i that lead to $q_i'(x_j)$. If u is on the spine, let $u\ell$ be the (unique) child of u that is on the spine. Then the look-ahead state at each child um of u with $m \neq \ell$ is the same in $C[p_1]$ and $C[p_2]$ (and if u is not on the spine, then this holds for all its children). The path is now extended by one edge to a (q_1', q_2', b')-path with label $(uj, v_1 z_1, v_2 z_2)$, where $b' = 1$ if $b = 1$ and $j = \ell$ (i.e., uj is the unique child of u on the spine), and

$b' = 0$ otherwise. The edge is labeled additionally with the rules r_1 and r_2 for technical reasons.

We now give two examples of a dependency graph.

Example 71. We first consider a very simple initialized la-uniform dtla M that changes the label of every node u of the input tree s into the label of the left-most leaf of s/u, and changes the label of every leaf into e. Let $\Sigma = \{\sigma^{(2)}, a^{(0)}, b^{(0)}\}$ and $\Delta = \{a^{(2)}, b^{(2)}, e^{(0)}\}$. The look-ahead states of M are p_a and p_b, which compute the label of the left-most leaf: $\delta(y) = p_y$ and $\delta(\sigma, p_y, p_z) = p_y$ for all $y, z \in \{a, b\}$. The states of M are q_a and q_b. For $y, w \in \{a, b\}$, the axioms of M are $A(p_y) = q_y(x_0)$ and its rules are $q_y(y) \to e$ and $r_{yw} = q_y(\sigma(x_1 : p_y, x_2 : p_w)) \to y(q_y(x_1), q_w(x_2))$.

The dependency graph G_M has eight nodes $(q_y, q_z, \bar{b})$, for $y, z \in \{a, b\}$ and $\bar{b} \in \{0, 1\}$. By (Ge), the four nodes $(q_y, q_z, 1)$ are the entry nodes of G_M. Applying (G1) to these entry nodes the following 32 edges are obtained, for all $y, z, w, w' \in \{a, b\}$:

(i) $j = 1, \ell = 1$: from $(q_y, q_z, 1)$ to itself labeled $(r_{yw}, r_{zw}, 1, (y, 1), (z, 1))$
(ii) $j = 1, \ell = 2$: from $(q_y, q_y, 1)$ to $(q_y, q_y, 0)$ labeled $(r_{yw}, r_{yw'}, 1, (y, 1), (y, 1))$
(iii) $j = 2, \ell = 2$: from $(q_y, q_y, 1)$ to $(q_w, q_{w'}, 1)$ labeled $(r_{yw}, r_{yw'}, 2, (y, 2), (y, 2))$
(iv) $j = 2, \ell = 1$: from $(q_y, q_z, 1)$ to $(q_w, q_w, 0)$ labeled $(r_{yw}, r_{zw}, 2, (y, 2), (z, 2))$.

Then, applying (G0) to all nodes $(q_y, q_y, 0)$, the following 8 edges are obtained, for all $y, w \in \{a, b\}$:

(v) $j = 1$: from $(q_y, q_y, 0)$ to itself labeled $(r_{yw}, r_{yw}, 1, (y, 1), (y, 1))$
(vi) $j = 2$: from $(q_y, q_y, 0)$ to $(q_w, q_w, 0)$ labeled $(r_{yw}, r_{yw}, 2, (y, 2), (y, 2))$.

Note that the nodes $(q_a, q_b, 0)$ and $(q_b, q_a, 0)$ are not reachable from an entry node and hence we do not care about the edges that leave them (in fact, there are none).

Let us consider the Σ-context $C = \sigma(b, \sigma(C', b))$ with $C' = \sigma(\bot, \sigma(a, \sigma(b, a)))$. Clearly, we have $M(C[p_a]) = b(e, a(t_a, e))$ with $t_a = a(\langle q_a, p_a \rangle, a(e, b(e, e)))$, and $M(C[p_b]) = b(e, b(t_b, e))$ with $t_b = b(\langle q_b, p_b \rangle, a(e, b(e, e)))$. Now consider the input node $u = (2, 1, 2, 2)$, i.e., the node with $C/u = \sigma(b, a)$, and note that $u \notin \mathrm{spi}(C)$. From Lemma 69(3), or from the definition of blink, it should be clear that $(q_b, u, \#) \in \mathrm{blink}_{C[p_a]}(v_a)$ and $(q_b, u, \#) \in \mathrm{blink}_{C[p_b]}(v_b)$ for the branches $v_a = (b, 2)(a, 1)(a, 2)(a, 2)$ and $v_b = (b, 2)(b, 1)(b, 2)(a, 2)$. According to the intuition described before, this corresponds to the $(q_b, q_b, 0)$-path $e_1 e_2 e_3 e_4$ with label (u, v_a, v_b) where, using (iii), (i), (iv), and (vi) above, respectively:

- e_1 is from entry node $(q_b, q_b, 1)$ to $(q_a, q_b, 1)$ labeled $(r_{ba}, r_{bb}, 2, (b, 2), (b, 2))$,
- e_2 is from $(q_a, q_b, 1)$ to itself labeled $(r_{ab}, r_{bb}, 1, (a, 1), (b, 1))$,
- e_3 is from $(q_a, q_b, 1)$ to $(q_a, q_a, 0)$ labeled $(r_{aa}, r_{ba}, 2, (a, 2), (b, 2))$, and
- e_4 is from $(q_a, q_a, 0)$ to $(q_b, q_b, 0)$ labeled $(r_{ab}, r_{ab}, 2, (a, 2), (a, 2))$.

Note that the prefixes of this path correspond in a similar way to the ancestors of u. □

The dtla M of the above example is very simple. If $(q, u, \#) \in \mathrm{blink}_{C[p]}(v)$ then $|u| = |v|$ and correspondingly, if (u, v_1, v_2) is the label of an entry path in G_M then $|v_1| = |v_2|$. The next example is more complicated in that respect, and hence can serve as an example later on.

Example 72. Let $\Sigma = \{\tau^{(2)}, \sigma^{(1)}, a^{(0)}, b^{(0)}\}$ and let $\Delta = \{\tau^{(2)}, \sigma_a^{(1)}, \sigma_b^{(1)}, a^{(0)}, b^{(0)}, e^{(0)}\}$. We consider an initialized la-uniform dtla M such that, for all $m, n \in \mathbb{N}$ and $y \in \{a, b\}$, $M(\tau(\sigma^m y, \sigma^n a)) = \tau(\sigma_y^m y, M_y(\sigma^n a))$ where $M_y(a) = a$,

$$M_a(\sigma^{n+1} a) = \tau(M_a(\sigma^n a), a) \text{ and } M_b(\sigma^{n+1} a) = \tau(\tau(a, M_b(\sigma^n a)), a).$$

We will not be interested in other input trees, which are translated by M into trees with at least one occurrence of e. The look-ahead states of M are p_a and p_b, which, as in the previous example, compute the label of the left-most leaf: $\delta(y) = p_y$, $\delta(\sigma, p_y) = p_y$, and $\delta(\tau, p_y, p_z) = p_y$ for all $y, z \in \{a, b\}$. The states of M are all q_{iy} with $i \in \{0, 1, 2\}$ and $y \in \{a, b\}$; the la-map of M is $\rho(q_{0y}) = \rho(q_{1y}) = p_y$ and $\rho(q_{2y}) = p_a$. For $y \in \{a, b\}$, the axioms of M are $A(p_y) = q_{0y}(x_0)$, and it has the following rules, where the missing rules all have right-hand side e. First, it has the rules

$$\begin{aligned}
r_{0y} &= q_{0y}(\tau(x_1{:}p_y, x_2{:}p_a)) \to \tau(q_{1y}(x_1), q_{2y}(x_2)), \\
r_{1y} &= q_{1y}(\sigma(x_1{:}p_y)) \to \sigma_y(q_{1y}(x_1)), \\
r_{2a} &= q_{2a}(\sigma(x_1{:}p_a)) \to \tau(q_{2a}(x_1), a), \text{ and} \\
r_{2b} &= q_{2b}(\sigma(x_1{:}p_a)) \to \tau(\tau(a, q_{2b}(x_1)), a).
\end{aligned}$$

Second, it has the rules $q_{1y}(y) \to y$ and $q_{2y}(a) \to a$.

The dependency graph G_M has the four entry nodes $(q_{0y}, q_{0z}, 1)$ for $y, z \in \{a, b\}$. We will not construct all edges of G_M, but just mention four interesting ones. First, by (G1) with $\ell = j = 1$, there is an edge e_{01} from $(q_{0a}, q_{0b}, 1)$ to $(q_{1a}, q_{1b}, 1)$ with label $(r_{0a}, r_{0b}, 1, (\tau, 1), (\tau, 1))$. Second, by (G1) with $\ell = 1$ and $j = 2$, there is an edge e_{02} from $(q_{0a}, q_{0b}, 1)$ to $(q_{2a}, q_{2b}, 0)$ with label $(r_{0a}, r_{0b}, 2, (\tau, 2), (\tau, 2))$. Third, by (G1), there is an edge e_1 from $(q_{1a}, q_{1b}, 1)$ to itself with label $(r_{1a}, r_{1b}, 1, (\sigma_a, 1), (\sigma_b, 1))$. And finally, by (G0), there is an edge e_2 from $(q_{2a}, q_{2b}, 0)$ to itself with the label $(r_{2a}, r_{2b}, 1, (\tau, 1), (\tau, 1)(\tau, 2))$. □

As observed above, we express in Lemma 75 the meaning of the entry paths in G_M. But we will need a stronger version of the implication $(1) \Rightarrow (2)$ of that lemma, which we prove now. Recall from the paragraph before Theorem 15 that $\mathsf{fix}(M)$ is a fixed set of representatives of the equivalence classes $[\![p]\!]$, $p \in P$.

Lemma 73. *Let $\pi = e_1 \cdots e_n$, $n \geq 0$, be an entry path in G_M, where e_m is an edge of G_M for every $m \in [n]$. For every m, $0 \leq m \leq n$, let the prefix $e_1 \cdots e_m$ of π be a $(q_1^{(m)}, q_2^{(m)}, b^{(m)})$-path with label $(u^{(m)}, v_1^{(m)}, v_2^{(m)})$. Then there are $C \in \mathcal{C}_\Sigma$ and $p_1, p_2 \in P$ such that for every m, $0 \leq m \leq n$, and $i = 1, 2$,*

(a) $(q_i^{(m)}, u^{(m)}, \#) \in \mathsf{blink}_{C[p_i]}(v_i^{(m)})$,
(b) $b^{(m)} = 1$ *if and only if* $u^{(m)} \in \mathsf{spi}(C)$,
(c) *if $m \leq n - 1$ and e_{m+1} has label (r_1, r_2, j, z_1, z_2), then*
    (1) $\mathsf{rhs}(q_i^{(m)}, C[p_i], u^{(m)}) = \mathsf{rhs}(r_i)$ *and*
    (2) *for every $j' \in \mathbb{N}_+$ such that $u^{(m)} j'$ is a node of C, if $j' \neq j$ or $m = n - 1$, then $C/u^{(m)} j' \in \mathsf{fix}(M) \cup \{\bot\}$.*

Proof. We proceed by induction on the length n of π. If $n = 0$, i.e., π is empty, then $m = 0$, $u^{(0)} = v_1^{(0)} = v_2^{(0)} = \varepsilon$, and $(q_1^{(0)}, q_2^{(0)}, b^{(0)}) = (q_{0,p_1}, q_{0,p_2}, 1)$ for $p_1, p_2 \in P$. Let $C = \bot$. It is easy to check requirements (a) and (b) for these C, p_1 and p_2; requirement (c) holds trivially.

Now let $\pi' = \pi e$ where π is a (q_1, q_2, b)-path with label (u, v_1, v_2), and where the last edge e has label (r_1, r_2, j, z_1, z_2) and ends at node (q_1', q_2', b'). Then π' is a (q_1', q_2', b')-path with label (u', v_1', v_2') where $u' = uj$ and $v_i' = v_i z_i$. Let r_i be the rule $q_i(a(x_1 : \bar{p}_{i,1}, \ldots, x_k : \bar{p}_{i,k})) \to \zeta_i$ of M, hence $k \geq 1$, $a \in \Sigma^{(k)}$, $j \in [k]$, and $z_i \in B(\zeta_i)$ with $\zeta_i / z_i = q_i'(x_j)$ for $i = 1, 2$. By induction, requirements (a), (b), and (c) hold for some $C \in \mathcal{C}_\Sigma$ and $p_1, p_2 \in P$, for every prefix of π. In particular (for $m = n$), $(q_i, u, \#) \in \mathsf{blink}_{C[p_i]}(v_i)$, and $b = 1$ if and only if $u \in \mathsf{spi}(C)$. We first consider the case where $b = 1$, i.e., the last edge e is obtained by (G1). Then there exists $\ell \in [k]$ such that $\bar{p}_{1,m} = \bar{p}_{2,m}$ for all $m \in [k] - \{\ell\}$, and $b' = 1$ if and only if $\ell = j$. Let $p_1' = \bar{p}_{1,\ell}$ and $p_2' = \bar{p}_{2,\ell}$. Let $s_\ell = \bot$ and for $m \in [k] - \{\ell\}$, let s_m be the unique tree in $\mathsf{fix}(M)$ such that $\delta(s_m) = \bar{p}_{1,m} = \bar{p}_{2,m}$. Define $C' = C[u \leftarrow C_a]$ where C_a is the Σ-context $a(s_1, \ldots, s_k)$; since $u \in \mathsf{spi}(C)$, C' is a Σ-context. We now claim that requirements (a), (b), and (c) hold for C' and p_1', p_2', for every prefix of π'.

Obviously, $u' = uj$ is a branch of C', and $C'/u\ell = \bot$. Hence, $u' \in \mathsf{spi}(C')$ if and only if $\ell = j$ if and only if $b' = 1$. For every prefix $e_1 \cdots e_m$ of the path π, by the definition of the label of a path, $u^{(m)}$ is a prefix of u, i.e., an ancestor of u. Hence, $u^{(m)} \in \mathsf{spi}(C')$ if and only if $u^{(m)} \in \mathsf{spi}(C)$ if and only if $b^{(m)} = 1$ (and in fact, since $b = 1$, they are all true). This proves requirement (b).

Since $(q_i, u, \#) \in \mathsf{blink}_{C[p_i]}(v_i)$, Lemmas 69 and 52(4) imply that $\delta(C[p_i]/u) = \rho(q_i)$ which equals $\delta(C_a[p_i'])$ by definition of C_a (and because M is la-uniform). Hence $\delta(C[p_i]/u) = \delta(C'[p_i']/u)$, and so $\delta(C[p_i]) = \delta(C'[p_i'])$ and every proper ancestor of u is similar in $C[p_i]$ and $C'[p_i']$. It now follows from Lemmas 59(1) and 69 that $(q_i, u, \#) \in \mathsf{blink}_{C'[p_i']}(v_i)$. Since $\mathsf{rhs}(q_i, C'[p_i'], u) = \mathsf{rhs}(q_i, C_a[p_i'], \varepsilon) = \zeta_i$, the definition of blink implies that $(q_i', uj, \#) \in \mathsf{blink}_{C'[p_i']}(v_i z_i)$, i.e., that $(q_i', u', \#) \in \mathsf{blink}_{C'[p_i']}(v_i')$. Since $u^{(m)}$ is an ancestor of u for every prefix $e_1 \cdots e_m$ of π, Lemmas 59(1) and 69 also imply that $(q_i^{(m)}, u^{(m)}, \#) \in \mathsf{blink}_{C'[p_i']}(v_i^{(m)})$. This proves requirement (a).

It should be clear from the observations in the previous two paragraphs that requirement (c) holds for every proper prefix $e_1 \cdots e_m$ of π. In fact, $u^{(m)}$ is a proper ancestor of u, and $\mathsf{rhs}(q, C[p_i], \hat{u}) = \mathsf{rhs}(q, C'[p_i'], \hat{u})$ for every proper ancestor $\hat{u}$ of u (and every q), because $\hat{u}$ is similar in $C[p_i]$ and $C'[p_i']$. Moreover, if $u^{(m)}j' \neq u^{(m+1)}$ then $C/u^{(m)}j' = C'/u^{(m)}j'$. For the proper prefix π of π' and the edge e, requirement (c) holds by definition of the context C'.

The proof for the case where e is obtained by (G0) is similar, with $b = b' = 0$. Since u is not on the spine of C, we let $s_m \in \mathsf{fix}(M)$ with $\delta(s_m) = \bar{p}_{1,m} = \bar{p}_{2,m}$ for every $m \in [k]$, and we take $p_1' = p_1$ and $p_2' = p_2$. $\qquad\square$

Example 74. For $n \in \mathbb{N}_+$, consider the entry path $\pi = e_{01} e_1^{n-1}$ in G_M, where M is the dtla of Example 72. It is a path from $(q_{0a}, q_{0b}, 1)$ to $(q_{1a}, q_{1b}, 1)$, and it has label $(1^n, (\tau, 1)(\sigma_a, 1)^{n-1}, (\tau, 1)(\sigma_b, 1)^{n-1})$ where 1^n is the sequence $(1, \ldots, 1)$ of length n. Let $\mathsf{fix}(M) = \{a, b\}$. Then the requirements of Lemma 73 are fulfilled for $C = \tau(\sigma^{n-1}\bot, a)$ and for $p_1 = p_a$ and $p_2 = p_b$.

Now consider the entry path $\pi = e_{02}e_2^{n-1}$ from $(q_{0a}, q_{0b}, 1)$ to $(q_{2a}, q_{2b}, 0)$ with label $(2 \cdot 1^{n-1}, (\tau, 2)(\tau, 1)^{n-1}, (\tau, 2)((\tau, 1)(\tau, 2))^{n-1})$. The requirements of Lemma 73 for this path are fulfilled for $C = \tau(\bot, \sigma^{n-1}a)$ and for $p_1 = p_a$ and $p_2 = p_b$. □

In the next lemma we express the meaning of the entry paths in G_M.

Lemma 75. *Let $q_1, q_2 \in Q$, $b \in \{0, 1\}$, $u \in \mathbb{N}_+^*$, and $v_1, v_2 \in \Delta_\mathcal{B}^*$. Then the following two statements are equivalent:*

(1) there is a (q_1, q_2, b)-path in G_M with label (u, v_1, v_2);
(2) there are $C \in \mathcal{C}_\Sigma$ and $p_1, p_2 \in P$ such that
  (a) $(q_1, u, \#) \in \mathsf{blink}_{C[p_1]}(v_1)$ and $(q_2, u, \#) \in \mathsf{blink}_{C[p_2]}(v_2)$, and
  (b) $b = 1$ if and only if $u \in \mathsf{spi}(C)$.

Proof. $(1) \Rightarrow (2)$.   Immediate from Lemma 73 for $m = n$ (and disregarding (c)).

$(2) \Rightarrow (1)$.   This proof is similar to the proof of Lemma 73, but it is easier because in the induction step we can use the same C, p_1, and p_2. An illustration of the proof can be found in Example 71. Here we proceed by induction on the length of u. Let $u = \varepsilon$. Then $u \in \mathsf{spi}(C)$ and so $b = 1$. Since $(q_i, \varepsilon, \#) \in \mathsf{blink}_{C[p_i]}(v_i)$, it follows from the definition of blink that $v_i = \varepsilon$, and that $q_i = q_{0, \bar{p}_i}$ where $\bar{p}_i = \delta(C[p_i])$. Hence $(q_1, q_2, b) = (q_{0, \bar{p}_1}, q_{0, \bar{p}_2}, 1)$ is an entry node. This proves statement (1) for this case, because the empty path from (q_1, q_2, b) to itself has label $(\varepsilon, \varepsilon, \varepsilon)$.

Now let $u' = uj$ with $u \in \mathbb{N}_+^*$ and $j \in \mathbb{N}_+$. Let $C \in \mathcal{C}_\Sigma$ and $p_1, p_2 \in P$ (and consider q_i', v_i', and b') such that $(q_i', u', \#) \in \mathsf{blink}_{C[p_i]}(v_i')$ for $i = 1, 2$ and such that $b' = 1$ if and only if $u' \in \mathsf{spi}(C)$. By the definition of blink, there exists a rule r_i of the form $q_i(a(x_1 : \bar{p}_{i,1}, \ldots, x_k : \bar{p}_{i,k})) \rightarrow \zeta_i$ such that $a = \mathsf{lab}(C, u)$ with $k \geq 1$, $a \in \Sigma^{(k)}$, and $j \in [k]$, and such that $\bar{p}_{i,m} = \delta(C[p_i]/um)$ for every $m \in [k]$; moreover, there exist $v_i \in \Delta_\mathcal{B}^*$ and $z_i \in B(\zeta_i)$ such that $v_i' = v_i z_i$, $(q_i, u, \#) \in \mathsf{blink}_{C[p_i]}(v_i)$, and $\zeta_i / z_i = q_i'(x_j)$. We first consider the case where $u \in \mathsf{spi}(C)$. Then, by induction, there is a $(q_1, q_2, 1)$-path π in G_M with label (u, v_1, v_2). Let $\ell \in [k]$ be such that $u\ell \in \mathsf{spi}(C)$; so, $b' = 1$ if and only if $\ell = j$. We observe that $\bar{p}_{1,m} = \bar{p}_{2,m}$ for all $m \in [k] - \{\ell\}$, because $C/um \in \mathcal{T}_\Sigma$. So, by (G1), there is an edge e labeled (r_1, r_2, j, z_1, z_2) from $(q_1, q_2, 1)$ to (q_1', q_2', b'). Hence, πe is a (q_1', q_2', b')-path with label $(uj, v_1 z_1, v_2 z_2) = (u', v_1', v_2')$.

The proof for the case where $u \notin \mathsf{spi}(C)$ is similar. Then, by induction, there is a $(q_1, q_2, 0)$-path π. Of course $u' \notin \mathsf{spi}(C)$, and so $b' = 0$. Since now $\bar{p}_{1,m} = \bar{p}_{2,m}$ for all $m \in [k]$, there is an edge e from $(q_1, q_2, 0)$ to (q_1', q_2', b') by (G0). □

For strings v_1, v_2, we define $\mathsf{diff}(v_1, v_2)$ to be the pair of strings (w_1, w_2) such that $v_1 = vw_1$ and $v_2 = vw_2$ where v is the longest common prefix of v_1 and v_2. Note that if $\mathsf{diff}(v_1, v_2) = (w_1, w_2)$, then $\mathsf{diff}(v_1 z_1, v_2 z_2) = \mathsf{diff}(w_1 z_1, w_2 z_2)$ for all strings z_1 and z_2. For a path π in G_M, we define $\mathsf{diff}(\pi) = \mathsf{diff}(\mathsf{out}(\pi))$. So, if $\pi = \pi_1 \pi_2$ then $\mathsf{diff}(\pi) = \mathsf{diff}(\mathsf{diff}(\pi_1) \cdot \mathsf{out}(\pi_2))$, where $\cdot$ is component-wise concatenation.

We say that a pair of strings $(w_1, w_2) \in \Delta_\mathcal{B}^* \times \Delta_\mathcal{B}^*$ is *ancestral* if $w_1 = \varepsilon$ or $w_2 = \varepsilon$ (or both). A path π in G_M is *ancestral* if $\mathsf{diff}(\pi)$ is ancestral. Thus, if $\mathsf{out}(\pi) = (v_1, v_2)$, then π is ancestral if and only if v_1 is a prefix of v_2 or vice versa. Clearly, if π is ancestral then every prefix of π is ancestral: if $\pi = \pi_1 \pi_2$ and $\mathsf{diff}(\pi_1)$ is not ancestral (i.e., both

its components are nonempty and their first symbols differ), then $\mathsf{diff}(\pi_1) \cdot \mathsf{out}(\pi_2)$ is not ancestral and $\mathsf{diff}(\pi) = \mathsf{diff}(\pi_1) \cdot \mathsf{out}(\pi_2)$.

By Lemmas 70 and 75, and using the above definitions, we can again reformulate the definition of ancestral bound (Definition 63), as follows.

Lemma 76. *A number $h_{\mathrm{a}}(M) \in \mathbb{N}$ is an ancestral bound for M if and only if $\mathsf{diff}(M)$ is infinite or the following holds for every ancestral entry path π in G_M: if $\mathsf{diff}(\pi) = (w, \varepsilon)$, then $|w| \leq h_{\mathrm{a}}(M)$.*

In the next lemma we show that every dtla M has an ancestral bound. It will be convenient to prove a slightly stronger result, where we consider more entry paths in G_M than the ancestral ones.

We say that a pair of strings $(w_1, w_2) \in \Delta_{\mathcal{B}}^* \times \Delta_{\mathcal{B}}^*$ *splits* if there exist $d \in \Delta^{(k)}$ and $j_1, j_2 \in [k]$ with $j_1 \neq j_2$ such that the first symbol of w_i is (d, j_i). For $(v_1, v_2) \in \Delta_{\mathcal{B}}^* \times \Delta_{\mathcal{B}}^*$, the fact that $\mathsf{diff}(v_1, v_2)$ splits means that there exists a tree $t \in \mathcal{T}_\Delta$ such that v_1 and v_2 are branches of t that correspond to distinct leaves of t. A path π in G_M is *nonsplitting* if $\mathsf{diff}(\pi)$ does not split, and *splitting* if $\mathsf{diff}(\pi)$ splits. Note that an ancestral pair of strings does not split, and hence every ancestral path in G_M is nonsplitting.

Recall again from the paragraph before Theorem 15 that $\mathsf{fix}(M)$ is a fixed set of representatives of the equivalence classes $\llbracket p \rrbracket$, $p \in P$. We define

$$\mathsf{maxfix}(M) = \max\{\mathsf{ht}(q_M(s)) \mid q \in Q,\ s \in \mathsf{fix}(M),\ \rho(q) = \delta(s)\}.$$

Note that since $\mathsf{fix}(M)$ is finite, $\mathsf{maxfix}(M) \in \mathbb{N}$.

The next lemma implies that $\mathsf{maxfix}(M) + \mathsf{lubdiff}(M)$ is an ancestral bound for M when $\mathsf{diff}(M)$ is finite, and so every dtla has an ancestral bound.

Lemma 77. *Let π be a nonsplitting entry path in G_M. If $\mathsf{diff}(\pi) = (w_1, w_2)$, then $|w_i| \leq \mathsf{maxfix}(M) + \mathsf{lubdiff}(M)$ for $i = 1, 2$.*

Proof. Let π be a (q_1, q_2, b)-path with label (u, v_1, v_2) and let $\mathsf{diff}(v_1, v_2) = (w_1, w_2)$. For reasons of symmetry, it suffices to show that $|w_1| \leq \mathsf{maxfix}(M) + \mathsf{lubdiff}(M)$. Thus, we assume that $w_1 \neq \varepsilon$ (and hence $\pi \neq \varepsilon$). We consider two cases, depending on the value of b.

Case 1: $b = 1$. In this case we show that $|w_1| \leq \mathsf{lubdiff}(M)$. By Lemma 73 there are $C \in \mathcal{C}_\Sigma$ and $p_1, p_2 \in P$ such that $(q_i, u, \#) \in \mathsf{blink}_{C[p_i]}(v_i)$ for $i = 1, 2$, and $C/u = \bot$. Hence, by Lemma 69, there is a reachable sentential form ξ for $C[p_i]$ such that $v_i \in B(\xi)$ and $\xi/v_i = q_i(u)$. Since $C[p_i]/u = p_i$, Lemma 2(2) implies that $v_i \in B(M(C[p_i]))$ and $M(C[p_i])/v_i = \langle q_i, p_i \rangle$. Let v be the longest common prefix of v_1 and v_2. So $v_i = vw_i$. Since (w_1, w_2) does not split, $\mathsf{nod}(v)$ is a difference node of $M(C[p_1])$ and $M(C[p_2])$ by Lemma 6: each of its proper ancestors has the same label in $M(C[p_1])$ and $M(C[p_2])$, but the node itself does not have the same label in $M(C[p_1])$ and $M(C[p_2])$. In fact, if w_1 and w_2 are both nonempty, then their first symbols are $(d_1, j_1), (d_2, j_2)$ with $d_1 \neq d_2$, and $\mathsf{nod}(v)$ has label d_i in $M(C[p_i])$. If $w_2 = \varepsilon$, then $v = v_2$ and hence $\mathsf{nod}(v)$ has label $\langle q_2, p_2 \rangle$ in $M(C[p_2])$; also, $\mathsf{nod}(vw_1)$ has label $\langle q_1, p_1 \rangle$ in $M(C[p_1])$ and hence the label of $\mathsf{nod}(v)$ is in Δ (because $w_1 \neq \varepsilon$). Thus, since $\mathsf{nod}(v)$ is a difference node of $M(C[p_1])$ and $M(C[p_2])$, $|w_1| \leq \mathsf{ht}(M(C[p_1])/v) \leq \mathsf{lubdiff}(M)$.

Case 2: $b = 0$. By Lemma 73, there exist $C \in \mathcal{C}_\Sigma$ and $p_1, p_2 \in P$ such that $(q_i, u, \#) \in \mathsf{blink}_{C[p_i]}(v_i)$ and $C/u \in \mathsf{fix}(M)$. Note that v_i is a branch of $M(C[p_i])$ by Lemma 69, and that $M(C[p_i])/v_i = q_{iM}(C/u)$ by Lemma 54(1).

First assume that $w_2 \neq \varepsilon$. This is similar to Case 1 above. Let v be the longest common prefix of v_1 and v_2. So $v_i = vw_i$. Since (w_1, w_2) does not split and both w_1 and w_2 are nonempty, $\mathsf{nod}(v)$ is a difference node of $M(C[p_1])$ and $M(C[p_2])$. So $|w_1| \leq \mathsf{ht}(M(C[p_1])/v) \leq \mathsf{lubdiff}(M)$.

Now assume that $w_2 = \varepsilon$, and so $v_1 = v_2 w_1$. If $|w_1| \leq \mathsf{maxfix}(M)$ then we are ready. If $|w_1| > \mathsf{maxfix}(M)$, then $|w_1| > \mathsf{ht}(q_{2M}(C/u))$ because $C/u \in \mathsf{fix}(M)$, and so $|w_1| > \mathsf{ht}(M(C[p_2])/v_2)$. Thus, the branch $v_2 w_1$ of $M(C[p_1])$ is longer than any branch of $M(C[p_2])$ with prefix v_2. By Lemma 68 (with $v := v_2$ and $w := w_1$) there is a difference node $\mathsf{nod}(v_2 w'_1)$ of $M(C[p_1])$ and $M(C[p_2])$ where w'_1 is a prefix of w_1. Since $\mathsf{nod}(v_2 w'_1)$ is a node of $M(C[p_2])$, $|w'_1| \leq \mathsf{ht}(M(C[p_2])/v_2) = \mathsf{ht}(q_{2M}(C/u))$. Hence, $|w_1| \leq |w'_1| + \mathsf{ht}(M(C[p_1])/v_2 w'_1) \leq \mathsf{ht}(q_{2M}(C/u)) + \mathsf{lubdiff}(M)$. This shows that $|w_1| \leq \mathsf{maxfix}(M) + \mathsf{lubdiff}(M)$ by the definition of $\mathsf{maxfix}(M)$. □

Corollary 78. *If* $\mathsf{diff}(M)$ *is finite, then the set* $\{\mathsf{diff}(\pi) \mid \pi$ *is a nonsplitting entry path in* $G_M\}$ *is finite.*

We now turn to the pumping of Σ-contexts, as discussed before. In terms of G_M that corresponds to the repetition of a cycle. In the next two lemmas we show that pumping (i.e., repeating) a cycle in an ancestral entry path produces again an ancestral path, and in Lemma 83 we show that this leads to infinitely many $\mathsf{diff}(\pi)$ values and hence to an infinite $\mathsf{diff}(M)$ by Corollary 78.

A path π in G_M is a cycle, and in particular a (q_1, q_2, b)-*cycle*, if it is nonempty and leads from the node (q_1, q_2, b) to itself; note that every node of π is of type b. For every $i \geq 1$, we denote by π^i the i-fold concatenation $\pi \cdots \pi$.

Lemma 79. *Let* $\mathsf{diff}(M)$ *be finite. Let* π_0 *be a* (q_1, q_2, b)-*path in* G_M, *and* π *be a* (q_1, q_2, b)-*cycle in* G_M *with output label* (v_1, v_2) *such that* $v_1 v_2 \neq \varepsilon$. *If* $\pi_0 \pi$ *is nonsplitting, then it is ancestral.*

Proof. The idea of the proof is that if $\pi_0 \pi$ is not ancestral, then we can pump π and thus obtain nonsplitting paths with arbitrarily large diff values, contradicting Corollary 78.

Let $\mathsf{diff}(\pi_0 \pi) = (w_1, w_2)$. Suppose that $\pi_0 \pi$ is not ancestral, i.e., both w_1 and w_2 are nonempty. Since (w_1, w_2) is nonsplitting, $(w_1, w_2) = ((d_1, j_1)w'_1, (d_2, j_2)w'_2)$ with $d_1 \neq d_2$. For every $k \geq 1$, consider the (q_1, q_2, b)-path $\pi_0 \pi^{k+1}$. Now $\mathsf{diff}(\pi_0 \pi^{k+1}) = \mathsf{diff}(\mathsf{diff}(\pi_0 \pi) \cdot \mathsf{out}(\pi^k)) = \mathsf{diff}(w_1 v_1^k, w_2 v_2^k) = ((d_1, j_1)w'_1 v_1^k, (d_2, j_2)w'_2 v_2^k)$. Hence $\pi_0 \pi^{k+1}$ is nonsplitting. Also since $v_1 v_2 \neq \varepsilon$, we obtain that $|(d_i, j_i)w'_i v_i^k| > k$ for $i = 1$ or $i = 2$. Since this holds for every k, the set $\{\mathsf{diff}(\pi_0 \pi^{k+1}) \mid k \geq 1\}$ is infinite, contradicting Corollary 78. □

Example 80. Let M be the dtla of Example 72 and let $\pi_0 = e_{01}$ and $\pi = e_1$. Then π has output label $((\sigma_a, 1), (\sigma_b, 1))$ and $(\sigma_a, 1)(\sigma_b, 1) \neq \varepsilon$. The path $\pi_0 \pi$ has output label $((\tau, 1)(\sigma_a, 1), (\tau, 1)(\sigma_b, 1))$ and so $\mathsf{diff}(\pi_0 \pi) = ((\sigma_a, 1), (\sigma_b, 1))$, which means that $\pi_0 \pi$ is nonsplitting but not ancestral. Indeed, consider the pumped path $\pi_0 \pi^{k+1}$. By Example 74, it has label $(1^{k+2}, (\tau, 1)(\sigma_a, 1)^{k+1}, (\tau, 1)(\sigma_b, 1)^{k+1})$ and satisfies the

requirements of Lemma 73 for $C = \tau(\sigma^{k+1}\bot, a)$ and for $p_1 = p_a$ and $p_2 = p_b$. By the proof of Lemma 77, that implies that the node $\mathsf{nod}((\tau,1)) = 1$ is a difference node of $M(C[p_a])$ and $M(C[p_b])$, and that $M(C[p_a])/1 = (\sigma_a, 1)^{k+1}\langle q_{1a}, p_a\rangle$ is a difference tree of M. Hence $\mathsf{diff}(M)$ is infinite. $\qquad\square$

In the proof of the next, technical pumping lemma we use that M is ultralinear (with mapping $\mu : Q \to \mathbb{N}$) and b-erasing (with graph E_M). We note here the obvious fact that if e is an edge in G_M from (q_1, q_2, b) to (q_1', q_2', b'), then $\mu(q_i) \leq \mu(q_i')$; and hence the same holds for paths in G_M. Thus, if e is part of a cycle in G_M, then $\mu(q_i) = \mu(q_i')$. Also, if e has output label (v_1, v_2) and $v_i = \varepsilon$, then there is an edge from q_i to q_i' in E_M; and hence the same holds for paths (of the same length) in G_M and E_M. Thus, if a (q_1, q_2, b)-cycle in G_M has output label $(\bar{v}_1, \bar{v}_2)$, then both $\bar{v}_1$ and $\bar{v}_2$ are nonempty.

Lemma 81. *Let* $\mathsf{diff}(M)$ *be finite. Let* π_0 *be a* (q_1, q_2, b)-*path in* G_M *and* π *be a* (q_1, q_2, b)-*cycle in* G_M. *If* $\pi_0\pi$ *is nonsplitting, then* $\pi_0\pi^k$ *is nonsplitting for every* $k \geq 1$.

Proof. Clearly, it suffices to show that $\pi_0\pi^2$ is nonsplitting. Suppose that there exist π_0 and π such that $\pi_0\pi^2$ is splitting. Let $\pi = e_1 \cdots e_n$ with $n \geq 1$ and e_m an edge of G_M for $m \in [n]$. Without loss of generality, we may assume that $\pi_0\pi e_1$ is splitting. In fact, suppose that e_j is the first edge of π that causes splitting, i.e., $\pi_0\pi e_1 \cdots e_{j-1}$ is nonsplitting and $\pi_0\pi e_1 \cdots e_j$ is splitting. Let e_{j-1} lead to node $(\widetilde{q}_1, \widetilde{q}_2, b)$, and let $\widetilde{\pi}_0 = \pi_0 e_1 \cdots e_{j-1}$ and $\widetilde{\pi} = e_j \cdots e_n e_1 \cdots e_{j-1}$. Then $\widetilde{\pi}_0$ is a $(\widetilde{q}_1, \widetilde{q}_2, b)$-path, $\widetilde{\pi}$ is a $(\widetilde{q}_1, \widetilde{q}_2, b)$-cycle, $\widetilde{\pi}_0\widetilde{\pi}$ is nonsplitting, and $\widetilde{\pi}_0\widetilde{\pi}e_j$ is splitting.

So, suppose that $\pi_0\pi e$ is splitting, where $e = e_1$. Let $\pi_0\pi$ have label (u, v_1, v_2), let $\mathsf{diff}(\pi_0\pi) = \mathsf{diff}(v_1, v_2) = (w_1, w_2)$ and let π have output label $(\bar{v}_1, \bar{v}_2)$, where $\bar{v}_i$ is a postfix of v_i. Since M is b-erasing, $\bar{v}_1 \neq \varepsilon$ and $\bar{v}_2 \neq \varepsilon$ (because if $\bar{v}_i = \varepsilon$, then there is a cycle of length n from q_i to q_i in E_M). Consequently, $\pi_0\pi$ is ancestral by Lemma 79, i.e., $w_1 = \varepsilon$ or $w_2 = \varepsilon$. Let e end at the node (q_1', q_2', b) and let (r_1, r_2, j, z_1, z_2) be its label. Since $\pi_0\pi e$ is splitting, $\mathsf{diff}(\pi_0\pi e) = \mathsf{diff}(v_1 z_1, v_2 z_2) = \mathsf{diff}(w_1 z_1, w_2 z_2) = ((d, j_1)y_1, (d, j_2)y_2)$ with $j_1 \neq j_2$. Assume that $w_1 = \varepsilon$, i.e., $v_2 = v_1 w_2$ (the case where $w_2 = \varepsilon$ is analogous). So $\mathsf{diff}(z_1, w_2 z_2) = ((d, j_1)y_1, (d, j_2)y_2)$. Let y be the longest common prefix of z_1 and $w_2 z_2$, i.e., $z_1 = y(d, j_1)y_1$ and $w_2 z_2 = y(d, j_2)y_2$. Then $v_1 y$ is the longest common prefix of $v_1 z_1$ and $v_2 z_2$, because $v_1 z_1 = v_1 y(d, j_1)y_1$ and $v_2 z_2 = v_1 w_2 z_2 = v_1 y(d, j_2)y_2$. Now recall that z_1 is a branch of the right-hand side ζ_1 of r_1 such that $\zeta_1/z_1 = q_1'(x_j)$. Since M is ultralinear, we have $\mu(q_1) \leq \mu(q_1')$. We also have $\mu(q_1') \leq \mu(q_1)$, because $e_2 \cdots e_n$ is a path in G_M from (q_1', q_2', b) to (q_1, q_2, b). Hence $\mu(q_1) = \mu(q_1')$, and so, again by ultralinearity, there is no other occurrence of x_j in ζ_1. In particular, since $\zeta_1/y(d, j_1)y_1 = \zeta_1/z_1 = q_1'(x_j)$, the subtree $\zeta_1/y(d, j_2)$ does not contain x_j.

Let $\pi' = e_2 \cdots e_n$, so $\pi = e\pi'$. Let $k \in \mathbb{N}$ be such that $k > \mathsf{lubdiff}(M) + \mathsf{maxrhs}(M) + \mathsf{maxfix}(M)$, and let π_k be the pumped path $\pi_0\pi^{k+2} = \pi_0 e\pi'\pi^k$. By Lemma 73, applied to π_k, there exist $C \in \mathcal{C}_\Sigma$ and $p_1, p_2 \in P$ such that requirements (a), (b), and (c) of that lemma hold for the prefix $\pi_0\pi$ of π_k. By (a), $(q_1, u, \#) \in \mathsf{blink}_{C[p_1]}(v_1)$. Hence, since $\mathsf{rhs}(q_1, C[p_1], u) = \mathsf{rhs}(r_1) = \zeta_1$ by (c)(1), it follows from Lemma 69 and the definition of link that $(q_1, u, \mathsf{nod}(y)) \in \mathsf{link}_{C[p_1]}(\mathsf{nod}(v_1 y))$ and hence $\mathsf{or}_{C[p_1]}(\mathsf{nod}(v_1 y)) = (q_1, u, \mathsf{nod}(y))$. So, we obtain from Lemma 54(2) that $M(C[p_1])/v_1 y(d, j_2) = \zeta_1/y(d, j_2)[\bar{q}(x_{j'}) \leftarrow \bar{q}_M(C[p_1]/uj') \mid \bar{q} \in Q, j' \in \mathbb{N}_+]$.

For every $j' \in \mathbb{N}_+$ with $j' \neq j$ and $uj' \in V(C)$, the node uj' is not on the spine of C because, by (b), u and uj are both on or both off the spine. Hence, by (c)(2), $C[p_1]/uj' = C/uj' \in \mathsf{fix}(M)$ and so $\mathsf{ht}(\bar{q}_M(C[p_1]/uj')) \leq \mathsf{maxfix}(M)$. Since as observed above, $\zeta_1/y(d, j_2)$ does not contain x_j, we get that $\mathsf{ht}(M(C[p_1])/v_1 y(d, j_2)) \leq \mathsf{maxrhs}(M) + \mathsf{maxfix}(M) < k$.

Let π' have label (v_1', v_2'). Then $\mathsf{out}(\pi_k) = (v_1 z_1 v_1' \bar{v}_1^k, v_2 z_2 v_2' \bar{v}_2^k)$ and since (a) also holds for the path π_k itself, $v_2 z_2 v_2' \bar{v}_2^k$ is a branch of $M(C[p_2])$ by Lemma 69. Recall that $v_2 z_2 = v_1 w_2 z_2 = v_1 y(d, j_2) y_2$. Hence $y_2 v_2' \bar{v}_2^k$ is a branch of $M(C[p_2])/v_1 y(d, j_2)$. Since $\bar{v}_2 \neq \varepsilon$, $|y_2 v_2' \bar{v}_2^k| \geq k$ (in the case where $w_2 = \varepsilon$, we need here that $\bar{v}_1 \neq \varepsilon$). By Lemma 68 (with $v := v_1 y(d, j_2)$ and $w := y_2 v_2' \bar{v}_2^k$, and with p_1 and p_2 interchanged), there is a prefix w' of $y_2 v_2' \bar{v}_2^k$ such that $\mathsf{nod}(v_1 y(d, j_2) w')$ is a difference node of $M(C[p_1])$ and $M(C[p_2])$. Since $\mathsf{nod}(w')$ is a node of $M(C[p_1])/v_1 y(d, j_2)$, we have $|w'| \leq \mathsf{maxrhs}(M) + \mathsf{maxfix}(M)$. So $\mathsf{ht}(M(C[p_2])/v_1 y(d, j_2) w') \geq |y_2 v_2' \bar{v}_2^k| - |w'| \geq k - (\mathsf{maxrhs}(M) + \mathsf{maxfix}(M)) > \mathsf{lubdiff}(M)$, a contradiction. □

Example 82. Let M be the dtla of Example 72 and let $\pi_0 = e_{02}$ and $\pi = e_2$. The path $\pi_0 \pi$ has output label $((\tau, 2)(\tau, 1), (\tau, 2)(\tau, 1)(\tau, 2))$ and so $\mathsf{diff}(\pi_0 \pi) = (\varepsilon, (\tau, 2))$, which means that $\pi_0 \pi$ is ancestral. On the other hand, the path $\pi_0 \pi^2$ has output label $((\tau, 2)(\tau, 1)(\tau, 1), (\tau, 2)(\tau, 1)(\tau, 2)(\tau, 1)(\tau, 2))$ which implies that $\mathsf{diff}(\pi_0 \pi^2) = ((\tau, 1), (\tau, 2)(\tau, 1)(\tau, 2))$, and so $\pi_0 \pi^2$ is splitting. Indeed, consider the pumped path $\pi_k = \pi_0 \pi^{k+2}$, which has label $(2 \cdot 1^{k+2}, (\tau, 2)(\tau, 1)^{k+2}, (\tau, 2)((\tau, 1)(\tau, 2))^{k+2})$ and satisfies the requirements of Lemma 73 for $C = \tau(\bot, \sigma^{k+2} a)$ and for $p_1 = p_a$ and $p_2 = p_b$, by Example 74. Now $M(C[p_a]) = \tau(\langle q_{1a}, p_a \rangle, \tau(\tau(q_{2aM}(\sigma^k a), a), a))$ and $M(C[p_b]) = \tau(\langle q_{1a}, p_a \rangle, \tau(\tau(a, \tau(\tau(a, q_{2bM}(\sigma^k a)), a)), a))$. Consequently, the node $\mathsf{nod}((\tau, 2)(\tau, 1)(\tau, 2)) = (2, 1, 2)$ is a difference node of $M(C[p_a])$ and $M(C[p_b])$, with $M(C[p_a])/(2, 1, 2) = a$ and $M(C[p_b])/(2, 1, 2) = \tau(\tau(a, q_{2bM}(\sigma^k a)), a)$. So, $\tau(\tau(a, q_{2bM}(\sigma^k a)), a)$ is a difference tree of M and has a branch $((\tau, 1)(\tau, 2))^{k+1}$, for every k. Hence $\mathsf{diff}(M)$ is infinite, which, of course, we already knew from Example 80. □

Finally, we prove that $(2 \cdot |Q|^2 - 1) \cdot \mathsf{maxrhs}(M)$ is an ancestral bound for M.

Lemma 83. *Let $\mathsf{diff}(M)$ be finite. Let π be an ancestral entry path in G_M. If $\mathsf{diff}(\pi) = (w, \varepsilon)$, then $|w| \leq (2 \cdot |Q|^2 - 1) \cdot \mathsf{maxrhs}(M)$.*

Proof. Suppose that $|w| > (2 \cdot |Q|^2 - 1) \cdot \mathsf{maxrhs}(M)$. For a prefix π' of π, we will denote the first component of $\mathsf{diff}(\pi')$ by $\mathsf{diff}_1(\pi')$. So, $|\mathsf{diff}_1(\varepsilon)| = 0$ and $|\mathsf{diff}_1(\pi)| > (2 \cdot |Q|^2 - 1) \cdot \mathsf{maxrhs}(M)$. We also observe that, by definition of G_M, every edge e of π adds at most $\mathsf{maxrhs}(M)$ to $|\mathsf{diff}_1(\pi')|$; formally, if $\pi' e$ is a prefix of π, then $|\mathsf{diff}_1(\pi' e)| \leq |\mathsf{diff}_1(\pi')| + \mathsf{maxrhs}(M)$, because $\mathsf{diff}(\pi' e) = \mathsf{diff}(\mathsf{diff}(\pi') \cdot \mathsf{out}(e))$ and so $\mathsf{diff}_1(\pi' e)$ is a postfix of $\mathsf{diff}_1(\pi') z_1$ if $\mathsf{out}(e) = (z_1, z_2)$.

Let $\pi_0 = \varepsilon$ and for $1 \leq i \leq 2 \cdot |Q|^2$, let π_i be the shortest prefix of π such that $|\mathsf{diff}_1(\pi_i)| > |\mathsf{diff}_1(\pi_{i-1})|$. Obviously, π_{i-1} is a proper prefix of π_i. Moreover, by the above observation, $|\mathsf{diff}_1(\pi_i)| \leq |\mathsf{diff}_1(\pi_{i-1})| + \mathsf{maxrhs}(M)$ and hence $|\mathsf{diff}_1(\pi_i)| \leq i \cdot \mathsf{maxrhs}(M)$, which shows that π_i is well defined for every $i \leq 2 \cdot |Q|^2$. Note that π_i is ancestral and $\mathsf{diff}(\pi_i) = (w_i, \varepsilon)$ for some branch w_i. Since we have defined $2 \cdot |Q|^2 + 1$ prefixes of π, there exist π_i and π_j with $i < j$ that are (q_1, q_2, b)-paths for the same node

(q_1, q_2, b) of G_M. So, π_i is a prefix of π_j and $|w_j| > |w_i|$. Let $\bar{\pi}$ be the (q_1, q_2, b)-cycle such that $\pi_j = \pi_i \bar{\pi}$, and let $\mathsf{out}(\bar{\pi}) = (v_1, v_2)$. Then $\mathsf{diff}(\pi_j) = \mathsf{diff}(\mathsf{diff}(\pi_i) \cdot \mathsf{out}(\bar{\pi}))$ and so $(w_j, \varepsilon) = \mathsf{diff}(w_i v_1, v_2)$, i.e., $w_i v_1 = v_2 w_j$. Since $|w_i| < |w_j|$, we obtain that $|v_1| > |v_2|$. We now pump $\bar{\pi}$ and consider the path $\pi_i \bar{\pi}^k$ for every $k \geq 1$. Let $(w'_{1,k}, w'_{2,k}) = \mathsf{diff}(\pi_i \bar{\pi}^k) = \mathsf{diff}(\mathsf{diff}(\pi_i) \cdot \mathsf{out}(\bar{\pi}^k)) = \mathsf{diff}(w_i v_1^k, v_2^k)$. By Lemmas 81 and 79, $\pi_i \bar{\pi}^k$ is ancestral. Since $|w_i v_1^k| > |v_2^k|$, it follows that v_2^k is a prefix of $w_i v_1^k$, and so $w'_{2,k} = \varepsilon$ and $w_i v_1^k = v_2^k w'_{1,k}$. Hence $|w'_{1,k}| = |w_i v_1^k| - |v_2^k| = |w_i| + k(|v_1| - |v_2|) \geq k$. Since this holds for every $k \geq 1$, it contradicts Corollary 78. $\qquad \square$

Theorem 84. *The number* $(2 \cdot |Q|^2 - 1) \cdot \mathsf{maxrhs}(M)$ *is an ancestral bound for* M.

Proof. Immediate from Lemmas 76 and 83. $\qquad \square$

We now present our second main result.

Theorem 85. *It is decidable for a total ultralinear bounded erasing dtla M whether there exists a dtop N such that $\llbracket M \rrbracket = \llbracket N \rrbracket$, and if so, such a dtop N can be constructed.*

Proof. We use Corollary 51 for the class $\mathcal{U}$ of total ultralinear b-erasing dtlas. We first consider an initialized and la-uniform $M \in \mathcal{U}$. As observed in Section 8, an output bound $h_{\mathrm{o}}(M)$ for $M \in \mathcal{U}$ can be computed from M; in fact, by Theorem 67, $h_{\mathrm{o}}(M) = \mathsf{maxrhs}(M) \cdot |Q| \cdot (|P| + 2)$ is an output bound for M. By Theorem 84, $h_{\mathrm{a}}(M) = (2 \cdot |Q|^2 - 1) \cdot \mathsf{maxrhs}(M)$ is an ancestral bound for M (which, of course, can be computed from M). Finally, Theorem 65 shows that $2 \cdot \mathsf{maxrhs}(M) + h_{\mathrm{o}}(M) + h_{\mathrm{a}}(M) + 1 = 1 + \mathsf{maxrhs}(M) \cdot (|Q| \cdot (|P| + 2) + 2 \cdot |Q|^2 + 1)$ is a difference bound for M, and hence so is the larger number $1 + 2 \cdot \mathsf{maxrhs}(M) \cdot (|Q| + |P|)^2$. It now follows from Lemmas 11 and 13 that we obtain a difference bound for *every* $M \in \mathcal{U}$ by modifying this expression as follows: first change $\mathsf{maxrhs}(M)$ into $2 \cdot \mathsf{maxrhs}(M)$ and $|Q|$ into $(|Q| + 1) \cdot |P|$, and then take the maximum with $\mathsf{maxrhs}(M)$ (cf. the paragraphs after Lemmas 11 and 13). Thus, $\mathcal{U}$ has Property (H) of Corollary 51 with the computable mapping $h : \mathcal{U} \to \mathbb{N}$ such that $h(M) = 1 + 4 \cdot \mathsf{maxrhs}(M) \cdot (|Q| + 2)^2 \cdot |P|^2$. $\qquad \square$

The same result holds for a total output-monadic dtla M, where *output-monadic* means that $\mathsf{rk}(d) \leq 1$ for every $d \in \Delta$. The reason is that Lemma 81, which is the only result that needs the assumption that M is ultralinear and bounded erasing, is trivial if M is output-monadic, because in that case every entry path in G_M is nonsplitting. Note that every output-monadic dtla is (ultra)linear, but not necessarily bounded erasing. This result is a slight generalization of the result of [3] for string transducers (more precisely, subsequential functions).

Theorem 85 also holds for a total initialized depth-uniform dtla M, where *depth-uniform* means that for every $a \in \Sigma^{(k)}$ and every $j \in [k]$ there exists a number $d_{a,j} \in \mathbb{N}$ such that for every rule $q(a(x_1 : p_1, \ldots, x_k : p_k)) \to \zeta$ and every $z \in V(\zeta)$, if $\zeta/z = q'(x_j)$ then $|z| = d_{a,j}$. In other words, every translation of the jth child of a node with label a is at the same depth of the right-hand side of every rule for a. This property implies that $|z_1| = |z_2|$ for every label (r_1, r_2, j, z_1, z_2) of an edge of G_M, hence $|v_1| = |v_2|$ for every output label (v_1, v_2) of an entry path in G_M, and hence $\mathsf{diff}(\pi) = (\varepsilon, \varepsilon)$ for every ancestral entry path in G_M. Thus, $h_{\mathrm{a}}(M) = 0$ is an ancestral bound for M, by Lemma 76.

63

Example 86. Let $\Sigma = \{\sigma^{(1)}, \tau^{(1)}, a^{(0)}, b^{(0)}\}$ and $\Delta = \{\sigma_a^{(2)}, \sigma_b^{(2)}, a^{(0)}, b^{(0)}\}$. Consider the initialized la-uniform dtla M that translates every input tree s with n occurrences of σ into the full binary tree of height n over $\{\sigma_y, y\}$, where y is the label of the leaf of s. For instance, $M(\sigma\tau\sigma a) = \sigma_a(\sigma_a(a, a), \sigma_a(a, a))$. It has look-ahead states p_a and p_b, with the usual transitions, and states q_a and q_b. For $y \in \{a, b\}$, it has axioms $A(p_y) = q_y(x_0)$ and rules $q_y(\sigma(x_1 : p_y)) \to \sigma_y(q_y(x_1), q_y(x_1))$, $q_y(\tau(x_1 : p_y)) \to q_y(x_1)$, and $q_y(y) \to y$. Clearly M is depth-uniform, with $d_{\sigma,1} = 1$ and $d_{\tau,1} = 0$. Note that M is neither ultralinear nor b-erasing. $\qquad\square$

The depth-uniform property can be weakened by defining it as follows: there is a number $h'_{\mathrm{a}}(M) \in \mathbb{N}$ such that for every entry path π in G_M, if $\mathsf{out}(\pi) = (v_1, v_2)$ then $||v_1| - |v_2|| \le h'_{\mathrm{a}}(M)$, i.e., the distance between $|v_1|$ and $|v_2|$ is at most $h'_{\mathrm{a}}(M)$. Clearly, such a number $h'_{\mathrm{a}}(M)$ is an ancestral bound for M by Lemma 76. This weak depth-uniform property is decidable, and if it holds then a bound $h'_{\mathrm{a}}(M)$ can be computed. In fact, it is easy to see that M is weak depth-uniform if and only if $|v_1| = |v_2|$ for every output label (v_1, v_2) of a cycle in G_M. This can be checked by considering all simple cycles in G_M, i.e., cycles in which no node of G_M occurs more than once. The bound $h'_{\mathrm{a}}(M)$ can then be computed by considering all entry paths in G_M that do not contain a cycle.

10 Conclusion

We have tried to answer the following questions: is there a method to determine for a given top-down tree transducer *with regular look-ahead*, whether or not its translation can be realized by a top-down tree transducer *without regular look-ahead*? And if so, can such a transducer be constructed from the given one?

We have obtained some partial answers to this question for total dtlas (i.e., total deterministic top-down tree transducers with regular look-ahead). In Theorem 50 we have given a general algorithm to decide for a given total dtla M with a given difference bound, whether M is equivalent to a deterministic top-down tree transducer (without regular look-ahead), and if so, to construct such a dtop. In Sections 8 and 9 we have shown that a difference bound can be computed for total dtlas that are ultralinear and bounded erasing, or output-monadic, or initialized and depth-uniform.

Of course, we would like to prove that a difference bound can be computed for *all* total dtlas. To do this it suffices, as shown in Theorems 65 and 67, to prove that an ancestral bound can be computed for all total dtlas. We would also like to extend our results to the non-total case where a dtla realizes a partial function, and to the case where the dtla and the dtop are restricted to a given regular tree language. And 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.

We did not investigate the precise time complexity of our algorithm. Consider a total ultralinear and bounded erasing dtla M. We have shown in the proof of Theorem 85 that M has a difference bound $h(M)$ that is polynomial in the size of M. Since $\mathsf{sumfix}(M)$ can be taken exponential in the size of M, the canonical dtla $\mathsf{can}(M)$ can be built in exponential time and has an exponential difference bound $h(\mathsf{can}(M))$, see Theorem 50.

Clearly, the running time of Algorithm 44 on input $\mathsf{can}(M)$ is at most the number of tuples of trees of height $\leq h(\mathsf{can}(M))$ in $\mathcal{T}_\Delta(Q \times P)^n$ (with $n = |P|$), which is double exponential in $h(\mathsf{can}(M))$. Thus, the time complexity of our algorithm is at most triple exponential in the size of M. On the other hand, it is at least exponential in the size of M, because the size of a dtop equivalent to M can be exponential in the size of M. To see this, consider Example 8 and, for every $n \in \mathbb{N}$, let M_n be the dtla M obtained from M by changing 80 into n, and let N_n be the dtop obtained from N by changing 50 into $n - 30$. Clearly, on input M_n our algorithm computes the dtop N_n. The size of M_n is polynomial in n, whereas, due to the states q_z of N_n with $z \in \{\sigma, \tau\}^*$ and $|z| \leq n - 30$, the size of N_n is exponential in n. Thus, our algoritm takes exponential time on input M_n. We do not know the precise time complexity of our algorithm. Also, we do not know the precise upper bound on the size of a dtop N equivalent to M (if such an N exists).

We have shown in Corollary 28 and Lemma 17 that, for M to be equivalent to a dtop, $\mathsf{diftup}(M)$ must be finite. But we do not know whether it is decidable if $\mathsf{diftup}(M)$ is finite. If so, and if, moreover, we could compute $\mathsf{diftup}(M)$ from M, then Algorithm 44 could be simplified: we could just (try to) compute the unique dtop N with $Q_N = \mathsf{diftup}(M)$ that is associated with M (see Lemma 26). Thus, we would like to get more grip on the difference tuples of M. We do not even know whether $\mathsf{diftup}(M)$ is a decidable subset of $\mathcal{T}_\Delta(Q \times P)^n$.

As observed at the end of the Introduction, regular look-ahead can always be removed from a macro tree transducer [12]. Hence another more general question is: Is it decidable for a given macro tree transducer whether it is equivalent to a top-down tree transducer? For total deterministic transducers, this problem can be reduced to our problem by first proving the decidability of the following related question: Is it decidable for a given macro tree transducer whether it is equivalent to a top-down tree transducer with regular look-ahead? This is an open problem of which it was recently shown in [14] that the answer is 'yes' when the transducers have origin-semantics, which means that they should associate the same origins in the input tree to the nodes of the output tree.

As observed by one of the referees, a problem related to our problem is whether it is decidable if a given *nondeterministic* top-down tree transducer is equivalent to a *deterministic* one (both without regular look-ahead). Since it is decidable whether a nondeterministic top-down tree transducer is functional (i.e., realizes a function), see [13], we may assume that the given nondeterministic transducer is functional. Hence, it can be transformed into a deterministic top-down tree transducer with regular look-ahead by the result of [7]. This transformation clearly preserves the properties of ultralinearity and bounded erasing (appropriately defined for nondeterministic transducers). Thus, it is decidable for a given total ultralinear and b-erasing nondeterministic top-down tree transducer whether it is equivalent to a deterministic top-down tree transducer. This also holds for (appropriately defined) output-monadic, or initialized and depth-uniform nondeterministic top-down tree transducers. As for dtlas, we do not know whether it holds in the general case. We note that the class of functional nondeterministic top-down tree transductions lies properly between the class of deterministic top-down tree transductions and the class of deterministic top-down tree transductions with regular look-ahead. For instance, the dtla M_{ex} from the Introduction is equivalent to a nonde-

terministic top-down tree transducer if $\Sigma = \Delta = \{\sigma^{(1)}, a^{(0)}, b^{(0)}\}$ (as in Example 7), but not if $\Sigma = \Delta = \{\sigma^{(2)}, a^{(0)}, b^{(0)}\}$.

References

1. R. Alur and L. D'Antoni. Streaming tree transducers. In A. Czumaj, K. Mehlhorn, A.M. Pitts, and R. Wattenhofer, editors, *ICALP 2012, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 42–53. Springer, 2012.
2. J. Berstel. *Transductions and Context-Free Languages*. Teubner-Verlag, 1979.
3. C. Choffrut. Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theor. Comput. Sci.*, 5(3):325–337, 1977.
4. F. Drewes. *Grammatical Picture Generation – A Tree-Based Approach*. Springer, 2006.
5. F. Drewes and J. Engelfriet. Decidability of the finiteness of ranges of tree transductions. *Inf. Comput.*, 145(1):1–50, 1998.
6. J. Engelfriet. Top-down tree transducers with regular look-ahead. *Mathematical Systems Theory*, 10:289–303, 1977.
7. J. Engelfriet. On tree transducers for partial functions. *Inf. Process. Lett.*, 7(4):170–172, 1978.
8. J. Engelfriet and S. Maneth. Macro tree transducers, attribute grammars, and MSO definable tree translations. *Inf. Comput.*, 154(1):34–91, 1999.
9. 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.
10. J. Engelfriet, S. Maneth, and H. Seidl. How to remove the look-ahead of top-down tree transducers. In A. M. Shur and M. V. Volkov, editors, *DLT 2014*, volume 8633 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 2014.
11. J. Engelfriet, G. Rozenberg, and G. Slutzki. Tree transducers, L systems, and two-way machines. *J. Comput. Syst. Sci.*, 20(2):150–202, 1980.
12. J. Engelfriet and H. Vogler. Macro tree transducers. *J. Comput. Syst. Sci.*, 31(1):71–146, 1985.
13. Z. Ésik. Decidability results concerning tree transducers I. *Acta Cybern.*, 5:1–20, 1980.
14. E. Filiot, S. Maneth, P.-A. Reynier, and J.-M. Talbot. Decision problems of tree transducers with origin. In M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, editors, *ICALP 2015, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 209–221. Springer, 2015.
15. E. Filiot and F. Servais. Visibly pushdown transducers with look-ahead. In M. Bieliková, G. Friedrich, G. Gottlob, S. Katzenbeisser, and G. Turán, editors, *SOFSEM 2012*, volume 7147 of *Lecture Notes in Computer Science*, pages 251–263. Springer, 2012.
16. Z. Fülöp, A. Kühnemann, and H. Vogler. A bottom-up characterization of deterministic top-down tree transducers with regular look-ahead. *Inf. Process. Lett.*, 91(2):57–67, 2004.
17. S. Ginsburg and E. H. Spanier. Finite-turn pushdown automata. *SIAM J. Control*, 4:429–453, 1966.
18. H. Hosoya. *Foundations of XML Processing – The Tree-Automata Approach*. Cambridge University Press, 2010.
19. K. Knight and J. Graehl. An overview of probabilistic tree transducers for natural language processing. In A. F. Gelbukh, editor, *CICLing 2005*, volume 3406 of *Lecture Notes in Computer Science*, pages 1–24. Springer, 2005.

20. A. Lemay, S. Maneth, and J. Niehren. A learning algorithm for top-down XML transfor-mations. In J. Paredaens and D. Van Gucht, editors, *PODS 2010*, pages 285–296. ACM, 2010.

21. A. Maletti. Tree transformations and dependencies. In M. Kanazawa, A. Kornai, M. Kracht, and H. Seki, editors, *MOL 12*, volume 6878 of *Lecture Notes in Artificial Intelligence*, pages 1–20. Springer, 2011.

22. A. Maletti, J. Graehl, M. Hopkins, and K. Knight. The power of extended top-down tree transducers. *SIAM J. Comput.*, 39(2):410–430, 2009.

23. S. Maneth and F. Neven. Structured document transformations based on XSL. In R. C. H. Connor and A. O. Mendelzon, editors, *DBPL'99*, volume 1949 of *Lecture Notes in Computer Science*, pages 80–98. Springer, 1999.

24. W. Martens and F. Neven. Frontiers of tractability for typechecking simple XML transfor-mations. *J. Comput. Syst. Sci.*, 73(3):362–390, 2007.

25. W. Martens, F. Neven, and M. Gyssens. Typechecking top-down XML transformations: Fixed input or output schemas. *Inf. Comput.*, 206(7):806–827, 2008.

26. W. C. Rounds. Mappings and grammars on trees. *Mathematical Systems Theory*, 4(3):257–287, 1970.

27. J. W. Thatcher. Generalized sequential machine maps. *J. Comput. Syst. Sci.*, 4(4):339–367, 1970.

28. A. van Deursen, P. Klint, and F. Tip. Origin tracking and its applications. In *Language Prototyping – An Algebraic Specification Approach*, pages 249–294. World Scientific, 1996.