Synthesis of Deterministic Top-down Tree Transducers from Automatic Tree Relations*

Christof Löding Sarah Winter

RWTH Aachen University, Germany

Abstract

We consider the synthesis of deterministic tree transducers from automaton definable specifications, given as binary relations, over finite trees. We consider the case of tree automatic specifications, meaning the specification is recognizable by a top-down tree automaton that reads the two given trees synchronously in parallel. In this setting we study tree transducers that are allowed to have either delay that remains in a given bound or arbitrary delay. Delay is caused whenever the transducer reads a symbol from the input tree but does not produce output. For specifications that are deterministic top-down tree automatic, we provide decision procedures for both bounded and arbitrary delay that yield deterministic top-down tree transducers which realize the specification for valid input trees, that is, realize the specification for trees that are part of the specification domain, and can behave arbitrarily on trees outside the domain. Similar to the case of relations over words, we use two-player games as the main technique to obtain our results.

1 Introduction

The synthesis problem asks, given a specification that relates possible inputs to allowed outputs, whether there is a program realizing the specification, and if so, construct one. This problem setting originates from Church's synthesis problem [1] which was already posed 1957 in [2]. Church considers the case where the input is an infinite bit sequence that has to be transformed, bit by bit, into an infinite bit sequence. The synthesis problem is then to decide whether there is a circuit which realizes the given input/output specification, and construct one if possible. A different terminology for the same problem uses the notion of uniformization of a (binary) relation, which is a function that selects for each element of the domain of the relation an element in its image. The synthesis problem asks for effective uniformization by functions that can be implemented in a specific way.

*This work was supported by the DFG grant (LO 1174/3-1).

Specifications are usually written in some logical formalism, while the uniformization, in particular in the synthesis setting, is required to be implemented by some kind of device. Since many logics can be translated into automata, which can also serve as implementations of a uniformization, it is natural to study uniformization problems in automata theory. Relations (or specifications) can be defined using automata with two input tapes, and uniformizations can be realized by transducers, that is, automata with output.

A first uniformization result in such a setting has been obtained by Büchi and Landweber in [3], who showed that for specifications over infinite words in monadic second-order logic, it is decidable whether they have a uniformization by a synchronous transducer (that outputs one symbol for each input letter). The specifications considered in [3] can be translated into finite automata that read the two input words synchronously. Relations that can be defined by synchronous automata are referred to as automatic relations over finite words, and as ω-automatic relations over infinite words.

The result of Büchi and Landweber has been extended to transducers with delay, that is, transducers that have the possibility to produce empty output in some transitions. Delay in the transducers can, e.g., be useful if parts of the input have to be deleted. For a bounded delay, decidability was shown in [4], and for an unbounded delay in [5]. In the case of finite words, it was shown in [6] that it is decidable whether an automatic relation has a uniformization by a deterministic subsequential transducer, that is, a transducer that can output finite words on each transition.

Our aim is to study these uniformization questions for relations over trees. Tree automata are used in many fields, for example as a tool for analyzing and manipulating rewrite systems or XML Schema languages (see [7]). Tree transformations that are realized by finite tree transducers thus become interesting in the setting of translations from one document scheme into another [8]. There are already some uniformization results for tree relations. For example, in [9] it is shown that each relation that can be defined by a nondeterministic top-down tree transducer, has a uniformization by a deterministic top-down tree transducer with regular lookahead (in which transitions can test regular properties of the current subtree). However, these results focus on the existence of a uniformization for each relation that can be expressed by the considered model for the specifications. In contrast to that, we are interested in the corresponding decision problem. More precisely, for a class $\mathcal{C}$ of tree relations and a class $\mathcal{F}$ of functions over trees, we are interested in a procedure that decides whether a given relation from $\mathcal{C}$ has a uniformization in $\mathcal{F}$. Note that this decision problem only makes sense if not every relation in $\mathcal{C}$ has a uniformization in $\mathcal{F}$.

In this paper, we start the investigation of such questions in the rich landscape of tree automaton and tree transducer models. We study uniformization of automatic tree relations over finite trees (corresponding to the class $\mathcal{C}$) by deterministic top-down tree transducers (corresponding to the class $\mathcal{F}$). For automatic tree relations that can be defined by deterministic top-down automata (that is, an automaton deterministically reads the two input trees synchronously in top-down fashion), we show that it is decidable whether a given relation has a uniformization by a top-down tree transducer, and if possible construct one. For this result, we require that the transducer for the uniformization works correctly on all input trees that are in the domain of the specification. For trees outside the domain, the transducer can produce an arbitrary output. Since the transducer does not have to validate that the input is in the domain of the specification, we refer to this setting as uniformization without input validation. We also briefly comment on the problems in the case of uniformization with input validation, and solve the problem for the restricted class of uniformizers without delay (synchronously producing one output symbol for each input symbol). These decidability results are obtained by constructing strategies in two-player games of infinite duration.

The paper is structured as follows. First, we fix some basic definitions and terminology. Then, in Section 3.1 and Section 3.2, we consider uniformization of deterministic top-down automatic tree relations by top-down tree transducers without input validation that have bounded delay and unbounded delay, respectively. In Section 3.3, we briefly consider the case of uniformization with input validation.

A preliminary version of this work has appeared in [10].

2 Preliminaries

The set of natural numbers containing zero is denoted by $\mathbb{N}$. For a set S, the powerset of S is denoted by 2^S. An *alphabet* Σ is a finite non-empty set of letters. A finite *word* is a finite sequence of letters. The set of all finite words over Σ is denoted by Σ^*. The length of a word $w \in \Sigma^*$ is denoted by $|w|$, the empty word is denoted by ε. For $w = a_1 \dots a_n \in \Sigma^*$ for some $n \in \mathbb{N}$ and $a_1, \dots, a_n \in \Sigma$, let $w[i]$ denote the ith letter of w, i.e., $w[i] = a_i$. Furthermore, let $w[i,j]$ denote the infix from the ith to the jth letter of w, i.e., $w[i,j] = a_i \dots a_j$. We write $u \sqsubseteq w$ if there is some v such that $w = uv$ for $u, v \in \Sigma^*$. A subset $L \subseteq \Sigma^*$ is called *language* over Σ.

A *ranked alphabet* Σ is an alphabet where each letter $f \in \Sigma$ has a finite set of ranks $rk(f) \subseteq \mathbb{N}$ (one could also work with a single rank for each symbol but we adhere to the standard of assigning a set of ranks to each symbol). The set of letters of rank i is denoted by Σ_i. A tree domain dom is a non-empty finite subset of $(\mathbb{N} \setminus \{0\})^*$ such that dom is prefix-closed and for each $u \in (\mathbb{N} \setminus \{0\})^*$ and $i \in \mathbb{N} \setminus \{0\}$ if $ui \in dom$, then $uj \in dom$ for all $1 \leq j < i$. We speak of ui as successor of u for each $u \in dom$ and $i \in \mathbb{N} \setminus \{0\}$.

A (finite Σ-labeled) *tree* is a pair $t = (dom_t, val_t)$ with a mapping $val_t : dom_t \to \Sigma$ such that for each node $u \in dom_t$ the number of successors of u is a rank of $val_t(u)$. The height h of a tree t is the length of its longest path, i.e., $h(t) = max\{|u| \mid u \in dom_t\}$. The set of all Σ-labeled trees is denoted by T_Σ. A subset $T \subseteq T_\Sigma$ is called *tree language* over Σ.

A *subtree* $t|_u$ of a tree t at node u is defined by $dom_{t|_u} = \{v \in \mathbb{N}^* \mid uv \in dom_t\}$ and $val_{t|_u}(v) = val_t(uv)$ for all $v \in dom_{t|_u}$. In order to formalize concatenation of trees, we introduce the notion of special trees. A *special tree* over Σ is a tree over $\Sigma \uplus \{\circ\}$ such that $\circ$ occurs exactly once at a leaf. Given $t \in T_\Sigma$ and $u \in dom_t$, we write $t[\circ/u]$ for the special tree that is obtained by deleting the subtree at u and replacing it by $\circ$. Let S_Σ be the set of special trees over Σ. For $t \in S_\Sigma$ and $s \in T_\Sigma$ or $s \in S_\Sigma$ let the *concatenation* $t \cdot s$ be the tree that is obtained from t by replacing $\circ$ with s.

For some of the proofs, it is convenient to concatenate a special tree with a single letter, instead of a tree. For $t \in S_\Sigma$ and $f \in \Sigma_i, i \geq 0$, let $t \cdot f$ denote the result of replacing $\circ$ with f in t. Formally, if $i > 0$, then the result is not a tree over Σ, but we treat it as such.

Let X_n be a set of n variables $\{x_1, \dots, x_n\}$ and Σ be a ranked alphabet. We denote by $T_\Sigma(X_n)$ the set of all trees over Σ which additionally can have variables from X_n at their leaves. We define X_0 to be the empty set, the set $T_\Sigma(\emptyset)$ is equal to T_Σ. Let $X = \bigcup_{n>0} X_n$. A tree from $T_\Sigma(X)$ is called *linear* if each variable occurs at most once. For $t \in T_\Sigma(X_n)$ let $t[x_1 \leftarrow t_1, \dots, x_n \leftarrow t_n]$ be the tree that is obtained by substituting each occurrence of $x_i \in X_n$ by $t_i \in T_\Sigma(X)$ for every $1 \leq i \leq n$.

A tree from $T_\Sigma(X_n)$ such that all variables from X_n occur exactly once and in the order $x_1, \dots, x_n$ when reading the leaf nodes from left to right, is called *n-context* over Σ. Given an n-context, the node labeled by x_i is referred to as ith hole for every $1 \leq i \leq n$. A special tree can be seen as a 1-context, a tree without variables can be seen a 0-context. If C is an n-context and $t_1, \dots, t_n \in T_\Sigma(X)$ we write $C[t_1, \dots, t_n]$ instead of $C[x_1 \leftarrow t_1, \dots, x_n \leftarrow t_n]$.

Tree automata. Tree automata can be viewed as a straightforward generalization of finite automata on finite words, when words are interpreted as trees over unary symbols. For a detailed introduction to tree automata see e.g. [11] or [7].

Let $\Sigma = \bigcup_{i=0}^m \Sigma_i$ be a ranked alphabet. A *non-deterministic top-down tree automaton* (an N↓TA) over Σ is of the form $\mathcal{A} = (Q, \Sigma, Q_0, \Delta)$ consisting of a finite set of states Q, a set $Q_0 \subseteq Q$ of initial states, and $\Delta \subseteq \bigcup_{i=0}^m (Q \times \Sigma_i \times Q^i)$ is the transition relation. For $i = 0$, we identify $Q \times \Sigma_i \times Q^i$ with $Q \times \Sigma_0$.

Let t be a tree and $\mathcal{A}$ be an N↓TA, a *run* of $\mathcal{A}$ on t is a mapping $\rho : dom_t \to Q$ compatible with Δ, i.e., $\rho(\varepsilon) \in Q_0$ and for each node $u \in dom_t$ with $i \geq 0$ successors $(\rho(u), val_t(u), \rho(u1), \dots, \rho(ui)) \in \Delta$. A tree $t \in T_\Sigma$ is *accepted* if, and only if, there is a run of $\mathcal{A}$ on t. The tree language *recognized* by $\mathcal{A}$ is $T(\mathcal{A}) = \{t \in T_\Sigma \mid \mathcal{A} \text{ accepts } t\}$.

A tree language $T \subseteq T_\Sigma$ is called *regular* if T is recognizable by a non-deterministic top-down tree automaton. Just as the class of regular word languages is closed under Boolean operations, so too is the class of regular tree languages.

A top-down tree automaton $\mathcal{A} = (Q, \Sigma, Q_0, \Delta)$ is *deterministic* (a D↓TA) if the set Q_0 is a singleton set and for each $f \in \Sigma_i$ and each $q \in Q$ there is at most one transition $(q, f, q_1, \dots, q_i) \in \Delta$. However, non-deterministic and deterministic top-down automata are not equally expressive.

An extension to regular tree languages are (binary) *tree-automatic relations*. A way for a tree automaton to read a tuple of finite trees is to use a ranked vector alphabet. Thereby, all trees are read in parallel, processing one node from each tree in a computation step. Hence, the trees are required to have the same domain. Therefore we use a padding symbol to extend the trees if necessary. Formally, this is done in the following way.

Let Σ, Γ be ranked alphabets and let $\Sigma_\perp = \Sigma \cup \{\perp\}$, $\Gamma_\perp = \Gamma \cup \{\perp\}$, where $\perp$ is a new symbol with rank 0. For an i-ary symbol $f \in \Sigma_\perp$ and a j-ary symbol $g \in \Gamma_\perp$, let $rk((f,g)) = max\{i,j\}$. The *convolution* of (t_1,t_2) with $t_1 \in T_\Sigma$, $t_2 \in T_\Gamma$ is the $\Sigma_\perp \times \Gamma_\perp$-labeled tree $t = t_1 \otimes t_2$ defined by $dom_t = dom_{t_1} \cup dom_{t_2}$, and $val_t(u) = (val^\perp_{t_1}(u), val^\perp_{t_2}(u))$ for all $u \in dom_t$, where $val^\perp_{t_i}(u) = val_{t_i}(u)$ if $u \in dom_{t_i}$ and $val^\perp_{t_i}(u) = \perp$ otherwise for $i \in \{1,2\}$. As a special case, given $t \in T_\Sigma$, we define $t \otimes \perp$ to be the tree with $dom_{t\otimes\perp} = dom_t$ and $val_{t\otimes\perp}(u) = (val_t(u), \perp)$ for all $u \in dom_t$. Analogously, we define $\perp \otimes t$. We define the *convolution of a tree relation* $R \subseteq T_\Sigma \times T_\Gamma$ to be the tree language $T_R := \{t_1 \otimes t_2 \mid (t_1,t_2) \in R\}$.

We call a (binary) relation R *tree-automatic* if there exists a regular tree language T such that $T = T_R$. For ease of presentation, we say a tree automaton $\mathcal{A}$ recognizes R if it recognizes the convolution T_R and denote by $R(\mathcal{A})$ the induced relation R.

A *uniformization* of a relation $R \subseteq X \times Y$ is a function $f_R : X \to Y$ such that for each domain element x the pair $(x, f_R(x))$ is in the relation, i.e., $(x, f_R(x)) \in R$ for all $x \in dom(R)$. In the following, we are interested whether for a given tree-automatic relation there exists a uniformization which can be realized by a tree transducer.

Tree Transducers. Tree transducers are a generalization of word transducers. As top-down tree automata, a top-down tree transducer reads the tree from the root to the leaves, but can additionally in each computation step produce finite output trees which are attached to the already produced output. For an introduction to tree transducers the reader is referred to [7].

A *top-down tree transducer* (a ↓TT) is of the form $\mathcal{T} = (Q, \Sigma, \Gamma, q_0, \Delta)$ consisting of a finite set of states Q, a finite input alphabet Σ, a finite output alphabet Γ, an initial state $q_0 \in Q$, and Δ is a finite set of transition rules of the form

$$q(f(x_1,\dots,x_i)) \to w[q_1(x_{j_1}),\dots,q_n(x_{j_n})],$$

where $f \in \Sigma_i$, w is an n-context over Γ, $q, q_1, \dots, q_n \in Q$ and variables $x_{j_1},\dots,x_{j_n} \in X_i$, or

$$q(x_1) \to w[q_1(x_1),\dots,q_n(x_1)] \quad (\varepsilon\text{-transition}),$$

where $f \in \Sigma_i$, w is an n-context over Γ, and $q, q_1, \dots, q_n \in Q$. A top-down tree transducer is *deterministic* (a D↓TT) if it contains no ε-transitions and there are no two rules with the same left-hand side. A top-down tree transducer is *linear* if all the trees in the transitions are linear.

A *configuration* of a top-down tree transducer is a triple $c = (t, t', \varphi)$ of an input tree $t \in T_\Sigma$, an output tree $t' \in T_{\Gamma\cup Q}$ and a function $\varphi : D_{t'} \to dom_t$, where

- $val_{t'}(u) \in \Gamma_i$ for each $u \in dom_{t'}$ with $i > 0$ successors
- $val_{t'}(u) \in \Gamma_0$ or $val_{t'}(u) \in Q$ for each leaf $u \in dom_{t'}$
- $D_{t'} \subseteq dom_{t'}$ with $D_{t'} = \{u \in dom_{t'} \mid val_{t'}(u) \in Q\}$
  (φ maps every node from the output tree t' that has a state-label to a node of the input tree t)

Let $c_1 = (t, t_1, \varphi_1)$ and $c_2 = (t, t_2, \varphi_2)$ be configurations of a top-down tree transducer over the same input tree. We define a *successor relation* $\rightarrow_{\mathcal{T}}$ on configurations as usual by applying one rule. Figure 1 illustrates a configuration sequence explained in Example 1 below. Formally, for the application of a non-ε-rule, this looks as follows:

$c_1 \rightarrow_{\mathcal{T}} c_2 \quad :\Leftrightarrow$ There is a state-labeled node $u \in D_{t'}$ of the output tree t_1 that is mapped to a node $v \in dom_t$ of the input tree t, i.e., $\varphi_1(u) = v$, and there is a rule $val_{t_1}(u)\,(val_t(v)(x_1, \ldots, x_i)) \rightarrow w[q_1(x_{j_1}), \ldots, q_n(x_{j_n})] \in \Delta$ such that the output tree is correctly updated, i.e., $t_2 = t_1[\circ/u] \cdot w[q_1, \ldots, q_n]$, and the mapping of state-labeled output nodes to input nodes is also correctly updated, i.e., $\varphi_2(u') = \varphi_1(u')$ if $u' \in D_{t_1} \setminus \{u\}$ and $\varphi_2(u') = v.j_i$ if $u' = u.u_i$ with u_i is the ith hole in w.

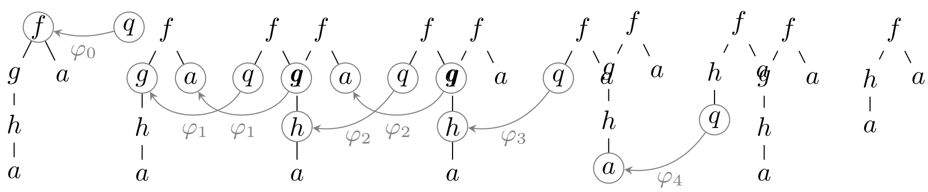
Furthermore, let $\rightarrow_{\mathcal{T}}^*$ be the reflexive and transitive closure of $\rightarrow_{\mathcal{T}}$ and $\rightarrow_{\mathcal{T}}^n$ the reachability relation for $\rightarrow_{\mathcal{T}}$ in n steps. From here on, let φ_0 always denote the mapping $\varphi_0(\varepsilon) = \varepsilon$. A configuration (t, q_0, φ_0) is called *initial configuration* of $\mathcal{T}$ on t. A configuration $c = (t, t', \varphi)$ is said to be *reachable* in a computation of $\mathcal{T}$ on t, if $c_0 \rightarrow_{\mathcal{T}}^* c$, where c_0 is the initial configuration of $\mathcal{T}$ on t. The relation $R(\mathcal{T})$ induced by a top-down tree transducer $\mathcal{T}$ is

$$R(\mathcal{T}) = \{(t, t') \in T_\Sigma \times T_\Gamma \mid (t, q_0, \varphi_0) \rightarrow_{\mathcal{T}}^* (t, t', \varphi)\}.$$

For a (special) tree $t \in T_\Sigma$ or $t \in S_\Sigma$ let $\mathcal{T}(t) \subseteq T_{\Gamma \cup Q}$ be the set of *final transformed outputs* of a computation of $\mathcal{T}$ on t, that is the set $\{t' \mid (t, q_0, \varphi_0) \rightarrow_{\mathcal{T}}^* (t, t', \varphi)$ s.t. there is no successor configuration of $(t, t', \varphi)\}$. Note, we explicitly do not require that the final transformed output is a tree over Γ. In the special case that $\mathcal{T}(t)$ is a singleton set $\{t'\}$, we also write $\mathcal{T}(t) = t'$. The class of relations definable by $\downarrow$TTs is called the class of *top-down tree transformations*.

Example 1 Let Σ be a ranked alphabet given by $\Sigma_2 = \{f\}$, $\Sigma_1 = \{g, h\}$, and $\Sigma_0 = \{a\}$. Consider the $\downarrow$TT $\mathcal{T}$ given by $(\{q\}, \Sigma, \Sigma, \{q\}, \Delta)$ with $\Delta = \{\, q(a) \rightarrow a,$ $q(g(x_1)) \rightarrow q(x_1),\ q(h(x_1)) \rightarrow h(q(x_1)),\ q(f(x_1, x_2)) \rightarrow f(q(x_1), q(x_2))\ \}$. For each $t \in T_\Sigma$ the transducer deletes all occurrences of g in t.

Consider $t := f(g(h(a)), a)$. A possible sequence of configurations of $\mathcal{T}$ on t is $c_0 \rightarrow_{\mathcal{T}}^5 c_5$ such that $c_0 := (t, q, \varphi_0)$ with $\varphi_0(\varepsilon) = \varepsilon$, $c_1 := (t, f(q, q), \varphi_1)$ with $\varphi_1(1) = 1$, $\varphi_1(2) = 2$, $c_2 := (t, f(q, q), \varphi_2)$ with $\varphi_2(1) = 11$, $\varphi_2(2) = 2$, $c_3 := (t, f(q, a), \varphi_3)$ with $\varphi_3(1) = 11$, $c_4 := (t, f(h(q), a), \varphi_4)$ with $\varphi_4(11) = 111$,



(a) c_0 (b) c_1 (c) c_2 (d) c_3 (e) c_4 (f) c_5

Figure 1: The configuration sequence c_0 to c_5 of $\mathcal{T}$ on t from Example 1.

and $c_5 := (t, f(h(a), a), \varphi_5)$. A visualization of this sequence is shown in Figure 1. ◁

In Sections 3, we consider restricted types of top-down tree transducers, namely *transducers with bounded (output) delay.* Intuitively, delay occurs in a computation of a transducer if there is a difference between the number of produced output symbols and read input symbols. If the output is behind this is called output delay. More formally, a configuration (t, t', φ) has *delay* d w.r.t. a node $u \in D_{t'}$ if the absolute value of $|\varphi(u)| - |u|$ equals d. We speak of *output delay* if $|\varphi(u)| - |u|$ is a positive integer. The maximum delay that of (t, t', φ) is defined as $max\{d \mid \exists u \in D_{t'} \text{ and } abs(|\varphi(u)| - |u|) = d\}$. We say the *delay* (resp. *output delay*) in a ↓TT $\mathcal{T}$ is *bounded* by k if for every reachable configuration c of $\mathcal{T}$ the maximum delay (resp. output delay) of c is at most k. We speak of ↓TTs *without delay*, if the delay is bounded by 0.

Example 2 Consider $\mathcal{T}$ from Example 1 and the configuration sequence of $\mathcal{T}$ given in Example 1. In c_2 occurs output delay 1 resp. 0 w.r.t. node 1 resp. 2 of the output tree. It is easy to see that the transducer has unbounded output delay, because it deletes all occurrences of g in an input tree. ◁

Games. A *safety game* $\mathcal{G} = (V, V_0, V_1, E, S)$ is played by two players, Player 0 and Player 1, on a directed game graph $G = (V, E)$, where

- $V = V_0 \uplus V_1$ is a partition of the vertices into positions V_0 belonging to Player 0 and positions V_1 belonging to Player 1,
- $E \subseteq V \times V$ is the set of allowed moves, and
- $S \subseteq V$ is a set of safe vertices.

Let $T \subseteq V$ denote the set of all terminal vertices, i.e., vertices without outgoing edges. A *play* is a finite or infinite sequence $v_0 v_1 v_2 \dots$ of vertices compatible to the edges of the game graph starting from an initial vertex $v_0 \in V$. A play is *maximal* if it is either infinite or it ends in a terminal vertex. Player 0 wins a play if it stays inside the safe region, i.e., $v_i \in S$ for all i.

Let $i \in \{0, 1\}$, a *strategy* for Player i is a function $\sigma_i : V^*(V_i \setminus T) \to V$ such that $\sigma_i(v_0 \dots v_n) = v_{n+1}$ implies that $(v_n, v_{n+1}) \in E$. A strategy σ_i is a *winning* *strategy* from a vertex $v_0 \in V$ for Player i if the player wins every play starting in v_0, no matter how the opponent plays, if Player i plays according σ_i.

Safety games are *positionally determined*, cf. [12], i.e., for each vertex $v \in V$ one of the players has a winning strategy from v. Furthermore, the player always has a positional winning strategy σ, meaning that the strategy does not consider the previously seen vertices, but only the current vertex. More formally, a positional strategy σ_i for Player i can be represented by a mapping $\sigma_i : V_i \setminus T \to V$ such that $(v, \sigma_i(v)) \in E$ for all $v \in V_i$.

3 Deterministic Top-down Specifications

Here we investigate uniformization of tree-automatic relations in the class of top-down tree transformations. We restrict ourselves in the scope of Section 3.1 and 3.2 to D↓TA-recognizable relations with total domain. Later on, in Section 3.3, we will briefly describe how we can deal with D↓TA-recognizable relations whose domain is not total but D↓TA-recognizable. In the course of this, we will consider two variants of uniformization. The classical uniformization setting, where invalid input trees have to be rejected and a more relaxed setting, where for each valid input tree the transducer selects one output tree, on each other input tree which is not part of the domain the transducer may behave arbitrarily. To distinguish between these cases, we will speak of uniformization with input validation and uniformization without input validation.

For Section 3.1 and Section 3.2, let $R \subseteq T_\Sigma \times T_\Gamma$ be a deterministic top-down tree automaton-definable relation with total domain and let $\mathcal{A} = (Q_\mathcal{A}, \Sigma_\bot \times \Gamma_\bot, q_0^\mathcal{A}, \Delta_\mathcal{A})$ be a D↓TA that recognizes R. Further, we assume Σ is given as $\bigcup_{i=0}^{m} \Sigma_i$. For $q \in Q_\mathcal{A}$, let $\mathcal{A}_q$ be the automaton that results from $\mathcal{A}$ by using q as single initial state.

To begin with, we investigate the connection between input and output. Intuitively, given an input tree t and an output tree t', a deterministic tree-automatic specification may only set a node u of t' in relation to a node v of t if v is a predecessor resp. successor of u, i.e., if $v \sqsubseteq u$ resp. $u \sqsubseteq v$. This is due to the fact that deterministic top-down tree automata cannot specify conditions that require information from divergent paths. However, a top-down tree transducer may produce output at a node u while reading some v' such that v' lies on a divergent path, i.e., $v' \not\sqsubseteq u$ and $u \not\sqsubseteq v'$. Later on, we will see that if a uniformizer does not use the input information on a path along u (but from a divergent path) in order to produce output at u, then the deterministic specification in fact does not relate – from some point on – the output to the input. Consequently, the specification can be satisfied by selecting a single output tree that matches all possible input trees. The lemma below shows that this property is decidable.

Lemma 3 *Given $\mathcal{A}$ and a state q of $\mathcal{A}$, the following properties are decidable:*

1. $\forall t \in T_\Sigma : t \otimes \bot \in T(\mathcal{A}_q)$,
2. $\exists t' \in T_\Gamma : \bot \otimes t' \in T(\mathcal{A}_q)$,

3. $\exists t' \in T_\Gamma\, \forall t \in T_\Sigma : t \otimes t' \in T(\mathcal{A}_q)$.

Proof.

1. Let $T_1 = T(\mathcal{A}_q) \cap (T_\Sigma \times \{\bot\})$, then $\forall t \in T_\Sigma : t \otimes \bot \in T(\mathcal{A}_q) \Leftrightarrow T_1$ is universal (over $\Sigma \times \{\bot\}$).

2. Let $T_2 = T(\mathcal{A}_q) \cap (\{\bot\} \times T_\Gamma)$, then $\exists t' \in T_\Gamma : \bot \otimes t' \in T(\mathcal{A}_q) \Leftrightarrow T_2$ is non-empty.

3. Let T_3 be the projection of $\overline{T(\mathcal{A}_q)} \cap (T_\Sigma \times T_\Gamma)$ onto its second component, i.e., $T_3 = \{t' \in T_\Gamma \mid \exists t \in T_\Sigma : t \otimes t' \notin T(\mathcal{A}_q)\}$. Then $\exists t' \in T_\Gamma\, \forall t \in T_\Sigma : t \otimes t' \in T(\mathcal{A}_q) \Leftrightarrow \overline{T_3}$ is non-empty.

Since regular tree languages are closed under Boolean operations and projection, T_1, T_2, and T_3 are regular. It follows that properties 1–3 are decidable because universality and emptiness of regular tree languages is decidable. □

3.1 Bounded delay

In this section, we consider the question whether there exists a uniformization of a D↓TA-recognizable relation with total domain such that the output delay remains inside a given bound. We will solve the following problem.

Theorem 4 *Given $k \geq 0$, it is decidable whether a given D↓TA-recognizable relation with total domain has a uniformization by a deterministic top-down tree transducer with output delay bounded by k.*

Before we present a decision procedure, we introduce some notations that will simplify the presentation. Given $\Sigma = \bigcup_{i=0}^{m} \Sigma_i$, let $\mathrm{dir}_\Sigma = \{1, \ldots, m\}$ be the set of directions compatible with Σ. For $\Sigma = \bigcup_{i=0}^{m} \Sigma_i$, the set Path_Σ of labeled paths over Σ is defined inductively by:

- ε is a labeled input path and each $f \in \Sigma$ is a labeled input path,
- given a labeled input path $\pi = x \cdot f$ with $f \in \Sigma_i$ $(i > 0)$ over Σ, then $\pi \cdot jg$ with $j \in \{1, \ldots, i\}$ and $g \in \Sigma$ is a labeled input path.

For $\pi \in \mathrm{Path}_\Sigma$, we define the path $path(\pi) \in \mathrm{dir}_\Sigma^*$ and the word $labels(\pi) \in \Sigma^*$ induced by π inductively by:

- if $\pi = \varepsilon$ or $\pi = f$, then $path(\varepsilon) = path(f) = \varepsilon$, $labels(\varepsilon) = \varepsilon$ and $labels(f) = f$,
- if $\pi = x \cdot jf$ with $j \in \mathrm{dir}_\Sigma$, $f \in \Sigma$, then $path(\pi) = path(x) \cdot j$, $labels(\pi) = labels(x) \cdot f$.

The length $||\,||$ of a labeled path over Σ is the length of the word induced by its path, i.e., $||\pi|| = |labels(\pi)|$.

For $\pi \in \text{Path}_\Sigma$ with $||\pi|| = k$ let

$$T^\pi_\Sigma := \{t \in T_\Sigma \mid val_t\big(path(\pi)[1,(i-1)]\big) = labels(\pi)[i] \text{ for } 1 \leq i \leq k\}$$

be the set of trees t over Σ such that π is a prefix of a labeled path through t.

For $t \in T_\Sigma$ and $u \in \text{dir}^*_\Sigma$, let $||t||^u := max\{|v| \mid v \in dom_t \text{ and } (u \sqsubseteq v \text{ or } v \sqsubseteq u)\}$. If $u \in dom_t$, then $||t||^u$ is the length of a maximal path in t through u. Otherwise, if $u \notin dom_t$, then $||t||^u$ is the length of the unique path in t that is a prefix of u.

Now, in order to solve the above decision problem, we consider a turn-based safety game between two players. The procedure is similar to a decision procedure presented in [6], where the question whether a uniformization of an automatic word relation by a word transducer exists, is reduced to the existence of winning strategies in a safety game. Here, the game is played between In and Out, where In can follow any path from the root to a leaf in an input tree such that In plays one input symbol at a time. Out can either react with an output symbol, or delay the output a bounded number of times and react with a direction in which In should continue with his input sequence. The objective of Out is to ensure that the pair of input sequence and output sequence obtained from the moves of the players satisfies the specification.

The vertices in the game graph keep track of the state of $\mathcal{A}$ on the input combined with the output on the same path and additionally of the input that is ahead, which is bounded by k. We will see that it is not necessary to consider situations where input and output are on divergent paths. The intuition behind this is that D↓TAs cannot compare information on divergent paths through an input tree. Formally, the game graph $G^k_\mathcal{A}$ is constructed as follows.

- $V_{\mathsf{In}} = \{\big(q, \pi j\big) \in Q_\mathcal{A} \times \text{Path}_\Sigma \cdot \text{dir}_\Sigma \mid \|\pi\| \leq k, \pi \in \text{Path}_\Sigma, j \in \text{dir}_\Sigma\} \cup 2^{Q_\mathcal{A}}$ is the set of vertices of player In.
- $V_{\mathsf{Out}} = \{\big(q, \pi\big) \in Q_\mathcal{A} \times \text{Path}_\Sigma \mid \|\pi\| \leq k+1\}$ is the set of vertices of player Out.
- From a vertex of In the following moves are possible:
  - i) $\big(q, \pi j\big) \to \big(q, \pi j f\big)$ for each $f \in \Sigma$ if $\|\pi\| < k+1$ (delay; In chooses the next input symbol)
  - ii) $\{q_1, \dots, q_n\} \to \big(q_i, f\big)$ for each $f \in \Sigma$ and each $i \in \{1, \dots, n\}$

    (no delay; In chooses the next direction and input symbol)
- From a vertex of Out the following moves are possible:
  - iii) $\big(q, f\big) \xrightarrow{r} \{q_1, \dots, q_i\}$ if there is $r = (q, (f,g), q_1, \dots, q_n) \in \Delta_\mathcal{A}$, $f \in \Sigma$ is i-ary, $g \in \Sigma_\bot$ is j-ary, $n = max\{i, j\}$, and if $j > i$, there exist trees $t_{i+1}, \dots, t_j \in T_\Gamma$ such that $\bot \otimes t_\ell \in T(\mathcal{A}_{q_l})$ for all $i < \ell \leq j$.

    (no delay; Out applies a transition; Out can pick output trees for all directions where the input has ended; In can continue from the other directions)

Note, if $f \in \Sigma_0$, i.e., the input symbol is a leaf, then the next reached vertex is $\emptyset \in V_{\mathsf{In}}$, which is a terminal vertex.

iv) $(q, fj\pi) \xrightarrow{r} (q_j, \pi)$ if there is $r = (q, (f, g), q_1, \dots, q_n) \in \Delta_{\mathcal{A}}$ with $n = max\{rk(f), rk(g)\}$ such that for each $\ell \neq j$ with $\ell \in \{1, \dots, n\}$ holds

- if $\ell \leq rk(f), rk(g)$, then there exists $t' \in T_\Gamma$ such that $t \otimes t' \in T(\mathcal{A}_{q_\ell})$ for all $t \in T_\Sigma$,

  (input and output continue)
- if $rk(g) < \ell \leq rk(f)$, then $t \otimes \bot \in T(\mathcal{A}_{q_\ell})$ for all $t \in T_\Sigma$,

  (input continues, output has ended)
- if $rk(f) < \ell \leq rk(g)$, then there exists $t' \in T_\Gamma$ such that $\bot \otimes t' \in T(\mathcal{A}_{q_\ell})$.

  (output continues, input has ended)

(delay; Out applies a transition, removes the leftmost input symbol and advances in direction of the labeled path ahead; Out can pick output trees for all divergent directions)

v) $(q, \pi j f) \rightarrow (q, \pi j f j')$ for each $j' \in \{1, \dots, i\}$ for $f \in \Sigma_i$ if $\|\pi j f\| < k + 1$

(Out delays and chooses a direction from where In should continue)

- The initial vertex is $\{q_0^{\mathcal{A}}\}$.

Note that the game graph can effectively be constructed, because Lemma 3 implies that it is decidable whether the edge constraints are satisfied. The edge labels are introduced to improve readability, but they are not necessary.

The winning condition should express that player Out loses the game if the input can be extended, but no valid output can be produced. This is represented in the game graph by a set of bad vertices B that contains all vertices of Out with no outgoing edges. If one of these vertices is reached during a play, Out loses the game. Thus, we define $\mathcal{G}_{\mathcal{A}}^k = (G_{\mathcal{A}}^k, V \setminus B)$ as safety game for Out.

Example 5 Let Σ be an input alphabet given by $\Sigma_2 = \{f\}$ and $\Sigma_0 = \{a\}$ and let Γ be an output alphabet given by $\Gamma_2 = \{f, g\}$ and $\Gamma_0 = \{b\}$. Consider the relation $R := \{(t, t') \in T_\Sigma \times T_\Gamma \mid dom_t = dom_{t'}$ and in t', every path of length at least 1 contains an $f\}$.

It is easy to see that D↓TA $\mathcal{A} = (\{q_0, q, q_f\}, \Sigma \times \Gamma, q_0, \Delta_{\mathcal{A}})$ with $\Delta_{\mathcal{A}} = \{(q_0, (a, b)),\ (q_0, (f, f), q_f, q_f),\ (q_0, (f, g), q, q),\ (q, (f, f), q_f, q_f),\ (q, (f, g), q, q),\ (q_f, (a, b)),\ (q_f, (f, f), q_f, q_f),\ (q_f, (f, g), q_f, q_f)\}$ recognizes R. For $k = 1$, the corresponding game graph $G_{\mathcal{A}}^0$ is depicted in Figure 2. ◁

The following two lemmata show that from the existence of a winning strategy a top-down tree transducer that uniformizes the relation can be obtained and vice versa.

Lemma 6 *If Out has a winning strategy in $\mathcal{G}_{\mathcal{A}}^k$, then R has a uniformization by a D↓TT in which the output delay is bounded by k.*

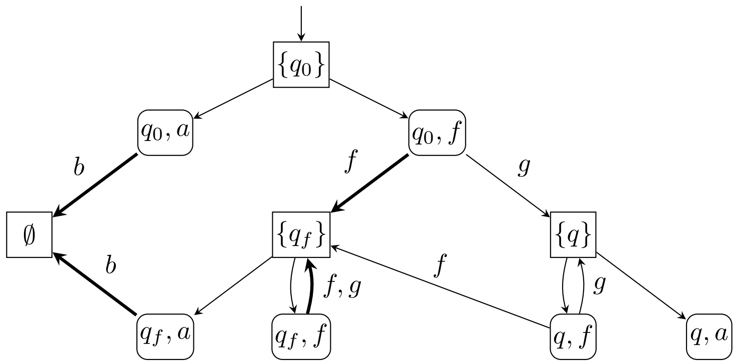


Figure 2: The game graph $G^0_{\mathcal{A}}$ constructed from the D↓TA $\mathcal{A}$ from Example 5. A possible winning strategy for Out in $\mathcal{G}^0_{\mathcal{A}}$ is emphasized in the graph.

Proof. Assume that Out has a winning strategy in the safety game $\mathcal{G}^k_{\mathcal{A}}$, then there is also a positional one. We can represent a positional winning strategy by a function $\sigma : V_{\mathsf{Out}} \to \Delta_{\mathcal{A}} \cup \mathrm{dir}_\Sigma$, because Out either plays one output symbol (corresponding to a unique transition in $\Delta_{\mathcal{A}}$), or a new direction for an additional input symbol.

We construct a deterministic ↓TT $\mathcal{T} = (Q_{\mathcal{A}} \cup \{(q, \pi j) \mid (q, \pi j f) \in V_{\mathsf{Out}}\}, \Sigma, \Gamma, q_0^{\mathcal{A}}, \Delta)$ from such a positional winning strategy σ as follows:

a) For each $\sigma : (q, f) \overset{r}{\mapsto} \{q_1, \ldots, q_i\}$ with $r = (q, (f, g), q_1, \ldots, q_n) \in \Delta_{\mathcal{A}}$:

   - add $q(f(x_1, \ldots, x_i)) \to g(q_1(x_1), \ldots, q_j(x_j))$ to Δ if $j \leq i$, or
   - add $q(f(x_1, \ldots, x_i)) \to g(q_1(x_1), \ldots, q_i(x_i), t_{i+1}, \ldots, t_j)$ to Δ if $j > i$

   where $f \in \Sigma_i$, $g \in \Gamma_j$ and $t_{i+1}, \ldots, t_j \in T_\Gamma$ chosen according to the r-edge constraints in (q, f).

b) For each $\sigma : (q, \pi j f) \mapsto (q, \pi j f j')$ add $(q, \pi j)(f(x_1, \ldots, x_i)) \to (q, \pi j f j')(x_{j'})$ to Δ.

If the strategy σ defines a sequence of moves of Out inside vertices of V_{Out}, that is, a sequence of moves of type iv), then this corresponds to an output sequence that is produced without reading further input. Each output of these moves can be represented by a special tree s as follows. A move of type iv) has the form $(q, fj\pi) \overset{r}{\to} (q', \pi)$ with $r = (q, (f, g), q_1, \ldots, q_n)$ and $q' = q_j$. Then, let $s = g(t_1, \ldots, t_{j-1}, \circ, t_{j+1}, \ldots, t_n) \in S_\Sigma$ be the special tree, where each $t_l \in T_\Gamma$ is chosen according to the r-edge constraints in $(q, fj\pi)$ for $l \neq j, 1 \leq l \leq n$. Eventually, the strategy defines a move of Out of type iii) or v) to a node of V_{In}, otherwise σ is not a winning strategy. These parts of the strategy are transformed as follows:

c) For each $(q, \pi j f) \xrightarrow{r_1} \ldots \xrightarrow{r_{l-1}} (q', \pi' j f) \to (q', \pi' j f j')$ add $(q, \pi j)(f(x_1, \ldots, x_i)) \to s_1 \cdot \ldots \cdot s_{l-1} \cdot (q', \pi' j f j')(x_{j'})$ to Δ, where each $s_i \in S_\Gamma$ is a special tree corresponding to the r_i-edge in the ith move.

d) For each $(q, \pi j f) \xrightarrow{r_1} \ldots \xrightarrow{r_{l-1}} (q', f) \xrightarrow{r_l} \{q_1, \ldots, q_i\}$ add $(q, \pi j)(f(x_1, \ldots, x_i)) \to s_1 \cdot \ldots \cdot s_{l-1} \cdot s$ to Δ, where each $s_i \in S_\Gamma$ is a special tree corresponding to the r_i-edge in the ith move and s is an output corresponding to r_l constructed as described in step a).

We now verify that $\mathcal{T}$ defines a uniformization of R. Clearly, $R(\mathcal{T}) \subseteq R$ because every rule of $\mathcal{T}$ corresponds to a rule of $\mathcal{A}$. We have to show $dom(R(\mathcal{T})) = dom(R) = T_\Sigma$. Let $t \in T_\Sigma$. We can show by induction on the number of steps needed to reach a configuration from the initial configuration $(t, q_0^{\mathcal{A}}, \varphi_0)$ that for each configuration $c = (t, t', \varphi)$ such that $D_{t'} \neq \emptyset$, in other words $t' \notin T_\Gamma$, there exists a successor configuration c'. The induction hypothesis states that in c, if $u \in D_{t'}$ with $\varphi(u) = v$, $val_t(v) = f \in \Sigma$, $val_{t'}(u) = (q, yj) \in Q_{\mathcal{A}} \times \mathrm{Path}_\Sigma \cdot \mathrm{dir}_\Sigma$, then the following conditions are satisfied:

- $t \in T_\Sigma^{xiyjf}$ for some $x \in \mathrm{Path}_\Sigma$ and $i \in \mathrm{dir}_\Sigma$ with $path(x)i = u$ and $path(xiy)j = v$, and
- in a play according to σ, the vertex $(q, yjf) \in V_{\mathsf{Out}}$ is reached after a sequence of moves corresponding to the labeled path $xiyjf$.

Since σ is a winning strategy, σ defines the next move of Out from (q, yjf). Consequently, there exists a corresponding transition with left-hand side $(q, yj)(f(x_1, \ldots, x_{rk(f)}))$ in $\Delta_{\mathcal{T}}$ that is applicable at u in t'. This guarantees the existence of a successor configuration c'. If $val_t(v) \in \Sigma_0$, i.e., if $\mathcal{T}$ reads a leaf, then by construction the transition that is applicable at u in t' is a rule of type a) or d) where the right-hand side is a tree over Γ.

From the above induction it follows that $(t, q_0^{\mathcal{A}}, \varphi_0) \to_{\mathcal{T}}^* (t, \mathcal{T}(t), \varphi)$ with $D_{\mathcal{T}(t)} = \emptyset$, i.e., $\mathcal{T}(t) \in T_\Gamma$, because in each computation step one input symbol is read and eventually, a leaf is reached and the output is a tree over Γ.

In $\mathcal{T}$ the output delay is bounded by k, because the existence of a winning strategy σ guarantees that from a vertex (q, π) with $||\pi|| = k$, that is reachable by playing according to σ, a move of Out follows. It follows from the construction that $\mathcal{T}$ produces output accordingly. □

The size of $G_{\mathcal{A}}^k$ is at most $Q_{\mathcal{A}} \cdot (|\Sigma| \cdot |\mathrm{dir}_\Sigma|)^{k-1} \cdot |\Sigma| + 2^{Q_{\mathcal{A}}}$. For the winning player, a positional winning strategy can be determined in linear time in the size of $G_{\mathcal{A}}^k$ (see Theorem 3.1.2 in [13], which can easily be adapted to safety games). For a positional winning strategy of Out, the above construction yields a D↓TT with delay bounded by k that uses at most $Q_{\mathcal{A}} \cdot (|\Sigma| \cdot \mathrm{dir}_\Sigma)^{k-1}$ states.

Before we show the other direction, we introduce some new notations. Since we will consider labeled paths through trees, it is convenient to define the notion of convolution also for labeled paths. For a labeled path $x \in \mathrm{Path}_\Sigma$ with $||x|| > 0$, let $dom_x := \{u \in \mathrm{dir}_\Sigma^* \mid u \sqsubseteq path(x)\}$ and $val_x : dom_x \to \Sigma$, where $val_x(u) = labels(x)[i]$ if $u \in dom_x$ with $|u| = i + 1$. Let $x \in \mathrm{Path}_\Sigma$, $y \in \mathrm{Path}_\Gamma$ with

$path(y) \sqsubseteq path(x)$ or $path(x) \sqsubseteq path(y)$, then the *convolution* of x and y is $x \otimes y$ defined by $dom_{x\otimes y} = dom_x \cup dom_y$, and $val_{x\otimes y}(u) = (val_x^{\perp}(u), val_y^{\perp}(u))$ for all $u \in dom_{x\otimes y}$, where $val_x^{\perp}(u) = val_x(u)$ if $u \in dom_x$ and $val_x^{\perp}(u) = \perp$ otherwise, analogously defined for $val_y^{\perp}(u)$.

Furthermore, it is useful to relax the notion of runs to labeled paths. Let $x \in \mathrm{Path}_\Sigma$, $y \in \mathrm{Path}_\Gamma$ such that $x \otimes y$ is defined, i.e., $path(y) \sqsubseteq path(x)$ or $path(x) \sqsubseteq path(y)$. We define the run of $\mathcal{A}$ on $x \otimes y$ such that it maps all nodes from $dom_{x\otimes y}$ as well as all nodes that are a direct successor of a node from $dom_{x\otimes y}$ to a state of $\mathcal{A}$. Formally, let the (partial) *run* of $\mathcal{A}$ on $x \otimes y$ be the partial function $\rho : \mathrm{dir}_\Sigma^* \to Q_\mathcal{A}$ such that $\rho(\varepsilon) = q_0^\mathcal{A}$, and for each $u \in dom_{x\otimes y}$: if $q := \rho(u)$ is defined and there is a transition $(q, val_{x\otimes y}(u), q_1, \ldots, q_i) \in \Delta_\mathcal{A}$, then $\rho(u.j) = q_j$ for all $j \in \{1, \ldots, i\}$. Let $path(x \otimes y) = v$ and $i \in \mathrm{dir}_\Sigma$. Shorthand, we write

$$\mathcal{A} : q_0^\mathcal{A} \xrightarrow{x\otimes y}_i q,$$

if $q := \rho(vi)$ is defined. We write $\mathcal{A} : q_0^\mathcal{A} \xrightarrow{x\otimes y} Acc$ if $rk(val_{x\otimes y}(v)) = 0$ and $(\rho(v), val_{x\otimes y}(v)) \in \Delta_\mathcal{A}$ to indicate that the (partial) run ρ of $\mathcal{A}$ on $x \otimes y$ is accepting.

We now show the other direction.

Lemma 7 *If R has a uniformization by a D↓TT in which the output delay is bounded by k, then* Out *has a winning strategy in $\mathcal{G}_\mathcal{A}^k$.*

Proof. Assume that R has a uniformization by some D↓TT $\mathcal{T} = (Q_\mathcal{T}, \Sigma, \Gamma, q_0^\mathcal{T}, \Delta_\mathcal{T})$ in which the output delay is bounded by k. A winning strategy for Out basically takes the moves corresponding to the output sequence that $\mathcal{T}$ produces for a read input sequence induced by the moves of In. We show by induction on the number of moves played by Out that the strategy in $\mathcal{G}_\mathcal{A}^k$ can be chosen such that in every play according to the strategy the following induction hypothesis is satisfied. In a vertex $(q, y) \in V_{\mathsf{Out}}$ that is reached by a sequence of moves that describe a labeled path $xiy \in \mathrm{Path}_\Sigma$ with $x, y \in \mathrm{Path}_\Sigma$, $i \in \mathrm{dir}_\Sigma$, $path(x)i = u$ and $path(xiy) = v$ holds:

a) There exists a tree $t \in T_\Sigma^{xiy}$ and a tree $s = t[\circ/v] \cdot val_t(v)$ obtained from t by deleting all nodes below v such that the deterministic (partial) run $\rho_\mathcal{A}$ of $\mathcal{A}$ on $x \otimes o$ yields
$$\mathcal{A} : q_0^\mathcal{A} \xrightarrow{x\otimes o}_i q,$$
where o is the unique labeled path over Γ such that $\mathcal{T}(s) \in T^o_{\Gamma\cup Q_\mathcal{T}}$ with $dom_{x\otimes o} = dom_x$, and

b) if $||y|| > 1$, then $(t, q_0^\mathcal{T}, \varphi_0) \to_\mathcal{T}^* (t, t_{\mathrm{out}}, \varphi)$ such that there is $w \in D_{t_{\mathrm{out}}}$ with $w \sqsubseteq u$ and $\varphi(w) = v$.

Condition a) states that the final transformed output $\mathcal{T}(s)$ of the computation of $\mathcal{T}$ on s specifies (at least) the output at every node on $path(x)$, which means the partial run $\rho_\mathcal{A}$ on $x \otimes o$ already defines the state that $\mathcal{A}$ reaches at u, because $u = path(x)i$. Note that if $||o|| < ||x||$, then o is a labeled path that ends with a leaf symbol, otherwise it is not possible that $\rho_{\mathcal{A}}(u)$ is defined. Condition b) states that in the computation of $\mathcal{T}$ on t, the output mapped to the subtree rooted at w in $\mathcal{T}(t)$ is dependent on the subtree rooted at v in t.

For the induction base, clearly, the induction hypothesis is true at the first reached vertex of Out, because $\rho_{\mathcal{A}}(\varepsilon) = q_0^{\mathcal{A}}$ for every run of $\mathcal{A}$.

For the induction step, assume we are in a node $(q, y) \in V_{\mathsf{Out}}$ after a sequence of moves that describe a labeled path $xiy \in \mathrm{Path}_\Sigma$ and the induction hypothesis holds for $t \in T_\Sigma^{xiy}$ and s obtained from t. From the induction hypothesis, we know that in $\mathcal{T}(s)$ output for every predecessor of u is specified. Then, Out can make her next move according to whether in $\mathcal{T}(s)$ also output at u is specified. Formally, this means we check if $val_{s\otimes\mathcal{T}(s)}(u) \in \Sigma \times \Gamma_\perp$, then output at u is specified. If the second component is $\perp$ this implies that the output sequence has already ended. As a consequence, Out has to chose $\perp$ as output move in this and in every subsequently reached vertex in this play, until the play ends. Otherwise, $val_{s\otimes\mathcal{T}(s)}(u) \in \Sigma \times Q_{\mathcal{T}}$, then output at u is not yet specified.

If no output was specified, i.e., $val_{s\otimes\mathcal{T}(s)}(u) \in \Sigma \times Q_{\mathcal{T}}$, this implies that in the computation of $\mathcal{T}$ on t the output at u is dependent on a node below v, because s contains no nodes below v. Thus, it must hold that there is a configuration (t, t', φ) of $\mathcal{T}$ with $(t, q_0^{\mathcal{T}}, \varphi_0) \to_{\mathcal{T}}^* (t, t', \varphi)$ such that $u \in D_{t'}$ with $\varphi(u) = vj$ for some $j \in \mathrm{dir}_\Sigma$. Then, Out delays and chooses direction j as next move. This can happen at most $k - ||y||$ times in a row because the output delay in $\mathcal{T}$ is bounded by k. The next reached vertex of Out is of the form $(q, yjg) \in V_{\mathsf{Out}}$ for some $g \in \Sigma$. In this vertex, the induction hypothesis can be satisfied for every $\hat{t} \in \{t[\circ/vj] \cdot s' \mid s' \in T_\Sigma^g\} \subseteq T_\Sigma^{xiyjg}$. We pick any such $\hat{t}$ from this set and denote by $\hat{s}$ the corresponding tree $\hat{t}[\circ/vj] \cdot g$ that results from $\hat{t}$ by deleting all nodes below vj. We now show that the induction hypothesis is satisfied. Clearly, condition a) is satisfied because $\mathcal{T}(\hat{s}) \in T_\Gamma^o$ and thus $\mathcal{A} : q_0^{\mathcal{A}} \xrightarrow{x\otimes o}_i q$. Also, condition b) is satisfied, because as explained above there is a configuration $(\hat{t}, \hat{t}_{\mathrm{out}}, \varphi)$ of $\mathcal{T}$ on $\hat{t}$ reachable such that $w \in D_{\hat{t}_{\mathrm{out}}}$ with $\varphi(w) = vj$ and $w = u$.

Otherwise, output was specified, i.e., $val_{s\otimes\mathcal{T}(s)}(u) \in \Sigma \times \Gamma_\perp$. Let $val_{s\otimes\mathcal{T}(s)}(u) = (f, f')$. Hence, there exists a rule r of the form $(q, (f, f'), q_1, \dots, q_n) \in \Delta_{\mathcal{A}}$ since $\mathcal{T}$ uniformizes R. We have to show that there is also an outgoing r-edge from (q, y) that Out can take. Clearly, this is the case if $y = f$, i.e., $||y|| = 1$. Next, Out reaches a vertex of the form $(q_j, g) \in V_{\mathsf{Out}}$ for some $j \le rk(f)$ and some $g \in \Sigma$. The induction hypothesis is satisfied for every $\hat{t} \in \{t[\circ/vj] \cdot s' \mid s' \in T_\Sigma^g\}$ because $\mathcal{A} : q_0^{\mathcal{A}} \xrightarrow{x\otimes o}_i q \xrightarrow{f\otimes f'}_j q_j$.

So, assume $||y|| > 1$ and let $y = fjy'$. We have to prove that the r-edge constraints of type iv) are satisfied. That means, we have to show for each direction $\ell \neq j$ with $\ell \in \{1, \dots, n\}$ that there exists an output tree that matches all possible input trees. More formally, we have to show for each $\ell \neq j$ that there exists an output tree s_ℓ such that $T_\Sigma \times \{s_\ell\} \subseteq T(\mathcal{A}_{q_\ell})$ (or $\{\perp\} \times \{s_\ell\} \subseteq T(\mathcal{A}_{q_\ell})$ if $\ell > rk(f)$).

We pick $\mathcal{T}(t)|_{u\ell}$ for s_ℓ. Towards a contradiction, assume $T_\Sigma \times \{s_\ell\} \not\subseteq T(\mathcal{A}_{q_\ell})$. Then, there is some t_ℓ such that $t_\ell \otimes s_\ell \notin T(\mathcal{A}_{q_\ell})$. Consider the tree $t' = t[\circ/u\ell] \cdot t_\ell$ obtained from t by replacing the subtree rooted at $u\ell$ by t_ℓ. Since $t' \in$

$dom(R)$, $t' \otimes \mathcal{T}(t') \in T(\mathcal{A})$. The induction hypothesis states that $(t, q_0^{\mathcal{T}}, \varphi_0) \rightarrow^*_{\mathcal{T}} (t, t_{\text{out}}, \varphi)$ such that $w \in D_{t_{\text{out}}}$ with $w \sqsubseteq u \sqsubseteq \varphi(w) = v$. This means, the output mapped to the subtree rooted at w is dependent on the subtree rooted at v. Since u is below w, this also means, the output mapped to the subtree rooted at u is dependent on the subtree rooted at v. Hence, $\mathcal{T}(t)|_u = \mathcal{T}(t')|_u$ because $t|_v = t'|_v$. Particularly, also $\mathcal{T}(t)|_{u\ell} = \mathcal{T}(t')|_{u\ell} = s_\ell$. Moreover, the run ρ of $\mathcal{A}$ on $t' \otimes \mathcal{T}(t')$ yields $\rho(u\ell) = q_\ell$. Thus, $t_\ell \otimes \mathcal{T}(t')|_{u\ell} = t_\ell \otimes s_\ell \in T(\mathcal{A}_{q_\ell})$, which is a contradiction.

Consequently, an r-edge from $(q, y) \in V_{\mathsf{Out}}$ exists that leads to $(q_j, y') \in V_{\mathsf{Out}}$. There, the induction hypothesis is satisfied for t and $(t, t_{\text{out}}, \varphi)$ as above, because $\mathcal{A} : q_0^{\mathcal{A}} \xrightarrow{x \otimes o}_i q \xrightarrow{f \otimes f'}_j q_j$ and $w \in D_{t_{\text{out}}}$ with $w \sqsubseteq uj \sqsubseteq \varphi(w) = v$.

As we have seen, Out never reaches a vertex without outgoing edges and therefore wins. □

As a consequence of Lemma 6 and Lemma 7 together with the fact that a winning strategy for Out can effectively be computed in $\mathcal{G}^k_{\mathcal{A}}$ we immediately obtain Theorem 4. Moreover, Lemmata 6 and 7 show that if there exists a uniformizer, then there also exists a uniformizer such that read input and produced output are always on the same path.

3.2 Arbitrary delay

Previously, we considered the question whether there exists a uniformization of a D↓TA-recognizable relation with total domain such that the output delay remains inside a given bound. In this section, we will show that this question is also decidable for the class of all deterministic top-down tree transducers, i.e., in particular, there is no restriction on the output delay. We provide a decision procedure that reduces this question to deciding the winner in a safety game similar to the safety game presented in the previous section on bounded delay.

To begin with, we give an example of a relation that is uniformizable, but not by a transducer that has bounded delay.

Example 8 Let Σ be given by $\Sigma_2 = \{f\}$ and $\Sigma_0 = \{a, b\}$. We consider the D↓TA-recognizable relation $R \subseteq T_\Sigma \times T_\Sigma$ that is defined by $\{(t, a) \mid \text{the leftmost leaf in } t \text{ is } a\} \cup \{(t, b) \mid \text{the leftmost leaf in } t \text{ is } b\}$.

A transducer can uniformize R by reading the leftmost path in an input tree without producing output. Eventually, when a leaf is reached, then the transducer outputs this leaf. This is realized by the following D↓TT $\mathcal{T} = (\{q\}, \Sigma, \Sigma, q, \Delta)$ with $\Delta = \{\ q(f(x_1, x_2)) \to q(x_1),\ q(a) \to a,\ q(b) \to b\ \}$.

However, there is no D↓TT with bounded delay that uniformizes R. Intuitively, a D↓TT that uniformizes R has to know the label of the leftmost leaf of the input tree in order to produce the correct output. If the output delay is bounded, say by k, then a D↓TT can wait at most k computation steps before is has to produce either a or b. If the length of the leftmost path in an input exceeds k, then the D↓TT has to guess the correct output, but there is no guarantee that the guess was right. ◁

Similar to [6] for automatic word relations, we will see that if the output delay exceeds a certain bound, then we can decide whether the uniformization is possible or not. The intuition is that if it is necessary to have such a long delay between input and output, then only one path in the tree is relevant to determine an output tree. We can define this property by introducing the term path-recognizable function. If a relation is uniformizable by a path recognizable function, then the relation has a uniformization by a D↓TT that first deterministically reads one path of the input tree and then outputs a matching output tree.

Formally, we say a relation R is *uniformizable by a path-recognizable function*, if there exists a D↓TT $\mathcal{T}$ that uniformizes R such that $\Delta_\mathcal{T}$ only contains transitions of the following form:

$$q(f(x_1,\dots,x_i)) \to q'(x_{j_1}) \quad \text{or} \quad q(a) \to t,$$

where $f \in \Sigma_i$, $i > 0$, $a \in \Sigma_0$, $q, q' \in Q$ and $j_1 \in \{1,\dots,i\}$ and $t \in T_\Gamma$.

Towards a decision procedure for uniformization by D↓TTs with arbitrary delay, we first show that it is decidable for a D↓TA-recognizable relation, whether it is uniformizable by a path-recognizable function (Theorem 9).

Beforehand, we need to fix some notations. For $R \subseteq T_\Sigma \times T_\Gamma$, $\pi \in \mathrm{Path}_\Sigma$ and $q \in Q_\mathcal{A}$ let

$$R^\pi := \{(t,t') \in R \mid t \in T_\Sigma^\pi\} \text{ and } R_q^\pi := \{(t,t') \in R(\mathcal{A}_q) \mid t \in T_\Sigma^\pi\}.$$

If $q = q_0^\mathcal{A}$, then R_q^π corresponds to R^π, if additionally $\pi = \varepsilon$, then R_q^π corresponds to R. Note that a D↓TA that recognizes R_q^π can be easily constructed from $\mathcal{A}$.

Sometimes it is sufficient to consider only the output that is mapped to a certain path. Recall, $||\mathcal{T}(t)||^u$ denotes the length of a maximal path in $\mathcal{T}(t)$ through (resp. the length of the maximal path in $\mathcal{T}(t)$ that is a prefix of) u. For a D↓TT $\mathcal{T}$, an input tree $t \in T_\Sigma$ or $t \in S_\Sigma$ and a path $u \in \mathrm{dir}_\Sigma^*$, we define

$$out_\mathcal{T}(t,u) := \{\pi \in \mathrm{Path}_\Gamma \mid \mathcal{T}(t) \in T_\Gamma^\pi(X) \text{ and } (path(\pi) \sqsubseteq u \text{ or } u \sqsubseteq path(\pi)) \text{ and } ||\mathcal{T}(t)||^u = |path(\pi)|\}$$

to be the set of maximal labeled paths in the output tree $\mathcal{T}(t)$ through u. Note that if $||\mathcal{T}(t)||^u < |u|$, then $out_\mathcal{T}(t,u)$ is a singleton set and contains the maximal labeled path π in $\mathcal{T}(t)$ such that $path(\pi)$ is a prefix of u. Then, we identify $out_\mathcal{T}(t,u)$ with its single element.

Since we want to capture uniformizability by path-recognizable functions, we introduce a partial function that yields the state transformation that a labeled path together with some output sequence on this path induces. However, we are only interested in the result of the state transformation if it suffices for a uniformizer to read this labeled path segment in an input tree to (partially) determine a matching output tree. Formally, let $x \in \mathrm{Path}_\Sigma$, $y \in \mathrm{Path}_\Gamma$ and $i \in \mathrm{dir}_\Sigma$ such that $x \otimes y$ is defined, and let ρ_q be the partial run of $\mathcal{A}_q$ on $x \otimes y$. We define the partial function $\tau_{xi,y} : Q_\mathcal{A} \to Q_\mathcal{A}$ with $\tau_{xi,y}(q) := q'$ if $\mathcal{A}_q : q \xrightarrow{x\otimes y}_i q'$ and for each uj with $u \in dom_{x\otimes y}, uj \not\sqsubseteq path(x \otimes y)i$, and $j \in \{1,\dots,rk\big((val_x^\perp(u), val_y^\perp(u))\big)\}$ holds

- if $r := \rho_q(uj)$ and $j \leq rk(val_x^{\perp}(u)), rk(val_y^{\perp}(u))$, then there exists $t' \in T_\Gamma$ such that for all $t \in T_\Sigma$ holds $t \otimes t' \in T(\mathcal{A}_r)$, and
- if $r := \rho_q(uj)$ and $rk(val_y^{\perp}(u)) < j \leq rk(val_x^{\perp}(u))$, then for all $t \in T_\Sigma$ holds $t \otimes \perp \in T(\mathcal{A}_r)$, and
- if $r := \rho_q(uj)$ and $rk(val_x^{\perp}(u)) < j \leq rk(val_y^{\perp}(u))$, then there exists $t' \in T_\Gamma$ such that $\perp \otimes t' \in T(\mathcal{A}_r)$.

Lemma 3 implies that it is decidable whether $\tau_{xi,y}(q)$ is defined. Basically, if $\tau_{xi,y}(q)$ is defined, then there exists a fixed (partial) output tree $s' \in S_\Gamma^{yi\circ}$ such that for each input tree $t \in T_\Sigma^x$ there exists some $t' \in T_\Gamma$ such that $t \otimes (s' \cdot t') \in T(\mathcal{A}_q)$. For convenience, if x resp. y is ε, we also write $\tau_{\perp i,y}$ resp. $\tau_{xi,\perp}$ instead of $\tau_{\varepsilon i,y}$ resp. $\tau_{xi,\varepsilon}$. For state transformations τ_1, τ_2, we define $\tau_1 \circ \tau_2$ as the composition of τ_1, τ_2, that is $\tau_1 \circ \tau_2(q) = \tau_2(\tau_1(q))$ if $\tau_1(q)$ is defined.

The following theorem shows that is it decidable whether a relation has a uniformization by a path-recognizable function.

Theorem 9 *It is decidable whether a D↓TA-recognizable relation with D↓TA-recognizable domain can be uniformized by a path-recognizable function.*

Proof. Let $R \subseteq T_\Sigma \times T_\Gamma$ be a top-down deterministic tree-automatic relation with top-down deterministic domain. For such a relation, we show that the existence of a uniformizer that realizes a path-recognizable function is a regular property over infinite trees. The key ingredient of this proof is an MSO-formula over infinite trees that asserts the existence of such a uniformizer. We can then use the equivalence between regular and MSO-definable languages over infinite trees.

The proof is split in two parts. Recall, to describe a path-recognizable function, we have to reason about labeled paths through input trees and corresponding output trees. Hence, first, we define a regular infinite tree that is suitable to specify (via MSO) a set of deterministically readable labeled paths. Secondly, we state an MSO-formula that is satisfiable by this infinite tree if, and only if, there exists a finite state uniformizer that realizes a path-recognizable function.

For the first part, we construct a regular infinite tree by defining a finite graph and using its unfolding to obtain a regular infinite tree, see e.g. [14]. Intuitively, to be able to specify labeled paths in this tree, the node labels alternate between directions from dir_Σ and possible input symbols from Σ (if the input has not yet ended). A node labeled by $f \in \Sigma_n$, $n > 0$ has exactly n children, where the ith child is labeled by i. A direction labeled node has an f-labeled child for every $f \in \Sigma$ (or $\perp$ if it occurs below a leaf-labeled node). For a node labeled by a leaf symbol $a \in \Sigma_0$, the nodes below form a string with labels alternating between 1 and $\perp$. As a special case, the root is labeled by 0.

Formally, let $\Sigma = \bigcup_{i=0}^m \Sigma_i$ and let $\mathcal{B} = (Q_\mathcal{B}, \Sigma, q_0^\mathcal{B}, \Delta_\mathcal{B})$ be a D↓TA that recognizes $dom(R)$. We want to construct the desired infinite tree w.r.t. the domain of the relation. Thus, we define the graph $\mathcal{G} = (V, E)$ with vertices

$V \subseteq (Q_{\mathcal{B}} \,\dot\cup \{p_\perp\}) \times (\Sigma_\perp \cup \{0\} \cup \mathrm{dir}_\Sigma)$ and a set of edges E that include the information of the runs of $\mathcal{B}$, and is defined by

$$\begin{aligned} &\big\{\big((q_0^{\mathcal{B}},0),(q_0^{\mathcal{B}},f)\big) \mid f \in \Sigma\big\} \\ \cup\ &\big\{\big((p,i),(p,f)\big) \mid p \in Q_{\mathcal{B}}, i \in \mathrm{dir}_\Sigma, f \in \Sigma\big\} \\ \cup\ &\big\{\big((p,f),(p_j,j)\big) \mid (p,f,p_1,\dots,p_k) \in \Delta_{\mathcal{B}}, f \in \Sigma_k, k>0, j \in \{1,\dots,k\}\big\} \\ \cup\ &\big\{\big((p,a),(p_\perp,1)\big) \mid (p,a) \in \Delta_{\mathcal{B}}, a \in \Sigma_0\big\} \\ \cup\ &\big\{\big((p_\perp,1),(p_\perp,\perp)\big),\big((p_\perp,\perp),(p_\perp,1)\big)\big\}\,. \end{aligned}$$

Let $(q_0^{\mathcal{B}},0)$ be the initial vertex of $\mathcal{G}$ and let $t_{\mathcal{G}}$ denote the unfolding of $\mathcal{G}$ from this initial vertex.

Now, we give an MSO-formula that is satisfiable by $t_{\mathcal{G}}$ if, and only if, R can be uniformized by a finite state transducer that realizes a path-recognizable function. Let T_V^{Inf} denote the set of ranked infinite trees over V, the rank of a symbol $v \in V$ is the outdegree of v in $\mathcal{G}$. As usual, an infinite tree $t \in T_V^{\mathrm{Inf}}$ corresponds to the logical structure $\underline{t} = (dom_t, S_1^t, \dots, S_m^t, S^t, (P_v^t)_{v\in V})$, where $S_i^t = i$th successor relation on dom_t, $S^t := \bigcup_{i=0}^m S_i^t$, and $P_v^t := \{u \in dom_t \mid val_t(u) = v\}$. To begin with, we introduce some simple auxiliary formulas:

$$P_\lambda(x) := \bigvee_{p \in (Q_{\mathcal{B}} \,\dot\cup \{p_\perp\})} P_{(p,\lambda)}(x) \text{ for all } \lambda \in (\Sigma_\perp \cup \{0\} \cup \mathrm{dir}_\Sigma), \text{ and } P_{\mathrm{dir}}(x) := \bigvee_{i=0}^{m} P_i(x).$$

Also, formulas for set inclusion, prefix closed sets, and partition are needed:

$$X \subseteq Y := \forall x\big(X(x) \to Y(x)\big),\ \phi_{\mathrm{pre}}(X) := \forall x \forall y\big(X(y) \wedge S(x,y) \to X(x)\big),\ \text{and}$$

$$\phi_{\mathrm{part}}(X_1,\dots,X_n) := \forall x \left(\bigvee_{i=1}^{n} X_i(x) \wedge \bigwedge_{i \neq j \wedge i,j \in \{1,\dots,n\}} \neg(X_i(x) \wedge X_j(x)) \right).$$

We give a formula that specifies an infinite labeled path. If such a labeled path contains a $\perp$-labeled node, then this labeled path describes a path through a finite input tree.

$$\mathrm{Path}(X) := \exists x\big(P_0(x) \wedge X(x)\big) \wedge \forall x\big(X(x) \to \exists y\,(S(x,y) \wedge X(y) \wedge \forall z\,(S(x,z) \wedge X(z) \to y = z))\big).$$

Furthermore, we want to specify a formula that is satisfied by a set of labeled paths such that a transducer is able to deterministically read (along a labeled path from this set) through every possible input tree. This is satisfied if for a Σ-labeled node that is part of the set, exactly one direction-labeled direct successor is specified, and in addition, for a direction-labeled node that is part of the set, every Σ-labeled direct successor is also part of the set, ensuring that a transducer can react to every input symbol.

$$\begin{aligned} \mathrm{PathSet}(X) := \exists x\big(P_0(x) \wedge X(x)\big) \wedge \forall x\big(X(x) \wedge P_{\mathrm{dir}}(x) \to \forall y(S(x,y) \to X(y))\big) \wedge \\ \forall x\big(X(x) \wedge P_\Sigma(x) \to \exists y\,(S(x,y) \wedge X(y) \wedge \forall z\,(S(x,z) \wedge X(z) \to y = z))\big). \end{aligned}$$

Ultimately, all formulas are designed to be evaluated in $\underline{t_\mathcal{G}}$. Thus, if $(\underline{t_\mathcal{G}}, X) \models \text{Path}(X)$ resp. $(\underline{t_\mathcal{G}}, X) \models \text{PathSet}(X)$, then X indeed describes a labeled path resp. a set of labeled paths that can be part of an input tree from the domain of the relation, because $t_\mathcal{G}$ is designed w.r.t. the domain of the relation.

Towards our desired formula, given a labeled path and a prefix of this path, we want to express that there exists an output labeling of the given prefix such that there is an accepting (partial) run of $\mathcal{A}$ on the combination of input labels and output labels on this path. Additionally, since we want to obtain a path-recognizable function in the end, we require that the partial run can be extend to a successful run for each input tree that contains this labeled path, i.e., we require that the run only depends on the given labeled path. Recall, we already have a formal notion for this. Let $\mathcal{A} = (Q_\mathcal{A}, \Sigma_\perp \times \Gamma_\perp, q_0^\mathcal{A}, \Delta_\mathcal{A})$ be a D↓TA that recognizes R. For $x \in \text{Path}_\Sigma$, $y \in \text{Path}_\Gamma$ and $i \in \text{dir}_\Sigma$ such that $x \otimes y$ is defined, i.e., $path(x) \sqsubseteq path(y)$ or $path(y) \sqsubseteq path(x)$, the function $\tau_{xi,y}$ is defined for a state q of $\mathcal{A}$, if there is a partial run ρ on $x \otimes y$ of $\mathcal{A}_q$ with $\mathcal{A} : q \xrightarrow{x\otimes y}_i q'$ and there exists a fixed partial output tree $s \in S_\Gamma^{yi\circ}$ such that ρ can be extend to a successful run of $\mathcal{A}_q$ on each input tree $t \in T_\Sigma^x$ together with an output tree $s \cdot t' \in T_\Gamma^y$, that is, an output tree whose first part is always s.

The following formula expresses the requirements stated above. Let $Q_\mathcal{A} = \{q_1, \ldots, q_n\}$, $q_1 = q_0^\mathcal{A}$, and $\Gamma = \{g_1, \ldots, g_\ell\}$, we then write

$$
\begin{aligned}
R_{\text{Path}}(X, Y) := {} & \text{Path}(X) \wedge Y \subseteq X \wedge \phi_{\text{pre}}(Y) \wedge \\
& \exists Y_{g_1} \ldots \exists Y_{g_\ell} \exists Y_\perp \exists X_{q_1} \ldots \exists X_{q_n} \Bigg(\phi_{\text{part}}(Y_{g_1}, \ldots, Y_{g_\ell}, Y_\perp) \wedge \phi_{\text{part}}(X_1, \ldots, X_n) \wedge \\
& \quad \forall y \big(Y(y) \to \neg Y_\perp(y)\big) \wedge \exists x \exists y \big(P_0(x) \wedge S(x,y) \wedge X(x) \wedge X_{q_1}(y)\big) \wedge \\
& \quad \forall x \forall d \forall y \Big[S(x,d) \wedge S(d,y) \wedge P_{\text{dir}}(d) \wedge X(x) \wedge X(d) \wedge X(y) \wedge \neg\big(P_\perp(y) \wedge Y_\perp(y)\big) \to \\
& \qquad \Big(\bigvee_{\substack{(q,(f,g),q_1,\ldots,q_j) \in \Delta_\mathcal{A}, \\ \tau_{fi,g}(q) = q_i}} X_q(x) \wedge P_f(x) \wedge Y_g(x) \wedge P_i(d) \wedge X_{q_i}(y) \Big) \Big] \wedge \\
& \quad \exists x \Big[X(x) \wedge \neg P_{\text{dir}}(x) \to \Big(\bigvee_{(q,(f,g)) \in \Delta_\mathcal{A}} X_q(x) \wedge P_f(x) \wedge Y_g(x) \Big) \Big] \Bigg).
\end{aligned}
$$

The above formula describes the existence of a successful (partial) run of $\mathcal{A}$ on $\pi \otimes o$ with $\tau_{\pi,o}(q_1) = Acc$, where π is the labeled path considered in the input tree (π corresponds to the valuation of X) and $path(o)$ is the path considered in the output tree ($path(o)$ corresponds to the valuation of Y, the formula expresses the existence of suitable output labels for o). For each state $q \in Q_\mathcal{A}$, X_q states that on the ith position in π, the automaton is in state q iff $X_q(i)$; for each output symbol $g \in \Gamma$, Y_g states that the label of the ith position in o is g iff $Y_g(i)$. The formula $\exists x \exists y \big(P_0(x) \wedge S(x,y) \wedge X(x) \wedge X_{q_1}(y)\big)$ states that the run ρ begins in the initial state q_1 of $\mathcal{A}$; the formula that occurs below in $R_{\text{Path}}(X, Y)$ states thats that for every pair of two successive positions (y is the ith child of x iff $P_i(d)$) in the run ρ there is a transition permissible by $\Delta_\mathcal{A}$ and additionally

$\tau_{fi,g}(\rho(x)) = \rho(y)$ if f is the input label of x and g is the output label; the last formula $\exists x\Big[X(x) \wedge \neg P_{\mathrm{dir}}(x) \to \Big(\bigvee_{(q,(f,g))\in\Delta_{\mathcal{A}}} X_q(x) \wedge P_f(x) \wedge Y_g(x)\Big)\Big]$ states that at the last position of $\pi \otimes o$, i.e., at the leaf, a transition is applicable which makes the run accepting.

We need one more auxiliary formula. To obtain a finite state transducer that realizes a path-recognizable function, we require that only finitely many different output trees are needed. If only finitely many different output trees are needed, then the length of the output sequences that are mapped to the relevant labeled paths is bounded, that is, the set of needed output sequences is finite. The next formula is satisfied by a finite set that describes prefixes of labeled paths. If all needed output sequences stay inside the positions specified by the set, then this implies that finitely many different output sequences suffice.

$$\phi_{\mathrm{pre,fin}}(X) := \phi_{\mathrm{pre}}(X) \wedge \forall X_1\Big(\mathrm{Path}(X_1) \to \exists x\Big[X_1(x) \wedge X(x) \wedge \neg\big(\exists y\big[S(x,y) \wedge X_1(y) \wedge X(y)\big]\big)\Big]\Big).$$

Now, we are ready to state the desired formula. The formula describes that there is a set of deterministically readable labeled paths such that for each labeled path from this set that describes a path through a finite input tree there is a matching output and additionally the set of used matching outputs (along the relevant paths) is finite.

$$\begin{aligned}\phi_{\mathrm{unif}} :=& \exists X \exists Y\Big(\mathrm{PathSet}(X) \wedge Y \subseteq X \wedge \phi_{\mathrm{pre,fin}}(Y) \wedge \\ & \forall X_1\big[X_1 \subseteq X \wedge \mathrm{Path}(X_1) \wedge \exists x\big(X_1(x) \wedge P_\perp(x)\big) \to \exists Y_1\big(Y_1 \subseteq Y \wedge R_{\mathrm{Path}}(X_1, Y_1)\big)\big]\Big).\end{aligned}$$

Since the equivalence between MSO-definable tree languages and regular tree languages is effective, we can construct an automaton $\mathcal{A}_{\phi_{\mathrm{unif}}}$ such that $T(\mathcal{A}_{\phi_{\mathrm{unif}}})$ is equivalent to $T(\phi_{\mathrm{unif}}) := \{t \in T_V^{\mathrm{Inf}} \mid \underline{t} \models \phi_{\mathrm{unif}}\}$.

Finally, we claim $t_{\mathcal{G}} \in T(\mathcal{A}_{\phi_{\mathrm{unif}}})$ if, and only if, R is uniformizable by a finite state transducer that realizes a path-recognizable function. If R is uniformized by a path-recognizable function, then there exists an assignment of the variables such that $\underline{t_{\mathcal{G}}} \models \phi_{\mathrm{unif}}$. From a D↓TT $\mathcal{T}$ that uniformizes R and realizes a path-recognizable function we can obtain a valuation as follows. Recall that a position in $t_{\mathcal{G}}$ can be either interpreted as a labeled path, or as a path resp. a position in a tree if we disregard the labels. Concerning the valuation of X, a position is included in the set X if it describes a (prefix of a) labeled path that the uniformizer reads from the root to a leaf in an input tree in order to produce an output tree. Generally, such a uniformizer specifies a set of finite labeled paths of arbitrary length. However, since ϕ_{unif}, more specifically its subformula $\mathrm{PathSet}(X)$, requires X to describe a set of infinite labeled paths, for each finite labeled path π that the uniformizer reads, X also includes the positions that describe its infinite extension $\pi(1\perp)^\omega$. Then X describes the desired set of infinite labeled paths. Concerning the valuation of Y, we have to consider the output trees that $\mathcal{T}$ produces. These are only finitely many different trees, say $t_1, \ldots, t_k$. We choose Y as the subset of X such that a position from X

is included in Y if there is (at least) one t_i such that this position describes a position that occurs in dom_{t_i}. Since the set $dom_{t_1} \cup \cdots \cup dom_{t_k}$ is finite, also Y is finite. Then $\underline{t_{\mathcal{G}}} \models \phi_{\text{unif}}$, because for each finite labeled path from the set X the set Y describes a prefix of this path such that there is an output sequence that can be mapped to this prefix (obtained from one of the t_is) such that $\mathcal{A}$ has a successful (partial) run on this input and output combination.

If $t_{\mathcal{G}} \in T(\mathcal{A}_{\phi_{\text{unif}}})$, then there exists a valuation of the variables such that a regular ([15], see also [14]) deterministically readable set of labeled paths can be obtained that captures every input tree. Since this set is regular, it is recognizable by a finite state transducer. Additionally, the valuation of Y yields a global bound such that the size of each output mapped to a relevant path through an input tree remains inside this bound. Consequently, only finitely many different output trees are needed. Hence, there exists a finite state uniformizer that realizes a path-recognizable function, such a uniformizer can e.g., be constructed in the following way. Clearly, a deterministic finite state top-down tree transducer that reads the relevant path in an input tree can be obtained from a regular set of labeled paths. Also, a global bound of the length of needed output sequences on the relevant paths is known. Since there are only finitely many output sequences that remain inside the bound, the transducer can simulate for each such output sequence the partial run of $\mathcal{A}$ on the read input sequence together with this output sequence in its state space. Eventually, when a leaf is reached, at least one of the simulated partial runs is accepting. The transducer then produces an output tree matching to the (chosen) accepting run. □

We have seen that it is decidable whether a specification has a uniformization by a path-recognizable function. Given a specification, our goal is to show that there exists a computable bound on the output delay with the following property: If it is necessary for a transducer to introduce delay that exceeds the bound in order to satisfy the specification, then either the uniformization task is impossible, or the remaining specification has a uniformization by a path-recognizable function, which is decidable by Theorem 9.

Towards the choice of the bound, we first introduce profiles for labeled path segments. We define the profile of a labeled path segment xi to be the set that contains all possible state transformations induced by x together with some y of same or smaller length. Formally, let $x \in \text{Path}_\Sigma$ and $i \in \text{dir}_\Sigma$, we define the *profile* of xi to be $P_{xi} = (P_{xi,=}, P_{xi,<}, P_{xi,\varepsilon})$ with

$$P_{xi,=} := \{\tau_{xi,y} \mid |y| = |x|\},\ P_{xi,<} := \{\tau_{xi,y} \mid y \neq \varepsilon \text{ and } |y| < |x|\}, \text{ and}$$
$$P_{xi,\varepsilon} := \{\tau_{xi,y} \mid y = \varepsilon\}.$$

From $P_{x_1 i_1}$ and $P_{x_2 i_2}$ the profile $P_{x_1 i_1 x_2 i_2}$ is uniquely determined, i.e.,

$$P_{x_1 i_1 x_2 i_2,=} := \{\tau_1 \circ \tau_2 \mid \tau_1 \in P_{x_1 i_1,=}, \tau_2 \in P_{x_2 i_2,=}\},$$
$$P_{x_1 i_1 y i,<} := \{\tau_1 \circ \tau_2 \mid \tau_1 \in P_{x_1 i_1,=}, \tau_2 \in P_{x_2 i_2,<}\} \cup \{\tau_1 \circ \tau_{x_2 i_2,\varepsilon} \mid \tau_1 \in P_{x_1 i_1,<}\},$$
$$\text{and } P_{x_1 i_1 x_2 i_2,\varepsilon} := \{\tau_{x_1 i_1,\varepsilon} \circ \tau_{x_2 i_2,\varepsilon}\}.$$

Thus, the concatenation of $P_{x_1 i_1}$, $P_{x_2 i_2}$ is naturally defined by $P_{x_1 i_1} \cdot P_{x_2 i_2} = P_{x_1 i_1 x_2 i_2}$. A segment $xi \in (\Sigma\text{dir}_\Sigma)^*\text{dir}_\Sigma$ of a labeled path is called *idempotent* if $P_{xi} = P_{xixi}$.

As a consequence of Ramsey's Theorem [16], we obtain the next remark.

Remark 10 *There exists a bound $K \in \mathbb{N}$ such that each labeled path $\pi \in \mathrm{Path}_\Sigma$ with $||\pi|| \geq K$ contains an idempotent factor.*

Proof. Ramsey's Theorem yields that for any number of colors c and any number r, there exists a number $K \in \mathbb{N}$ such that if the edges of a complete graph with at least K vertices are colored with c colors, then the graph must contain a complete subgraph with r vertices such that all edges have the same color, c.f. [17].

Let $\pi \in \mathrm{Path}_\Sigma$ with the factorization $\pi = f_1 j_1 \dots j_{n-1} f_n$, $f_1, \dots, f_n \in \Sigma$ and $j_1, \dots, j_{n-1} \in \mathrm{dir}_\Sigma$. Consider the complete graph $G = (V, E, col)$ with edge-coloring $col : E \to Cols$, where $V := \{1, \dots, n\}$, $E := V \times V$, $Cols$ is the finite set of profiles and $col(e) := P_{f_i j_i \dots f_k j_k}$ if $e = (i,k)$ for all $e \in E$. If there exist $i, j, k \in \mathbb{N}$ with $i < j < k \leq n$ such that the edges (i,k), (i,j) and (j,k) have the same color, i.e., the respective profiles are the same, then π has a factorization that contains an idempotent factor.

As a consequence of Ramsey's Theorem, for $r = 3$ and $c = |Cols|$, if $n \geq K$, then π must contain an idempotent factor. □

We introduce some additional notation on how to split a tree in parts and how to repeat a part of a tree that contains an idempotent factor. For a special tree $s \in S_\Sigma$, we inductively define the special tree $s^n \in S_\Sigma$ by $s^n := s^{n-1} \cdot s$ and $s^0 := \circ$ for $n \in \mathbb{N}$. Let $x, y \in \mathrm{Path}_\Sigma$, $i, j \in \mathrm{dir}_\Sigma$, and $path(x)i = u$, $path(y)j = v$. For a tree $t \in T_\Sigma^{xiy}$, we introduce shorthand notations $t_{[:u]}$, $t_{[u:uv]}$, and $t_{[uv:]}$ to denote $t[\circ/u]$, $t[\circ/uv]|_u$, and $t|_{uv}$, respectively. Note that $t = t_{[:u]} \cdot t_{[u:uv]} \cdot t_{[uv:]}$. Furthermore, let $y \neq \varepsilon$ and yj be an idempotent factor, we fix $t^n_{(u,v)}$ to be the tree that results from *repeating the idempotent factor* n times. More formally, we define

$$t^n_{(u,v)} := t_{[:u]} \cdot t^n_{[u:uv]} \cdot t_{[uv:]} \text{ for } n \in \mathbb{N}.$$

If it is clear from the context to which idempotent factor we refer, then we leave out the subscript.

The next lemma formally establishes the connection between long output delay and path-recognizable functions. Basically, the lemma states that if there exists a uniformization by a D↓TT such that an idempotent path segment can be repeated any number of times and the length of the output on the repetition is bounded, i.e., the output delay is unbounded, then there also exists a uniformization by a path-recognizable function.

Lemma 11 *Given a D↓TA-recognizable relation R with D↓TA-recognizable domain, $x, y \in \mathrm{Path}_\Sigma$, $i, j \in \mathrm{dir}_\Sigma$ with $path(x)i = u$, $path(y)j = v$, $y \neq \varepsilon$ and yj idempotent. For a tree $t \in T_\Sigma^{xiy}$ let t^n denote the tree obtained from t, where the idempotent factor yj is repeated n times.*

If R^{xiy} is uniformized by a D↓TT $\mathcal{T}$ such that for each $t \in T_\Sigma^{xiy}$ and each $n \in \mathbb{N}$ there is a reachable configuration (t^n, t^n_o, φ_n) of $\mathcal{T}$ on t^n such that

1. *$\mathcal{T}$ reads the path uv^n in t^n, i.e., there is a node $\alpha_n \in dom_{t^n_o}$ with $\varphi_n(\alpha_n) = uv^n$ and*

2. the output produced on uv^n does not exceed the node u in t^n, i.e., $\alpha_n \sqsubseteq u$,

then R^{xiy} can be uniformized by a path-recognizable function.

Proof. We use the characterization from the proof of Theorem 9. To show the statement of this lemma it suffices to show that the MSO-sentence ϕ_{unif} constructed from a D↓TA $\mathcal{A}$ for R^{xiy} is satisfied by $\underline{t_{\mathcal{G}}}$ constructed from a D↓TA $\mathcal{B}$ for its domain. We state the sentence again:

$$\phi_{\text{unif}} := \exists X \exists Y \Big(\text{PathSet}(X) \wedge Y \subseteq X \wedge \phi_{\text{pre,fin}}(Y) \wedge$$
$$\forall X_1 \big[X_1 \subseteq X \wedge \text{Path}(X_1) \wedge \exists x \big(X_1(x) \wedge P_\perp(x) \big) \rightarrow \exists Y_1 \big(Y_1 \subseteq Y \wedge R_{\text{Path}}(X_1, Y_1) \big) \big] \Big).$$

We have to provide a suitable valuation for X and Y. Recall, the set X has to describe a regular set of labeled paths such that a transducer is able to deterministically read through every possible input tree. The set Y has to be a finite set representing the needed output trees, that is, for a labeled path π described by X there exists a suitable output tree $t_o \in T_\Gamma$ described by Y such that $(t, t_o) \in R^{xiy}$ for each $t \in T_\Sigma^\pi$.

We proceed as follows. For each input tree $t \in dom(R^{xiy})$, we choose a labeled path π from the root to a leaf that should be described by X and choose a matching output tree t_o that should be described by Y. Then, we prove that the resulting set X describes a deterministically readable set of labeled paths and that the resulting set Y is finite.

We use the following observation. For each $t \in T_\Sigma^{xiy}$, recall t is of the form $t_{[:u]} \cdot t_{[u:uv]} \cdot t_{[uv:]}$, holds that if the idempotent factor yj is repeated often enough, say n times, then $||out_{\mathcal{T}}(t_{[:u]} \cdot t^n_{[u:uv]} \cdot t_{[uv:]}, uv^n)|| < |uv^n|$. This follows from the assumption that the output produced on $t_{[:u]} \cdot t^n_{[u:uv]}$ does not exceed u. So if n is large enough, the additional output produced on $t_{[uv:]}$ does not exceed uv^n. In the following, this allows us to pick an output tree t_o whose height only depends on xiy such that $(t, t_o) \in R^{xiy}$ for each input tree $t \in dom(R^{xiy})$. Another needed (technical) observation is, there exists a state s of $\mathcal{T}$ and integer $m_1 < m_2$ such that s is reached at the node uv^{m_1} and again at uv^{m_2} in a tree t^{m_2} for each $t \in T_\Sigma^{xiy}$.

Now, consider an arbitrary input tree $t \in dom(R^{xiy})$, we can pick some k such that $||out_{\mathcal{T}}(t^k, uv^k)|| < |uv^k|$ and in a computation of $\mathcal{T}$ on t^k the uniformizer is in state s at the node uv^k. Then, the set X includes the description of a labeled path $\pi \in \text{Path}_\Sigma$ of the form $xiyjz$ with $path(z) = w$ if $t \in T_\Sigma^\pi$ and the following holds. The labeled path $xi(yj)^k z$ must be the unique labeled path from the root to a leaf in t^k that the uniformizer reads in order to produce output mapped to $xi(yj)^{k-1}y$, that is, output mapped to the repetitions of the idempotent factor. Formally, this means $out_{\mathcal{T}}(t^k, uv^k w) = o$ for some $o \in \text{Path}_\Gamma$ and $path(o) \sqsubseteq uv^k$. Consider the factorization of $o = o_1 i o_2$ such that $||o_1|| = ||x||$, then $||o_2|| \leq ||(yj)^{k-1}y||$, because the choice of k ensures $||out_{\mathcal{T}}(t^k, uv^k)|| < |uv^k|$.

We now pick some $t_o \in T_\Gamma$ such that $(t, t_o) \in R^{xiy}$. Let the specification be recognized by some D↓TA $\mathcal{A}$. Consider the deterministic successful run of $\mathcal{A}$ on

$t^k \otimes \mathcal{T}(t^k)$ along uv^kw which looks as follows:

$$\mathcal{A} : q_0 \xrightarrow{x \otimes o_1}_i q_1 \xrightarrow{(yj)^{k-1} y \otimes o_2}_j q_2 \xrightarrow{z \otimes \varepsilon} Acc.$$

Note that since $\mathcal{A}$ is deterministic, for each tree $\tilde{t}$ from $T_\Sigma^{xi(yj)^k z}$ the run of $\mathcal{A}$ on $\tilde{t}^k \otimes \mathcal{T}(\tilde{t}^k)$ has the same form along $xi(yj)^k z$. Hence, $\tau_{xi,o_1}(q_0) = q_1$, $\tau_{(yj)^k,o_2}(q_1) = q_2$ and $\tau_{z,\varepsilon}(q_2) = Acc$. In words, there exists a fixed output tree $\tilde{t}_o \in T_\Gamma^{o_1 i o_2}$ such that $(\tilde{t}, \tilde{t}_o) \in R^{xiy}$ for each tree $\tilde{t}$ from $T_\Sigma^{xi(yj)^k z}$. Also, since yj is idempotent, there is some $o_3 \in \mathrm{Path}_\Gamma$ such that $\tau_{(yj)^k,o_2} = \tau_{yj,o_3}$ with $||o_3|| \leq ||y||$. Let t_o denote an output tree corresponding to $\tau_{xiyjz,o_1 i o_3}$. Thus, a run of $\mathcal{A}$ on $t \otimes t_o$ along uvw has the following form:

$$\mathcal{A} : q_0 \xrightarrow{x \otimes o_1}_i q_1 \xrightarrow{y \otimes o_3}_j q_2 \xrightarrow{z \otimes \varepsilon} Acc,$$

i.e., $(t, t_o) \in R^{xiy}$. Hence, we chose the description of t_o to be included in the set Y. Consequently, for a valuation of $X_1 \subseteq X$ corresponding to $\pi = xiyjz$ and a valuation of $Y_1 \subseteq Y$ corresponding to $o_1 i o_3$ the MSO-formula $R_{\mathrm{Path}}(X_1, Y_1)$ describing the existence of a successful (partial) run of $\mathcal{A}$ on $\pi \otimes o_1 i o_3$ with $\tau_{\pi,o_1 i o_3}(q_0) = Acc$ is satisfied.

Now we argue why X describes a regular set of deterministically readable paths. Recall the conditions that a labeled path π for a tree $t \in T_\Sigma^\pi$ fulfills when it is described by X. The labeled path π is of the form $xiyjz$ for some z and $xi(yj)^k z$ is the labeled path that is read by $\mathcal{T}$ in t^k in order to determine the output on $xi(yj)^{k-1}y$ for some suitable choice of k such that the segment z is read from state s of $\mathcal{T}$. Thus, the form of each labeled path in X is determined by $\mathcal{T}_s$. Hence, X is a subset of the labeled paths read by $\mathcal{T}_s$ (prefixed by $xiyj$). A finite deterministic automaton that recognizes the labeled paths read by $\mathcal{T}_s$ which determine the output mapped to the repetitions of the idempotent factor can be constructed from $\mathcal{T}_s$. This implies that X describes a regular set of labeled paths deterministically readable by a transducer.

We have shown that for each finite labeled path π from the root to a leaf that is described by X there exists a suitable output tree t_o, furthermore the size of such a t_o is bounded by the length of xiy, thus only finitely many different t_os are needed. Hence Y is a finite set representing the needed t_os. This concludes the proof, we have seen that a suitable valuation for X and Y exists such that the MSO-sentence ϕ_{unif} is satisfied, i.e., R^{xiy} is uniformizable by a path-recognizable function. □

As we have seen, if a transducer that uniformizes a relation introduces long output delay, then the relation can also be uniformized by a path-recognizable function.

Before we present the decision procedure, we introduce a complexity measure for D↓TTs w.r.t. the lookahead that the transducer introduces. This will help us reason about the behavior of a uniformizer with minimal complexity. The idea is to measure the complexity of a uniformizer by counting the introduced lookahead. Since it is not necessary for a uniformizer to have divergent input and output paths, we will only consider lookahead-paths where input and produced output overlap. Also we do not consider lookahead-paths that could be part of a path-recognizable function, because this is generally an infinite set of paths.

For a D↓TA-recognizable relation R, a D↓TT $\mathcal{T}$ that uniformizes R, and a limit $l \in \mathbb{N}$, we define the *lookahead-path-language* $\mathrm{LPL}_{\mathcal{T}}(R,l)$ such that $\pi \in \mathrm{Path}_\Sigma$ is included if the following conditions hold:

- There is a $t \in T_\Sigma^\pi \cap dom(R)$ with $(t, q_0^{\mathcal{T}}, \varphi_0) \to_{\mathcal{T}} (t, t_1, \varphi_1) \to_{\mathcal{T}} \cdots \to_{\mathcal{T}} (t, t_n, \varphi_n)$ such that there is $u_i \in D_{t_i} \cap dom_\pi$ with $u_i \sqsubseteq \varphi_i(u_i) \sqsubseteq path(\pi)$ and $|u_i| < l$ and $\varphi_n(u_n) = path(\pi)$ and it occurs output delay w.r.t. u_i in (t, t_i, φ_i) for all $i \in \{1, \dots, n\}$, and
- R^π is not uniformizable by a path-recognizable function.

Note, $\mathrm{LPL}_{\mathcal{T}}(R,l)$ is a prefix-closed set. Furthermore, we define $value(\mathrm{LPL}_{\mathcal{T}}(R,l))$ to be the sum of all lengths of all labeled paths, i.e., $\sum_{\pi \in \mathrm{LPL}_{\mathcal{T}}(R,l)} ||\pi||$.

Lemma 12 *The set* $\mathrm{LPL}_{\mathcal{T}}(R,l)$ *is finite.*

Proof. We prove this statement by contradiction. Assume $\mathrm{LPL}_{\mathcal{T}}(R,l)$ is infinite. Then $\mathrm{LPL}_{\mathcal{T}}(R,l)$ contains labeled paths of arbitrary length. We pick some $\pi \in \mathrm{LPL}_{\mathcal{T}}(R,l)$ such that $||\pi|| > l + l \cdot K|Q_{\mathcal{T}}|$ with K from Remark 10. Now, we prove that R^π is uniformizable by a path-recognizable function. This is a contradiction to $\pi \in \mathrm{LPL}_{\mathcal{T}}(R,l)$.

Pick any $t \in T_\Sigma^\pi \cap dom(R)$ and let $c_0 = (t, q_0^{\mathcal{T}}, \varphi_0)$. The configuration sequence witnessing membership of π in $\mathrm{LPL}_{\mathcal{T}}(R,l)$ has length $> l + l \cdot K|Q_{\mathcal{T}}|$. For each $j \in \{0, \dots, l-1\}$, we can thus pick a subsequence of configurations $c_1 = (t, t_1, \varphi_1)$, $c_2 = (t, t_2, \varphi_2)$, …, $c_K = (t, t_K, \varphi_K)$ with $c_0 \to_{\mathcal{T}}^* c_1 \to_{\mathcal{T}}^* \cdots \to_{\mathcal{T}}^* c_K$ such that there is $s \in Q_{\mathcal{T}}$ and $u_i \in D_{t_i} \cap dom_\pi$ with $val_{t_i}(u_i) = s$ for all $i \in \{1, \dots, K\}$ and furthermore $l + j \cdot K|Q_{\mathcal{T}}| < |\varphi_1(u_1)|$ and $|\varphi_K(u_K)| \leq l + (j+1) \cdot K|Q_{\mathcal{T}}|$.

Let π_j denote the jth segment of length $K|Q_{\mathcal{T}}|$ of π starting after the first l letters of π. Together with Remark 10 it follows that each such segment contains an idempotent factor yj_2 such that $\pi = xj_1yj_2z$ with $path(x_1)j_1 = \varphi_m(u_m)$ and $path(xj_1y)j_2 = \varphi_n(u_n)$ for some $m < n \leq K$ w.r.t. a suitable subsequence. There are at least l such segments in π, because $||\pi|| > l + l \cdot K|Q_{\mathcal{T}}|$.

Since $\pi \in \mathrm{LPL}_{\mathcal{T}}(R,l)$, there is at least one segment π_j such that $\mathcal{T}$ does not produce output while reading the idempotent factor in π_j. Otherwise, $\mathcal{T}$ would produce at least l output symbols while reading π, which is a contradiction to $\pi \in \mathrm{LPL}_{\mathcal{T}}(R,l)$. Consider a subsequence that yields an idempotent factor yj_2 such that $\pi = xj_1yj_2z$ with $path(x_1)j_1 = \varphi_m(u_m)$ and $path(xj_1y)j_2 = \varphi_n(u_n)$ for some $m < n \leq K$ such that $\mathcal{T}$ produces no output while reading yj_2. Let $\varphi_m(u_m) = v_m$ and $\varphi_n(u_n) = v_n$. Then, since $||out_{\mathcal{T}}(t_{[:v_m]} \cdot t^k_{[v_m:v_n]}, path(\pi))|| < l$ for all $k \in \mathbb{N}$, Lemma 11 implies that R^π can be uniformized by a path-recognizable function. □

From the above lemma it follows directly that $value(\mathrm{LPL}_{\mathcal{T}}(R,l)) \in \mathbb{N}$. This allows us to compare uniformizers by comparing their values. We say $\mathcal{T}$ has *minimal complexity w.r.t. l* if $value(\mathrm{LPL}_{\mathcal{T}}(R,l)) \leq value(\mathrm{LPL}_{\mathcal{U}}(R,l))$ for every D↓TT $\mathcal{U}$ that uniformizes R. The next lemma gives a bound on the length of the elements in the lookahead-path language of a uniformizer with minimal complexity. The idea is that a uniformizer with minimal complexity does not further delay the output if the lookahead-path contains an idempotent factor (unless the relation has to be uniformized by a path-recognizable function). We use this statement in the proof of Lemma 14 for the construction of a winning strategy from a uniformizer. There we need to work with a uniformizer that is not of absolute minimal complexity, but only among those uniformizers that have a specific path in their lookahead-path language. For this reason, we need to introduce the extra path γ in the statement of Lemma 13. Let K be as in Remark 10.

Lemma 13 *Let R be a D↓TA-recognizable relation with D↓TA-recognizable domain and let $\mathcal{T}$ be a D↓TT that uniformizes R. Given some limit $l \in \mathbb{N}$ and given some $\gamma \in \mathrm{LPL}_{\mathcal{T}}(R,l)$ with $||\gamma|| \leq l$, if $\mathcal{T}$ has minimal complexity w.r.t. l in comparison to every other D↓TT $\mathcal{U}$ with $\gamma \in \mathrm{LPL}_{\mathcal{U}}(R,l)$ that uniformizes R, then the length of a lookahead-path from $\mathrm{LPL}_{\mathcal{T}}(R,l)$ is bounded by $l+K$.*

Proof. Proof by contradiction. Assume there is a labeled path $\pi \in \mathrm{LPL}_{\mathcal{T}}(R,l)$ such that $||\pi|| > l + K$, then by Remark 10 there exists a factorization of π into $xiyjz$ such that yj is an idempotent factor and $l \leq ||x||$. Furthermore, it follows from Remark 10 that yj can be split into y_1dy_2j such that y_1d, y_2j and yj have the same profile, i.e., $P_{y_1d} = P_{y_2j} = P_{yj}$. Towards a contradiction, first, we show that there is a uniformizer $\mathcal{U}$ with $\gamma \in \mathrm{LPL}_{\mathcal{U}}(R,l)$ such that xiy_1dy_2 is never a prefix of a labeled path in $\mathrm{LPL}_{\mathcal{U}}(R,l)$, but xiy_1 is a prefix of some labeled paths in $\mathrm{LPL}_{\mathcal{U}}(R,l)$. Intuitively, this means, it is in fact not necessary for a uniformizer to read the idempotent factor y_2j when y_1d has already been read. Consequently, the lookahead-paths can be shortened. Secondly, we show that $value(\mathrm{LPL}_{\mathcal{U}}(R,l)) < value(\mathrm{LPL}_{\mathcal{T}}(R,l))$.

For the first part of the proof, our goal is to construct a uniformizer $\mathcal{U}$ based on $\mathcal{T}$ that realizes the following transformation. If $t \notin T_\Sigma^{xiy_1}$, then $\mathcal{U}$ produces the same output as $\mathcal{T}$, i.e., $\mathcal{T}(t) = \mathcal{U}(t)$. Otherwise, if $t \in T_\Sigma^{xiy_1}$, then $\mathcal{U}$ produces an output tree obtained as follows. Let $path(x)i = u$, $path(y_1)d = v_1$, $path(y_2)j = v_2$ and we fix t_{y_2} to be a special tree from $S_\Sigma^{y_2j\cdot\circ}$. For a tree $t \in T_\Sigma^{xiy_1}$, we let $t_x = t_{[:u]}$, $t_{y_1} = t_{[u:uv_1]}$ and $\tilde{t} = t_{[uv_1:]}$, that is, t is factorized into $t_x \cdot t_{y_1} \cdot \tilde{t}$. Furthermore, we denote by t' the tree $t_x \cdot t_{y_1} \cdot t_{y_2} \cdot \tilde{t}$ that is obtained from t by inserting t_{y_2}. The output of $\mathcal{U}$ on t is based on the computation of $\mathcal{T}$ on t'. Consider the output tree $\mathcal{T}(t') \in T_\Gamma$ which can be split into $t_{o_1} \cdot t_{o_2} \cdot t_{o_3} \cdot t_{o_4}$, where $t_{o_1} \in S_\Sigma^{o_1i\circ}$, $t_{o_2} \in S_\Sigma^{o_2d\circ}$, $t_{o_3} \in S_\Sigma^{o_3j\circ}$ and $t_{o_4} \in T_\Sigma$ such that $o_1io_2do_3$ is the labeled path over Γ with $||x|| = ||o_1||$, $||y_1|| = ||o_2||$, $||y_2|| = ||o_3||$ and $path(xiy_1dy_2) = path(o_1io_2do_3)$. That is, $o_1io_2do_3$ is the part of the output tree $\mathcal{T}(t')$ that is mapped to xiy_1dy_2 in the input tree t'. In particular, t_{o_1} is mapped to t_x, t_{o_2} is mapped to t_{y_1}, t_{o_3} is mapped to t_{y_2}, and t_{o_4} is mapped to $\tilde{t}$. Then, for $t = t_x \cdot t_{y_1} \cdot \tilde{t}$ the D↓TT $\mathcal{U}$ produces $\mathcal{U}(t) = t_{o_1} \cdot t_o \cdot t_{o_4}$, such that $t_o \in S_\Gamma^{od\circ}$ is an output tree chosen as described below. Let R be recognized by a D↓TA $\mathcal{A}$. Then, the run of $\mathcal{A}$ on $t' \otimes \mathcal{T}(t')$ along uv_1v_2 results in

$$\mathcal{A} : q_0^{\mathcal{A}} \xrightarrow{x \otimes o_1}_i q_1 \xrightarrow{y_1 \otimes o_2}_d q_2 \xrightarrow{y_2 \otimes o_3}_j q_3.$$

Note that we obtain $(q_1, q_3) \in \tau_{y_1dy_2j,o_2do_3}$, because the (partial) output tree $t_{o_3} \cdot t_{o_4}$ produced by $\mathcal{T}$ only depends on the read lookahead. Since y_2j is idempotent and $P_{y_1d} = P_{y_2j}$, we obtain $P_{y_1d} = P_{y_1dy_2j}$. Thus we can pick a labeled path $o \in \text{Path}_\Gamma$ such that $||o|| = ||y_1||$ and $y_1 \otimes o$ induces the same state transformation as $y_1dy_2 \otimes o_2do_3$ on $\mathcal{A}$ w.r.t. direction d. Let $t_o \in S_\Gamma^{odo}$ be a (partial) output tree compatible to o in the sense that for each input tree $t_{in} \in T_\Sigma^{y_1}$ there is some $t_{out} \in T_\Gamma$ such that $t_{in} \otimes (t_o \cdot t_{out}) \in T(\mathcal{A}_{q_1})$, which is guaranteed to exist because $\tau_{y_1d,o}(q_1)$ is defined. Now we show that $t \otimes \mathcal{U}(t) \in R$. The run of $\mathcal{A}$ on $t \otimes \mathcal{U}(t)$ along uv_1 results in

$$\mathcal{A} : q_0^{\mathcal{A}} \xrightarrow{x \otimes o_1}_i q_1 \xrightarrow{y_1 \otimes o}_d q_3,$$

where o_1io is the labeled path over Γ such that $\mathcal{U}(t) \in T_\Gamma^{o_1io}$. By construction of $\mathcal{U}$ we have $\mathcal{T}(t')|_{uv_1v_2} = \mathcal{U}(t)|_{uv_1} = t_{o_4}$ and also $t'|_{uv_1v_2} = t|_{uv_1} = \tilde{t}$. Since the run ρ of $\mathcal{A}$ on $t' \otimes \mathcal{T}(t')$ results in $\rho(uv_1v_2) = q_3$ and since $\tilde{t} \otimes t_{o_4} \in R_{q_3}$, we obtain $t \otimes \mathcal{U}(t) \in R$. Hence, $\mathcal{U}$ uniformizes R.

We have seen that $\mathcal{U}$ is a uniformizer for R. Now, we explain how $\mathcal{U}$ can be constructed from $\mathcal{T}$. First, we have to verify whether a labeled path that has xiy_1 as prefix is read. This can be done by copying for the first $||xiy_1||$ computation steps the behavior of $\mathcal{T}$ and additionally storing the so far read labeled path in the state space. If some $(s_1, z_1d_1) \in Q_\mathcal{T} \times \text{Path}_\Sigma\text{dir}_\Sigma$ at a σ_1-labeled node with $z_1d_1\sigma_1 \not\sqsubseteq xiy_1$ is reached, then a labeled path that does not have xiy_1 as prefix is read and $\mathcal{U}$ switches to $\mathcal{T}_{s_1}$. Otherwise, a state $(s_2, z_2d_2) \in Q_\mathcal{T} \times \text{Path}_\Sigma$ at a σ_2-labeled node with $xiy_1 = z_2d_2\sigma_2$ is reached, from then on $\mathcal{U}$ continues differently than $\mathcal{T}_{s_2}$. Recall, then t is of the form $t_x \cdot t_{y_1} \cdot \tilde{t}$. Let s denote the state that $\mathcal{T}$ reaches at the node $uv_1v_2 \in dom_{t'}$ in a computation on t' which is of the form $t_x \cdot t_{y_1} \cdot t_{y_2} \cdot \tilde{t}$. Then, $\mathcal{U}$ continues to read from the node $uv_1 \in dom_t$ and simulates the computation of $\mathcal{T}_s$ on $\tilde{t}$. In the process, $\mathcal{U}$ follows the unique path in $\tilde{t}$ that $\mathcal{T}_s$ chooses to read to produce output that is mapped to xiy_1dy_2, i.e., output that is mapped to the path uv_1v_2 in t'. Assume that uv_1v_2w is the path that is read in t' in order to produce output mapped to xiy_1dy_2 in t', i.e., $||out_\mathcal{T}(t'_{[:uv_1v_2w]}, uv_1v_2)|| = ||xiy_1dy_2||$. When $\mathcal{U}$ reaches uv_1w in t, then $\mathcal{U}$ produces an output tree t_o compatible to a labeled path o, where o is chosen as described above. Note, then $||out_\mathcal{U}(t_{[:uv_1w]}, uv_1)|| = ||xiy_1||$. After producing t_o, the transducer $\mathcal{U}$ switches to $\mathcal{T}_{s_k}$ at the kth child of uv_1w in t, if s_k is the state that $\mathcal{T}$ would be in when $\mathcal{T}$ reads the node uv_1v_2wk in t'. Such a transducer $\mathcal{U}$ realizes the above described uniformization. Note that $\gamma \in \text{LPL}_\mathcal{U}(R, l)$, because the behavior of $\mathcal{U}$ only differs from $\mathcal{T}$ after reading xiy_1, and since $\gamma \leq l$, we have $||\gamma|| < ||xiy_1||$.

For the second part, we now show $value(\text{LPL}_\mathcal{U}(R, l)) < value(\text{LPL}_\mathcal{T}(R, l))$. We first remark, that if $\pi \in \text{LPL}_\mathcal{U}(R, l)$ with $xiy_1 \not\sqsubseteq \pi$, then also $\pi \in \text{LPL}_\mathcal{T}(R, l)$, because on paths that do not have xiy_1 as prefix, $\mathcal{U}$ works like $\mathcal{T}$. Also, if

$\pi \in \text{LPL}_{\mathcal{U}}(R,l)$ with $\pi \sqsubseteq xiy_1$, then $\pi \in \text{LPL}_{\mathcal{T}}(R,l)$, because for the first $||xiy_1||$ computation steps, $\mathcal{U}$ works like $\mathcal{T}$. Otherwise, if $\pi \in \text{LPL}_{\mathcal{U}}(R,l)$ with $xiy_1d \sqsubseteq z$, let π be of the form xiy_1dz for some $z \in \text{Path}_\Sigma$ and let $path(z) = w$. We show $xiy_1dy_2jz \in \text{LPL}_{\mathcal{T}}(R,l)$. Consider an input tree $t' \in T_\Sigma^{xiy_1dy_2jz}$ that is obtained from a tree $t \in T_\Sigma^{xiy_1dz}$ by inserting t_{y_2}. Then, from the construction of $\mathcal{U}$ follows that $\mathcal{T}$ reads xiy_1dy_2jz in t'. Since $xiy_1dz \in \text{LPL}_{\mathcal{U}}(R,l)$, we know less then l output symbols have been produced while reading the path uv_1w in t, i.e., $||out_{\mathcal{U}}(t_{[:uv_1w]},u)|| < l$. Moreover, it follows from the construction of $\mathcal{U}$ that if $||out_{\mathcal{U}}(t_{[:uv_1w]},u)|| < l$, then also $||out_{\mathcal{T}}(t'_{[:uv_1v_2w]},u)|| < l$, because by assumption $l \leq ||x||$ and by construction $out_{\mathcal{T}}(t'_{[:uv_1v_2w]},u) = out_{\mathcal{U}}(t_{[:uv_1w]},u)$ as long as $||out_{\mathcal{T}}(t'_{[:uv_1v_2w]},u)|| \leq ||x||$. We can conclude that $xiy_1dy_2jz \in \text{LPL}_{\mathcal{T}}(R,l)$ if $R^{xiy_1dy_2jz}$ is not uniformizable by a path-recognizable function. Since $xiy_1dz \in \text{LPL}_{\mathcal{U}}(R,l)$, we know R^{xiy_1dz} is not uniformizable by a path-recognizable function. Towards a contradiction, assume $R^{xiy_1dy_2jz}$ is uniformizable by a path-recognizable function. Then, from a uniformizer for $R^{xiy_1dy_2jz}$ that realizes a path-recognizable function, we can easily obtain a uniformizer for R^{xiy_1dz} that realizes a path-recognizable function, because $Py_1dy_2j = P_{y_1d}$. This is a contradiction to $xiy_1dz \in \text{LPL}_{\mathcal{U}}(R,l)$.

Altogether, for each labeled path $\pi \in \text{LPL}_{\mathcal{U}}(R,l)$, there is either $\pi \in \text{LPL}_{\mathcal{T}}(R,l)$ if $\pi \not\sqsubseteq xiy_1$ or $\pi \sqsubseteq xiy_1$, or $xiy_1dy_2jz \in \text{LPL}_{\mathcal{T}}(R,l)$ if π is of the form xiy_1dz for some $z \in \text{Path}_\Sigma$. That means, $value(\text{LPL}_{\mathcal{U}}(R,l)) \leq value(\text{LPL}_{\mathcal{T}}(R,l))$. However, by assumption there is some $z \in \text{Path}_\Sigma$ such that $xiy_1dy_2jz \in \text{LPL}_{\mathcal{T}}(R,l)$ and by construction we obtain $xiy_1dz \in \text{LPL}_{\mathcal{U}}(R,l)$. Thus, $value(\text{LPL}_{\mathcal{U}}(R,l)) < value(\text{LPL}_{\mathcal{T}}(R,l))$. This is a contradiction to $\mathcal{T}$ being a uniformizer for R with $\gamma \in \text{LPL}_{\mathcal{T}}(R,l)$ that has minimal complexity w.r.t. l compared to every other uniformizer $\mathcal{U}$ for R with $\gamma \in \text{LPL}_{\mathcal{U}}(R,l)$. □

Now that we have completed all preparations, we present a decision procedure for the case of arbitrary delay. Therefore, we consider a similar safety game as in the previous section on uniformization with bounded output delay. We only have to make one adaption to the game graph. Let K be as in Remark 10, and let $\mathcal{G}_\mathcal{A}^{2K}$ be defined as $G_\mathcal{A}^{2K}$ from Section 3.1 with the following modification. From each vertex $(q,\pi) \in V_{\mathsf{Out}}$ we add a move that allows Out to stay in this vertex if R_q^π can be uniformized by a path-recognizable function. These changes to the game graph can be made effectively, because Theorem 9 implies that it is decidable whether there exists a corresponding uniformization by a path-recognizable function. Using Lemma 13, we see (in the proof of the lemma below) that it suffices to explicitly model situations in which the output delay is at most $2K$.

Lemma 14 *R has a uniformization by a D↓TT if, and only if, Out has a winning strategy in the safety game $\mathcal{G}_\mathcal{A}^{2K}$.*

Proof. Assume that Out has a winning strategy in $\mathcal{G}_\mathcal{A}^{2K}$, then there also exists a positional winning strategy for Out. To construct a D↓TT $\mathcal{T}$ that uniformizes R, we proceed as presented in the proof of Lemma 6 with one addition. We construct for each $(q,\pi) \in V_{\mathsf{Out}}$ such that R_q^π can be uniformized by a path-recognizable function a D↓TT $\mathcal{T}_q^\pi$ that uniformizes R_q^π. In $\mathcal{T}$ we switch to $\mathcal{T}_q^\pi$ at the respective states.

For the other direction, assume that R is uniformizable by a D↓TT. Again, the proof is similar to the proof of Lemma 7. We show by induction on the number of moves played by Out that the strategy in $\mathcal{G}_{\mathcal{A}}^{2K}$ can be chosen such that in every play according to the strategy the following induction hypothesis is satisfied. Let (q,π) denote a vertex of Out that is reached after a sequence of moves in a play. W.l.o.g., we make the assumption that in this sequence until now no vertex of Out was reached that has a self-loop. If a vertex with self-loop was reached, then Out can stay in this vertex and wins. We claim that there is some $l \leq K$ such that π can be split into $xiyjf$ for some $x, y \in \mathrm{Path}_\Sigma$, $i,j \in \mathrm{dir}_\Sigma$ and $f \in \Sigma$ with $||x|| = l$ and there exists a D↓TT $\mathcal{T}$ that uniformizes R_q^x such that the following holds: $xiy \in \mathrm{LPL}_{\mathcal{T}}(R_q^x, l)$ if $||xiy|| \geq 1$, and for every D↓TT $\mathcal{U}$ that uniformizes R_q^x with $x \in \mathrm{LPL}_{\mathcal{U}}(R_q^x, l)$ holds $value(\mathrm{LPL}_{\mathcal{T}}(R_q^x, l)) \leq value(\mathrm{LPL}_{\mathcal{U}}(R_q^x, l))$.

In words, if in a play $(q,\pi) \in V_{\mathsf{Out}}$ is reached, then there exists a factorization of π into $xiyjf$ such that there is a uniformizer $\mathcal{T}$ of R_q^x with xiy as lookahead-path and $\mathcal{T}$ has minimal complexity w.r.t. $||x||$ compared to every uniformizer of R_q^x that has x as lookahead-path.

Note, we allow to choose x or y resp. x and y as ε, then we identify $xiyjf$ with yjf or xif resp. f. First, we show that the induction hypothesis is true at the first reached vertex of Out in a play. Such a vertex is of the form $(q_0, f) \in V_{\mathsf{Out}}$ for some $f \in \Sigma$. For $l = 0$, i.e., $x = \varepsilon$ and $y = \varepsilon$, the induction hypothesis can be satisfied. Since R is uniformizable, there exists a D↓TT $\mathcal{T}$ with minimal complexity w.r.t. limit 0 that uniformizes $R_{q_0}^\varepsilon = R$.

Now we define the strategy. Assume the play is in a vertex $(q,\pi) \in V_{\mathsf{Out}}$ and the induction hypothesis is true for some $l \leq K$ and π can be split accordingly into $xiyjf$ with $||x|| = l$ and let $\mathcal{T}$ by a uniformizer that satisfies the claim. To define the next move of Out, we consider the computation of $\mathcal{T}$ on some $t \in T_\Sigma^{xiyjf}$ and check whether the output produced while reading $xiyjf$ exceeds the limit l. Note, the induction hypothesis states that $xiy \in \mathrm{LPL}_{\mathcal{T}}(R_q^x, l)$, which means the output produced while reading xiy does not exceed the limit l.

If $||\mathcal{T}(t[\circ/path(xiy)j] \cdot f)||^{path(x)} \geq l$, that is, $\mathcal{T}$ produces at least l output symbols while reading $xiyjf$, then the strategy defines output moves. We pick an arbitrary $z \in out_{\mathcal{T}}(t[\circ/path(xiy)j] \cdot f, path(x))$, then Out makes l output moves according to the prefix of length l of z. Let o denote this prefix. Note that o is the greatest common prefix of every labeled path from $out_{\mathcal{T}}(t[\circ/path(xiy)j] \cdot f, path(x))$. This leads to a vertex $(q', yjf) \in V_{\mathsf{Out}}$ with $\mathcal{A} : q \xrightarrow{x \otimes o}_i q'$. Since $\mathcal{T}$ is a uniformizer for R_q^x that reads $xiyjf$ and $\tau_{xi,o}(q) = q'$, there exists also a uniformizer for $R_{q'}^{yjf}$. Moreover, from the induction hypothesis follows $||out_{\mathcal{T}}(t[\circ/path(xiy)j], path(x))|| < l$. This implies that there exists a uniformizer for $R_{q'}^{yjf}$ such that yif is a lookahead-path of this uniformizer w.r.t. limit $||yif||$. Let $\mathcal{T}'$ denote such a uniformizer that has minimal complexity w.r.t. $||yjf||$ compared to every uniformizer of $R_{q'}^{yjf}$ that also has yif as lookahead-path. For $(q', yjf) \in V_{\mathsf{Out}}$, the induction hypothesis is then satisfied by choosing a new limit $l = ||yjf||$ and $\mathcal{T}'$ as described.

Otherwise, if $||\mathcal{T}(t[\circ/path(xiy)j] \cdot f)||^{path(x)} < l$, that is, $\mathcal{T}$ produces at less than l output symbols while reading $xiyjf$, we distinguish two cases. For the first case assume $||\pi|| \leq 2K$, then Out delays and picks direction d chosen as follows. Since $xiy \in \mathrm{LPL}_{\mathcal{T}}(R_q^x, l)$ and $||out_{\mathcal{T}}(t[\circ/path(xiy)j] \cdot f, path(x))|| < l$, it follows that for every $s \in T_\Sigma^{xiyjf}$ there is a configuration $c = (s, s', \varphi)$ of $\mathcal{T}$ reachable such that there is $u \in D_{s'}$ with $\varphi(u) = path(xiyjf)$ and $u \sqsubseteq \varphi(u)$. Then, there exists a configuration $c' = (s, s'', \varphi')$ with $c \rightarrow_{\mathcal{T}} c'$ such that there is $u' \in D_{s''}$ with $u \sqsubseteq u' \sqsubseteq path(xiyjf)$ and $\varphi'(u') = path(xiyif)d$ for some direction d. Out moves to $(q, xiyjfd)$. Then, the next reached vertex of Out is of the form $(q, xiyjfdg)$ for some $g \in \Sigma$. The induction hypothesis is satisfied for the same limit $||x||$ and $\mathcal{T}$ as before: Clearly, we obtain $xiyjf \in \mathrm{LPL}_{\mathcal{T}}(R_q^x, l)$, because $||out_{\mathcal{T}}(t[\circ/path(xiy)j] \cdot f, path(x))|| < l$. Also, as before, $\mathcal{T}$ has minimal complexity w.r.t. $||x||$ compared to every uniformizer that also has x as lookahead-path.

For the second case assume $||\pi|| = 2K + 1$. Since l is at most K, $||y|| \geq K$. By induction hypothesis, we are guaranteed that $\mathcal{T}$ is a uniformizer for R_q^x with minimal complexity w.r.t. l compared to every uniformizer that also has x as lookahead-path. Together with Lemma 13, this implies that $xiyjf \notin \mathrm{LPL}_{\mathcal{T}}(R_q^x, l)$, because the length of a path from $\mathrm{LPL}_{\mathcal{T}}(R_q^x, l)$ is bounded by $K + l < 2K + 1$. However, since $\mathcal{T}$ uniformizes R_q^x and reads $xiyjf$ this means that R_q^{xiyjf} is uniformizable by a path-recognizable function. Consequently, the vertex $(q, \pi) \in V_{\mathsf{Out}}$ has a self-loop and Out stays in this vertex from then on.

The strategy is winning because it ensures that Out can always make a move. □

As a consequence of Lemma 14 and the fact that a winning strategy for Out in $\mathcal{G}_{\mathcal{A}}^{2K}$ can effectively be computed we immediately obtain our main result.

Theorem 15 *It is decidable whether a D↓TA-recognizable relation with total domain has a uniformization by a deterministic top-down tree transducer.*

As mentioned in the beginning, the presented results are also valid for D↓TA-recognizable relations with D↓TA-recognizable domain in the sense that a ↓TT that realizes a uniformization of a relation may behave arbitrarily on trees that are not part of the domain. The presented constructions have to be adapted such that In, given a D↓TA for the domain, also keeps track of the state in the input tree in order to play only correct input symbols.

3.3 Input validation

In the former section, we assumed that a top-down tree transducer that implements a uniformization of a relation is only given valid input trees. In this section we consider the case that a top-down transducer also has to validate the correctness of a given input tree.

We will see that in this case it can be necessary that a transducer takes divergent paths for input and output. The following example shows that there exists a D↓TA-recognizable relation with D↓TA-recognizable domain that can be uniformized by a D↓TT, but every such D↓TT has a reachable configuration (t, t', φ) such that $\varphi(u) \not\sqsubseteq u$ and $u \not\sqsubseteq \varphi(u)$ for some node u.

Example 16 Let Σ be given by $\Sigma_2 = \{f\}$ and $\Sigma_0 = \{a, b\}$. We consider the relation $R_1 \subseteq T_\Sigma \times T_\Sigma$ defined by $\{\big(f(b,t), f(t',b)\big) \mid \neg\exists u \in \mathit{dom}_t : \mathit{val}_t(u) = b\}$. Clearly, both R_1 and $\mathit{dom}(R_1)$ are D↓TA-recognizable. One way to uniformize R_1 is by swapping the left and right subtree of the root and verify the required properties, as done by the following D↓TT $\mathcal{T} = (\{q_0, q_1, q_2\}, \Sigma, \Sigma, q_0, \Delta)$ with $\Delta =$

$$\begin{aligned} \{ \quad & q_0(f(x_1,x_2)) \to f(q_1(x_2), q_2(x_1)), \quad q_1(a) \to b, \\ & q_1(f(x_1,x_2)) \to f(q_1(x_1), q_1(x_2)), \quad q_2(b) \to b \quad \}. \end{aligned}$$

However, there exists no D↓TT $\mathcal{T}'$ that uniformizes R_1 such that the read input sequence and the produced output are on the same path. Intuitively, a D↓TT $\mathcal{T}$ that uniformizes R_1 must read the whole right subtree of an input tree $f(b,t)$ in order to verify that there is no occurrence of b. If an f in t is read and no output is produced, a D↓TT can either continue to read left or right, but cannot verify both subtrees. Therefore, in order to verify t, a D↓TT has to produce an output symbol at each read inner node which results in an output tree of the same size. Assume such a D↓TT $\mathcal{T}'$ exists, then for an initial state q_0 there is a transition of the form $q_0(f(x_1,x_2)) \to f(q_1(x_1), q_2(x_2))$. It follows that $\mathcal{T}'_{q_2}$ must induce the relation $\{(t,b) \mid t \in T_\Sigma \wedge \neg\exists u \in \mathit{dom}_t : \mathit{val}_t(u) = b\}$. The only output that $\mathcal{T}'_{q_2}$ can produce is exactly one b. Thus, there is a transition with left-hand side $q_2(f(x_1,x_2))$ that has one of the following right-hand sides: b, $q_3(x_1)$, or $q_3(x_2)$. No matter which right-hand side is chosen, $\mathit{dom}(R(\mathcal{T}'_{q_2}))$ must also contain trees with occurrences of b. ◁

It follows directly from the above example that the presented decision procedure is invalid if the domain of a considered relation is not total. However, if we restrict ourselves to uniformizations such that a D↓TT only contains rules of the form $q(f(x_1,\dots,x_i)) \to g(q_1(x_{j_1}),\dots,q_n(x_{j_n}))$, where $g \in \Gamma_n$ is a single symbol i.e., read input symbol and correspondingly produced output are always on the same tree level, it is possible to adapt the presented decision procedure from Section 3.1. Recall the definition of bounded delay, this kind of D↓TTs are referred to as D↓TTs without delay.

Theorem 17 *It is decidable whether a D↓TA-recognizable relation with D↓TA-recognizable domain has a uniformization by a deterministic top-down tree transducer without delay.*

For this purpose, we can change the game graph in the following way. Let $\mathcal{A}$ be a D↓TA for a relation and $\mathcal{B}$ be a D↓TA for its domain. The main difference to the previous section is that we have to model that input and output can be on divergent paths.

Regarding the moves of In, the vertices track which $\mathcal{B}$-state is reached on the played input sequence. Regarding the moves of Out, we distinguish two cases. If the played input sequence and the correspondingly produced output sequence are on the same path, then – as before – the vertices track which $\mathcal{A}$-state is reached on the combination of the played input and produced output sequence. Otherwise, if the played input sequence and the correspondingly produced output sequence are on divergent paths, then the vertices track which $\mathcal{A}$-states can be reached on the combination of each possible input sequence (that shares the same path as the produced output sequence) together with the produced output sequence.

The move constraints for In are chosen such that it is guaranteed that the played input sequence is valid. The move constraints for Out are chosen such that it is guaranteed that the combination of the played input sequence together with her produced output sequence resp. the combination of each possible input sequence together with her produced output sequence is valid. Details for this construction can be found in [18].

4 Conclusion

In this paper, we considered the synthesis of deterministic top-down tree transducers from tree automatic specifications. We have shown that it is decidable whether a deterministic top-down specification can be realized by a top-down tree transducer under the restriction that the transducer is not required to validate the input, meaning that a transducer implementing a uniformization can behave arbitrarily on invalid inputs. If uniformization is possible, our decision procedure yields a top-down tree transducer that realizes the specification.

We have seen that the presented decision procedure concerning uniformization without input validation cannot be transferred directly to decide the problem corresponding to the classical uniformization question (with input validation). The reason for this is that in the employed transducer model it is not possible to verify the input without producing output.

In future work, we would like to develop methods for solving the general case of nondeterministic specifications. Furthermore, we would also like to consider nondeterministic top-down tree transducers as specification formalism. For this class of specifications, the synthesis problem in its full generality is undecidable, because it already is for the restriction to words (synthesis of sequential transducers from rational relations [6]). We would like to identify interesting decidable restrictions of this problem.

References

[1] A. Church, Logic, arithmetic and automata, in: Proceedings of the International Congress of Mathematicians, 1962, pp. 23–35.

[2] A. Church, Applications of recursive arithmetic to the problem of circuit synthesis, Summaries of the Summer Institute of Symbolic Logic 1 (1957) 3–50.

[3] J. R. Büchi, L. H. Landweber, Solving sequential conditions by finite-state strategies, Transactions of the American Mathematical Society 138 (1969) 295–311. doi:10.1090/S0002-9947-1969-0280205-0.

[4] F. A. Hosch, L. H. Landweber, Finite delay solutions for sequential conditions, in: ICALP, 1972, pp. 45–60.

[5] M. Holtmann, Ł. Kaiser, W. Thomas, Degrees of lookahead in regular infinite games, in: Foundations of Software Science and Computational Structures, Vol. 6014 of Lecture Notes in Computer Science, Springer, 2010, pp. 252–266. doi:/10.1007/978-3-642-12032-9_18.

[6] A. Carayol, C. Löding, Uniformization in Automata Theory, in: Proceedings of the 14th Congress of Logic, Methodology and Philosophy of Science Nancy, July 19-26, 2011, London: College Publications, 2014, pp. 153–178.

[7] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, M. Tommasi, Tree automata techniques and applications, release October, 12th 2007 (2007).
URL http://www.grappa.univ-lille3.fr/tata

[8] T. Milo, D. Suciu, V. Vianu, Typechecking for xml transformers, J. Comput. Syst. Sci. 66 (1) (2003) 66–97. doi:10.1016/S0022-0000(02)00030-2.

[9] J. Engelfriet, On tree transducers for partial functions, Inf. Process. Lett. 7 (4) (1978) 170–172. doi:10.1016/0020-0190(78)90060-1.

[10] C. Löding, S. Winter, Synthesis of deterministic top-down tree transducers from automatic tree relations, in: Proceedings Fifth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2014, Verona, Italy, September 10-12, 2014., Vol. 161 of EPTCS, 2014, pp. 88–101. doi:10.4204/EPTCS.161.

[11] F. Gécseg, M. Steinby, Tree Automata, Akadémiai Kiadó, Budapest, 1984.

[12] E. Grädel, W. Thomas, T. Wilke (Eds.), Automata, Logics, and Infinite Games, Vol. 2500 of Lecture Notes in Computer Science, Springer, 2002.

[13] E. Grädel, Finite model theory and descriptive complexity, in: Finite Model Theory and Its Applications, Springer, 2007, pp. 125–230. doi:10.1007/3-540-68804-8_3.

[14] W. Thomas, Automata on infinite objects, in: Handbook of Theoretical Computer Science, Vol. B: Formal Models and Semantics, Elsevier Science Publishers, Amsterdam, 1990, pp. 133–192.

[15] M. O. Rabin, Automata on Infinite Objects and Church's Problem, American Mathematical Society, Boston, MA, USA, 1972.

[16] F. Ramsey, On a problem of formal logic, Proceedings of the London Mathematical Society 2 (1) (1930) 264. doi:10.1007/978-0-8176-4842-8_1.

[17] R. Diestel, Graph Theory, 2nd Edition, Vol. 173 of Graduate Texts in Mathematics, Springer, 2000.

[18] S. Winter, Uniformization of automaton definable tree relations, Masterthesis, RWTH Aachen, Germany (2013).