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

Christof Löding Sarah Winter
Lehrstuhl für Informatik 7, RWTH Aachen, Germany
{loeding,winter}@automata.rwth-aachen.de

We consider the synthesis of deterministic tree transducers from automaton definable specifications, given as binary relations, over finite trees. We consider the case of specifications that are deterministic top-down tree automatic, meaning the specification is recognizable by a deterministic 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 bounded delay or arbitrary delay. Delay is caused whenever the transducer reads a symbol from the input tree but does not produce output. 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. Similar to the case of relations over words, we use two-player games 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 [3] which was already posed in 1957. 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 related notion is the one 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.

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 [1], 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 [1] can be translated into finite automata that read the two input words synchronously. Such relations 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. For a bounded delay decidability was shown in [10], and for an unbounded delay in [9]. In the case of finite words, it was shown in [2]

Adriano Peron and Carla Piazza (Eds.):
Proceedings of the Fifth International Symposium on
Games, Automata, Logics and Formal Verification (GandALF 2014)
EPTCS 161, 2014, pp. 88–101, doi:10.4204/EPTCS.161.10

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 tool for analyzing and manipulating rewrite systems or XML Schema languages (see [4]). Tree transformations that are realized by finite tree transducers thus become interesting in the setting of translations from one document scheme into another [12]. There are already some uniformization results for tree relations. For example, in [5] 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. However, these results focus on the existence of a uniformization for each relation that can be specified by the considered model. In contrast to that, we are interested in the corresponding decision problem. More precisely, for a class $\mathscr{C}$ of tree relations and a class $\mathscr{F}$ of functions over trees, we are interested in a procedure that decides whether a given relation from $\mathscr{C}$ has a uniformization in $\mathscr{F}$.

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 by deterministic top-down tree transducers. We distinguish between two variants of uniformization. In the first setting, we do not require that a transducer validates whether an input tree is part of the domain of the given specification. We allow a transducer to behave arbitrarily on invalid input trees. In the second setting, the desired transducer has to reject invalid input trees. We speak of uniformization without resp. with input validation. For uniformization without input validation, we consider the case that the transducer defining a uniformization has no restrictions. In particular the transducer is allowed to skip an unbounded number of output symbols thereby introducing delay. 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 uniformization with input validation, we see that this variant is more complex than uniformization without input validation. We show decidability in case that a transducer realizing a uniformization, synchronously produces one output symbol per read input symbol.

The paper is structured as follows. First, we fix some basic definitions and terminology. Then, in Section 3 and Section 4, we consider uniformization by top-down tree transducers without input validation that have bounded delay and unbounded delay, respectively. In Section 5, we briefly consider the case of uniformization with input validation.

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 $w = uv$ 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 arities $rk(f) \subseteq \mathbb{N}$. The set of letters of arity 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 corresponds to a rank of $val_t(u)$. 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 $\Sigma \cup \{\circ\}$ 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 $t \in T_\Sigma$ or $t \in S_\Sigma$ and $s \in S_\Sigma$ let the *concatenation* $t \cdot s$ be the tree that is obtained from t by replacing $\circ$ with s.

Let X_n be a set of n variables $\{x_1, \ldots, x_n\}$ 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. Let $X = \bigcup_{n>0} X_n$. For $t \in T_\Sigma(X_n)$ let $t[x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n]$ 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 $x_1, \ldots, x_n$ when reading the leaf nodes from left to right, is called *n-context* over Σ. A special tree can be seen as an 1-context. If C is an n-context and $t_1, \ldots, t_n \in T_\Sigma(X)$ we write $C[t_1, \ldots, t_n]$ instead of $C[x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n]$.

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. [6] or [4].

Let $\Sigma = \bigcup_{i=1}^m \Sigma_i$ be a ranked alphabet. A *non-deterministic top-down tree automaton* (an N$\downarrow$TA) over Σ is of the form $\mathscr{A} = (Q, \Sigma, Q_0, \Delta)$ 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 $\mathscr{A}$ be an N$\downarrow$TA, a *run* of $\mathscr{A}$ on t is a mapping $\rho : dom_t \to Q$ compatible with Δ, i.e., $\rho(\varepsilon) \in Q_0$ and for each node $u \in dom_t$, if $val_t(u) \in \Sigma_i$ with $i \geq 0$, then $(\rho(u), val_t(u), \rho(u1), \ldots, \rho(ui)) \in \Delta$. A tree $t \in T_\Sigma$ is *accepted* if, and only if, there is a run of $\mathscr{A}$ on t. The tree language *recognized* by $\mathscr{A}$ is $T(\mathscr{A}) = \{t \in T_\Sigma \mid \mathscr{A} \text{ accepts } t\}$.

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

A top-down tree automaton $\mathscr{A} = (Q, \Sigma, Q_0, \Delta)$ is *deterministic* (a D$\downarrow$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 $(q, f, q_1, \ldots, q_i) \in \Delta$. 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 $\Sigma_\perp = \Sigma \cup \{\perp\}$, $\Gamma_\perp = \Gamma \cup \{\perp\}$. 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 $val_t(u) = (val^\perp_{t_1}(u), val^\perp_{t_2}(u))$ with $rk(val_t(u)) = max\left\{rk(val^\perp_{t_1}(u)), rk(val^\perp_{t_2}(u))\right\}$ 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 $\mathscr{A}$ recognizes R if it recognizes the convolution T_R and denote by $R(\mathscr{A})$ 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 for a given tree-automatic relation, whether 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 leafs, 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 [4].

A *top-down tree transducer* (a TDT) is of the form $\mathscr{T} = (Q, \Sigma, \Gamma, q_0, \Delta)$ 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

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

where $f \in \Sigma_i$, w is an n-context over Γ, $q, q_1, \ldots, q_n \in Q$ and $j_1, \ldots, j_n \in \{1, \ldots, i\}$, or

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

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

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 $c_1 = (t, t_1, \varphi_1), c_2 = (t, t_2, \varphi_2)$ be configurations of a top-down tree transducer. We define a successor relation $\to_{\mathscr{T}}$ on configurations as usual by applying one rule, which looks formally (for the application of a non-ε-rule) as follows:

$$c_1 \to_{\mathscr{T}} c_2 :\Leftrightarrow \begin{cases} \exists u \in \mathrm{dom}_{t_1}, q \in Q, v \in \mathrm{dom}_t \text{ with } \mathrm{val}_{t_1}(u) = q \text{ and } \varphi_1(u) = v \\ \exists q(\mathrm{val}_t(v)(x_1, \ldots, x_i)) \to w[q_1(x_{j_1}), \ldots, q_n(x_{j_n})] \in \Delta \\ t_2 = s \cdot w[q_1, \ldots, q_n] \text{ with } s = t_1[\circ/u] \\ \varphi_2 \text{ with } D_{t_2} = D_{t_1} \setminus \{u\} \cup \{u_i \mid u \sqsubseteq u_i, \mathrm{val}_{t_2}(u_i) = q_i, 1 \le i \le n\} \\ \forall u' \in D_{t_1} \setminus \{u\} : \varphi_2(u') = \varphi_1(u') \\ \forall u_i, u \sqsubseteq u_i, \mathrm{val}_{t_2}(u_i) = q_i : \varphi_2(u_i) = v.j_i \end{cases}$$

Furthermore, let $\to^*_{\mathscr{T}}$ be the reflexive and transitive closure of $\to_{\mathscr{T}}$ and $\to^n_{\mathscr{T}}$ the reachability relation for $\to_{\mathscr{T}}$ in n steps. The relation $R(\mathscr{T}) \subseteq T_\Sigma \times T_\Gamma$ induced by a top-down tree transducer $\mathscr{T}$ is

$$R(\mathscr{T}) = \{(t, t') \mid (t, q_0, \varphi_0) \to^*_{\mathscr{T}} (t, t', \varphi') \text{ with } \varphi_0(\varepsilon) = \varepsilon, \varphi' \text{ is empty and } t' \in T_\Gamma\}.$$

For a tree $t \in T_\Sigma$ let $\mathscr{T}(t)$ be the final transformed output of $\mathscr{T}$ for t. The class of relations definable by TDTs 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 TDT $\mathscr{T}$ given by $(\{q\}, \Sigma, \Sigma, \{q\}, \Delta)$ with $\Delta = \{\ q(a) \to a,\ q(g(x_1)) \to q(x_1),\ q(h(x_1)) \to h(q(x_1)),\ q(f(x_1, x_2)) \to f(q(x_1), q(x_2))\ \}$. 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 $\mathscr{T}$ on t is $c_0 \to^5_{\mathscr{T}} c_5$ 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 $\varphi_4(11) = 111$, and $c_5 := (t, f(h(a), a), \varphi_5)$. A visualization of c_2 is shown in Figure 1. ◁

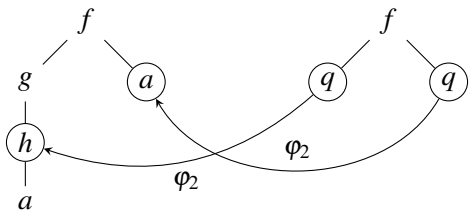

Figure 1: The configuration $c_2 = (t, f(q,q), \varphi_2)$ of $\mathscr{T}$ on t from Example 1.

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

- $V = V_0 \dot\cup V_1$ 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.

A play is a maximal 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 vertex without outgoing edges. 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 $\sigma_i : V^* V_i \to V$ 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. [7], 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 is a mapping $\sigma_i : V_i \to V$ such that $(v, \sigma_i(v)) \in E$ for all $v \in V_i$.

3 Bounded Delay

In this section we investigate uniformization of tree-automatic relations in the class of top-down tree transformations. We restrict ourselves in the scope of this section to D↓TA-recognizable relations with D↓TA-recognizable domain. 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 issues, we will speak of uniformization with input validation and uniformization without input validation. For ease of presentation, in this and the following section we will only consider relations with total domain. Later on, we will briefly describe how we can deal with relations whose domain is not total but deterministic top-down tree automatic.

For the remainder of this paper, let $R \subseteq T_\Sigma \times T_\Gamma$ be a deterministic top-down tree automaton-definable relation with total domain and let $\mathscr{A} = (Q_\mathscr{A}, \Sigma_\perp \times \Gamma_\perp, q_0^\mathscr{A}, \Delta_\mathscr{A})$ be a D↓TA that recognizes R. For $q \in Q_\mathscr{A}$, let $\mathscr{A}_q$ be the automaton that results from $\mathscr{A}$ by using q as single initial state.

To begin with, we investigate the connection between input and output. A TDT $\mathscr{T}$ possibly reaches a point such that the position v of the input symbol under consideration and the position u of the correspondingly produced output are different (formally, the mapping φ from the configuration maps u to

a node $v \neq u$). As a consequence, if u and v lie on divergent paths, then $\mathscr{T}$ selects the output at u (and below u) independent of the subtree at u of the input tree. That means, in order to satisfy a given specification, if $\mathscr{T}$ selects the output independent of the input from some point on, then the produced output tree must match all possible input trees from then on. Such cases are considered in the Lemma below.

Lemma 2 *Let $q \in Q_{\mathscr{A}}$. It is decidable whether the following holds:*

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

Now, we formally define what is meant by output delay. Let $\mathscr{T}$ be a TDT and let $c = (t, t', \varphi)$ be a configuration of $\mathscr{T}$. Consider a node $u \in D_{t'}$. If $|\varphi(u)| \geq |u|$, we say the transducer has an *output delay* of $|\varphi(u)| - |u|$. If there is a $k \in \mathbb{N}$ such that for all reachable configurations $c = (t, t', \varphi)$ of $\mathscr{T}$ holds that for all $u \in D_{t'}$ the output delay $|\varphi(u)| - |u|$ is at most k, then the output delay is bounded to k.

We will solve the following problem.

Theorem 3 *Given $k > 0$, 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 to 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* and the word *lbls* induced by π inductively by:

- if $\pi = \varepsilon$ or $\pi = f$, then $path(\varepsilon) = path(f) = \varepsilon$, $lbls(\varepsilon) = \varepsilon$ and $lbls(f) = f$,
- if $\pi = x \cdot jf$ with $j \in \mathbb{N}$, $f \in \Sigma$, then $path(\pi) = path(x) \cdot j$, $lbls(\pi) = lbls(x) \cdot f$.

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

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

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

be the set of trees t over Σ such that π is a prefix of a labeled path through t. For $\Pi \subseteq \mathrm{Path}_\Sigma$ let

$$T_\Pi := \{t \in T_\Sigma \mid \exists \pi \in \Pi \text{ and } t \in T_\Sigma^\pi\}$$

be the set of trees such that each tree contains a labeled path starting with π for some $\pi \in \Pi$.

For $t \in T_\Sigma$ and $u \in \mathrm{dir}_\Sigma^*$, let $||t||^u := max\{|v| \mid v \in dom_t \text{ and } (u \sqsubseteq v \text{ or } v \sqsubseteq u)\}$ be the length of a maximal path through t along u.

Now, in order to solve the above decision problem, we consider a safety game between two players. The procedure is similar to a decision procedure presented in [2], 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. 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 and react with a direction in which In should continue with his input sequence.

The vertices in the game graph keep track of the state of $\mathscr{A}$ on the input combined with the output on the same path and additionally of the input that is ahead, which is bounded to 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 $G^k_{\mathscr{A}}$ is constructed as follows.

- $V_{\mathsf{In}} = \{(q,\pi j) \in Q_{\mathscr{A}} \times \mathrm{Path}_\Sigma \cdot \mathrm{dir}_\Sigma \mid \|\pi\| \leq k, \pi \in \mathrm{Path}_\Sigma, j \in \mathrm{dir}_\Sigma\} \cup 2^{Q_{\mathscr{A}}}$ is the set of vertices of player In.
- $V_{\mathsf{Out}} = \{(q,\pi) \in Q_{\mathscr{A}} \times \mathrm{Path}_\Sigma \mid \|\pi\| \leq k\}$ is the set of vertices of player Out.
- From a vertex of In the following moves are possible:
  - i) $(q,\pi j) \to (q,\pi j f)$ for each $f \in \Sigma$ if $\|\pi\| < k$ (delay; In chooses the next input)
  - ii) $\{q_1,\ldots,q_n\} \to (q_i,f)$ for each $i \in \{1,\ldots,n\}$ (no delay; In chooses the next direction and input)
- From a vertex of Out the following moves are possible:
  - iii) $(q,f) \xrightarrow{r} \{q_1,\ldots,q_i\}$ if there is $r = (q,(f,g),q_1,\ldots,q_n) \in \Delta_{\mathscr{A}}$, $f \in \Sigma$ is i-ary, $g \in \Sigma_\perp$ is j-ary, and if $j > i$, there exist trees $t_{i+1},\ldots,t_j \in T_\Gamma$ such that $\perp \otimes t_l \in T(\mathscr{A}_{q_l})$ for all $i < l \leq j$. (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)
  - iv) $(q,\pi j' f') \xrightarrow{r} (q',\pi' j' f')$ for each $g \in \Gamma_\perp$ such that $\pi = f j \pi'$, there is $r = (q,(f,g),q_1,\ldots,q_n) \in \Delta_{\mathscr{A}}$ with $q' = q_j$, and for each $l \neq j$ with $l \in \{1,\ldots,n\}$ holds
    - if $l \leq rk(f), rk(g)$, then $\exists t' \in T_\Gamma \forall t \in T_\Sigma : t \otimes t' \in T(\mathscr{A}_{q_l})$
    - if $rk(g) < l \leq rk(f)$, then $\forall t \in T_\Sigma : t \otimes \perp \in T(\mathscr{A}_{q_l})$
    - if $rk(f) < l \leq rk(g)$, then $\exists t' \in T_\Gamma : \perp \otimes t' \in T(\mathscr{A}_{q_l})$

    (delay; Out applies a transition, removes the leftmost input and advances in direction of the labeled path ahead; Out can pick output trees for all divergent directions)
  - v) $(q,\pi j f) \to (q,\pi j f j')$ for each $j' \in \{1,\ldots,i\}$ for $f \in \Sigma_i$ if $\|\pi j f\| < k$ (Out delays and chooses a direction from where In should continue)
- The initial vertex is $\{q_0^{\mathscr{A}}\}$.

Note that the game graph can effectively be constructed, because Lemma 2 implies that it is decidable whether the edge constraints are satisfied.

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 $\mathscr{G}^k_{\mathscr{A}} = (G^k_{\mathscr{A}}, V \setminus B)$ as safety game for Out.

Example 4 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 that contains exactly the pairs of trees $(t,t') \in T_\Sigma \times T_\Gamma$ such that t and t' have the same domain and on every path through t' occurs an f if $t \neq a$.

It is easy to see that D↓TA $\mathscr{A} = (\{q_0,q,q_f\}, \Sigma \times \Gamma, q_0, \Delta_{\mathscr{A}})$ with $\Delta_{\mathscr{A}} = \{(q_0,(a,b)), (q_0,(f,f),q_f,q_f), (q_0,(f,g),q,q), (q,(f,f),q_f,q_f), (q,(f,g),q,q), (q_f,(a,b)), (q_f,(f,f),q_f,q_f), (q_f,(f,g),q_f,q_f)\}$ recognizes R. For $k = 1$, the corresponding game graph $G^1_{\mathscr{A}}$ 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.

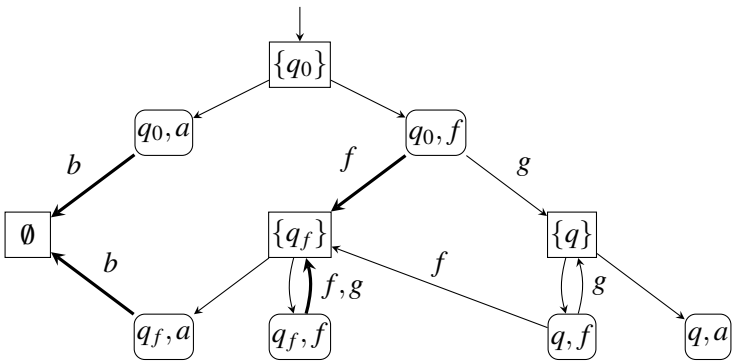

Figure 2: The game graph $G^1_{\mathscr{A}}$ constructed from the D↓TA $\mathscr{A}$ from Example 4. A possible winning strategy for Out in $\mathscr{G}^1_{\mathscr{A}}$ is emphasized in the graph.

Lemma 5 *If Out has a winning strategy in $\mathscr{G}^k_{\mathscr{A}}$, then R has a uniformization by a DTDT in which the output delay is bounded to k.*

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

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

a) For each $\sigma : (q,f) \overset{r}{\mapsto} \{q_1,\dots,q_i\}$ with $r = (q,(f,g),q_1,\dots,q_n) \in \Delta_{\mathscr{A}}$:
   - add $q(f(x_1,\dots,x_i)) \to g(q_1(x_1),\dots,q_j(x_j))$ to Δ if $j \le i$, or
   - add $q(f(x_1,\dots,x_i)) \to g(q_1(x_1),\dots,q_i(x_i),t_{i+1},\dots,t_j)$ to Δ if $j > i$

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

b) For each $\sigma : (q,\pi jf) \mapsto (q,\pi jfj')$ add $(q,\pi j)(f(x_1,\dots,x_i)) \to (q,\pi jfj')(x_{j'})$ 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 $r = (q,(f,g),q_1,\dots,q_n)$ and $q' = q_j$. Then, let $s = g(t_1,\dots,t_{j-1},\circ,t_{j+1},\dots,t_n) \in S_\Sigma$ be the special tree, where each $t_l \in T_\Gamma$ is choosen according to the r-edge constraints in $(q,fj\pi)$ for $l \ne j, 1 \le l \le n$. 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 $(q,\pi jf) \overset{r_1}{\to} \dots \overset{r_{l-1}}{\to} (q',\pi' jf) \to (q',\pi' jfj')$ add $(q,\pi j)(f(x_1,\dots,x_i)) \to s_1 \cdot \ldots \cdot s_{l-1} \cdot (q',\pi' jfj')(x_{j'})$ 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 $(q,\pi jf) \overset{r_1}{\to} \dots \overset{r_{l-1}}{\to} (q',f) \overset{r_l}{\to} \{q_1,\dots,q_i\}$ add $(q,\pi j)(f(x_1,\dots,x_i)) \to s_1 \cdot \ldots \cdot s_{l-1} \cdot s$ 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 $\mathscr{T}$ defines a uniformization of R. Let $t \in T_\Sigma$. We can show by induction on the number of steps needed to reach a configuration from the initial configuration $(t, q_0^{\mathscr{A}}, \varphi_0)$ 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'. Thus, $(t, \mathscr{T}(t)) \in R$.

In $\mathscr{T}$ the output delay is bounded to 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 $\mathscr{T}$ produces output accordingly. □

The size of $G^k_{\mathscr{A}}$ is at most $Q_{\mathscr{A}} \cdot (|\Sigma| \cdot |\mathrm{dir}_\Sigma|)^{k-1} \cdot |\Sigma| + 2^{Q_{\mathscr{A}}}$. For the winning player, a positional winning strategy can be determined in linear time in the size of $G^k_{\mathscr{A}}$ (see Theorem 3.1.2 in [8], which can easily be adapted to safety games). For a positional winning strategy of Out, the above construction yields a DTDT with delay bounded to k that uses at most $Q_{\mathscr{A}} \cdot (|\Sigma| \cdot \mathrm{dir}_\Sigma)^{k-1}$ states.

We now show the other direction.

Lemma 6 *If R has a uniformization by a DTDT in which the output delay is bounded to k, then Out has a winning strategy in $\mathscr{G}^k_{\mathscr{A}}$.*

Proof. Assume that R has a uniformization by some DTDT $\mathscr{T} = (Q, \Sigma, \Gamma, q_0, \Delta)$ in which the output delay in bounded to k. A winning strategy for Out basically takes the moves corresponding to the output sequence that $\mathscr{T}$ produces for a read input sequence induced by the moves of In. We construct the strategy inductively. In a play, a vertex $\big(q, y\big)$ is reached by a sequence of moves that describe a labeled path $xiy \in \mathrm{Path}_\Sigma$ with $x, y \in \mathrm{Path}_\Sigma$, and $i \in \mathrm{dir}_\Sigma$. Let $path(xi) = u$ and $path(xiy) = v$. The strategy in $G^k_{\mathscr{A}}$ can be chosen such that in every play according to the strategy for each reached vertex $\big(q, y\big)$ the following property is satisfied. There is a $t \in T_\Sigma^{xiy}$ such that the deterministic run $\rho_{\mathscr{A}}$ of $\mathscr{A}$ on $t \otimes \mathscr{T}(t)$ yields $\rho_{\mathscr{A}}(u) = q$. Further, if $||y|| > 1$, then there is a configuration (t, t', φ) of $\mathscr{T}$ with $(t, q_0, \varphi_0) \to^*_{\mathscr{T}} (t, t', \varphi)$ reachable such that there is $u \in D_{t'}$ with $\varphi(u) = v$, or there is $u' \in D_{t'}$ with $u \sqsubset u'$ and $\varphi(u') = vj$ for some $j \in \mathrm{dir}_\Sigma$.

We define the strategy as follows. First, we consider the case $y = f \in \Sigma$, i.e., $||y|| = 1$ and $u = v$. For a $t \in T_\Sigma^{xif}$ let $s = t[\circ/u] \cdot f$ be the tree that is obtained by deleting all nodes below u. Then, Out can make her next move according to the outcome of $s \otimes \mathscr{T}(s)$ at node u. If output was produced that is mapped to u, then there has to exist a rule r of the form $(q, val_{t \otimes \mathscr{T}(t)}(u), q_1, \ldots, q_n) \in \Delta_{\mathscr{A}}$ since $\mathscr{T}$ uniformizes R. Also, there has to exist an outgoing r-edge from (q, f) that Out can take. Otherwise, if no output was produced that is mapped to u, then the output is dependent on a node below u. Thus, it must hold that there is a configuration (t, t', φ) with $\varphi(u) = v$ reachable and there exists a successor configuration (t, t'', φ') with $\varphi'(u) = vj$ for some $j \in \mathrm{dir}_\Sigma$. Out delays the output and chooses direction j as next move.

Secondly, we consider the case $||y|| > 1$. There are two possibilities. Either $\mathscr{T}$ reaches a configuration (t, t', φ) with $\varphi(u) = v$ and produces no output in the next configuration step, or $\mathscr{T}$ reaches (t, t', φ) with $\varphi(u') = vj$ for some u' with $u \sqsubset u'$ and some $j \in \mathrm{dir}_\Sigma$, meaning that $\mathscr{T}$ has produced output after reading the input symbol at node v. If no output was produced, Out also delays. This can happen at most k times in a row, since the output delay in $\mathscr{T}$ is bounded to k. Otherwise, since $\mathscr{T}$ uniformizes R, there has to exist a rule r of the form $(q, val_{t \otimes \mathscr{T}(t)}(u), q_1, \ldots, q_n) \in \Delta_{\mathscr{A}}$. We show that there also exists an r-labeled edge outgoing from (q, y) that Out can choose. We have to prove that the r-edge constraints of type iv) are satisfied. Let $val_{t \otimes \mathscr{T}(t)}(u) = (f, g) \in \Sigma \times \Gamma_\perp$. Consider $l \neq j$ with $l \in \{1, \ldots, n\}$. There are three possibilities for l. First, if $l \leq rk(f), rk(g)$, then $t_l \otimes \mathscr{T}(t)|_{ul} \in T(\mathscr{A}_{ql})$ for all $t_l \in T_\Sigma$. Assume that is not true, then there exists $t'_l \in T_\Sigma$ such that $t'_l \otimes \mathscr{T}(t)|_{ul} \notin T(\mathscr{A}_{ql})$. Since $dom_{t|ul} \cap dom_{t|v} = \emptyset$, we can pick $t' = t[\circ/ul] \cdot t'_l$ and obtain $\mathscr{T}(t) = \mathscr{T}(t')$. Thus, $t'_l \otimes \mathscr{T}(t)|_{ul} \in T(\mathscr{A}_{ql})$ which is a contradiction. Secondly, if $rk(g) < l \leq rk(f)$, then $t_l \otimes \perp \in T(\mathscr{A}_{ql})$ for all $t_l \in T_\Sigma$ and thirdly, if $rk(g) < l \leq rk(f)$, then $\perp \otimes \mathscr{T}(t)|_{ul} \in T(\mathscr{A}_{ql})$. For the correctness, the same argumentation as in the case $l \leq rk(f), rk(g)$ can be applied.

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

As a consequence of Lemma 5 and Lemma 6 together with the fact that a winning strategy for Out can effectively be computed in $\mathcal{G}^k_{\mathscr{A}}$ we immediately obtain Theorem 3.

4 Unbounded Delay

Previously, we considered the question whether there exists a uniformization without input validation of a D↓TA-recognizable relation with D↓TA-recognizable domain such that the output delay is bounded. In this section, we will show, that this question is also decidable if the output delay is unbounded. Similar to [2] 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 DTDT 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 DTDT $\mathscr{T}$ that uniformizes R such that $\Delta_{\mathscr{T}}$ 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 $t \in T_\Gamma$.

In the following, we will show that there exists a bound on the output delay that we have to consider in order to decide whether a uniformization by a path-recognizable function is possible.

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

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

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

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 $val_x(u) = lbls(x)[i]$ 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 $i \in \mathrm{dir}_\Sigma$, $x \in \mathrm{Path}_\Sigma$, $y \in \mathrm{Path}_\Gamma$ such that $x \otimes y$ is defined, i.e., $path(y) \sqsubseteq path(x)$ or $path(x) \sqsubseteq path(y)$. We define $\rho_{\mathscr{A}} : \mathrm{dir}_\Sigma^* \to Q_{\mathscr{A}}$ to be the partial function with $\rho_{\mathscr{A}}(\varepsilon) = q_0^{\mathscr{A}}$, and for each $u \in dom_{x\otimes y}$: if $q := \rho_{\mathscr{A}}(u)$ is defined and there is a transition $(q, val_{x\otimes y}(u), q_1, \dots, q_i) \in \Delta_{\mathscr{A}}$, then $\rho_{\mathscr{A}}(u.j) = q_j$ for all $j \in \{1,\dots,i\}$. Let $path(x \otimes y) = v$. Shorthand, we write

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

if $q := \rho_{\mathscr{A}}(vi)$ is defined. We write $\mathscr{A} : q_0^{\mathscr{A}} \xrightarrow{x\otimes y} F_{\mathscr{A}}$ if $(\rho_{\mathscr{A}}(v), val_{x\otimes y}(v)) \in \Delta_{\mathscr{A}}$ to indicate that the (partial) run $\rho_{\mathscr{A}}$ of $\mathscr{A}$ on $x \otimes y$ is accepting.

Sometimes it is sufficient to consider only the output that is mapped to a certain path. For an input tree $t \in T_\Sigma$ or $t \in S_\Sigma$ and a path $u \in \mathrm{dir}_\Sigma^*$, we define

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

to be the set of labeled paths through the output tree $\mathscr{T}(t)$ along u. Note, that if $\mathscr{T}$ is deterministic and $||\mathscr{T}(t)||^u < |u|$, then $out_{\mathscr{T}}(t,u)$ is a singleton set.

We introduce a partial function that yields the state transformations on a path π induced by the input sequence of π together with some output sequence on the same path of same or smaller length. Formally, for $x \in \mathrm{Path}_\Sigma$, $y \in \mathrm{Path}_\Gamma$ and a direction $i \in \mathrm{dir}_\Sigma$ such that $path(y) \sqsubseteq path(x)$, we define the partial function $\tau_{xi,y} : Q_{\mathscr{A}} \to Q_{\mathscr{A}}$ with $\tau_{xi,y}(q) := q'$ if $\mathscr{A} : q \xrightarrow{x \otimes y}_i q'$ and for each uj with $u \in dom_x : uj \not\sqsubseteq path(xi)$ and $j \in \{1, \ldots, rk\big((val_x^\perp(u), val_y^\perp(u))\big)\}$ holds

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

where $\rho_{\mathscr{A}}$ is the run of $\mathscr{A}_q$ on $x \otimes y$. Lemma 2 implies that it is decidable whether $\tau_{xi,y}(q)$ is defined. Basically, if $q' := \tau_{xi,y}(q)$ 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 $t \otimes (s' \cdot t') \in T(\mathscr{A}_q)$.

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 $x \in \mathrm{Path}_\Sigma$ and $i \in \mathrm{dir}_\Sigma$, we define the profile of xi to be $P_{xi} = (P_{xi,=}, P_{xi,<}, P_{xi,\varepsilon})$ with

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

A segment $xi \in (\Sigma \mathrm{dir}_\Sigma)^* \mathrm{dir}_\Sigma$ of a labeled path is called idempotent if $P_{xi} = P_{xixi}$.

As a consequence of Ramsey's Theorem [13], we obtain the next remark.

Remark 7 *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.*

For the rest of this paper we fix how we repeat the part of a tree that contains an idempotent factor in a labeled path segment. Let $x, y \in \mathrm{Path}_\Sigma$, $i, j \in \mathbb{N}$ with $y \neq \varepsilon$ and yj idempotent. For any $t \in T_\Sigma^{xiy}$ we fix t^n to be the tree that results from repeating the idempotent factor n times. More formally, let $path(x) = u$ and $path(y) = v$, we define

$$t^n := \underbrace{t[\circ/ui]}_{s_x} \cdot (\underbrace{t|_{ui}[\circ/uivj]}_{s_y})^n \cdot \underbrace{t|_{uivj}}_{\hat{t}} \text{ for } n \in \mathbb{N}.$$

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

Lemma 8 *For $q \in Q_{\mathscr{A}}$, $x, y \in \mathrm{Path}_\Sigma$, $i, j \in \mathbb{N}$ with $path(x) = u$, $path(y) = v$, $y \neq \varepsilon$ and yj idempotent, it is decidable whether R_q^{xiy} can be uniformized by a path-recognizable function.*

The following Lemma establishes the connection between long output delay and path-recognizable functions. Basically, the lemma states that if there exists a uniformization by a DTDT 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 9 *Let $q \in Q_{\mathscr{A}}$, $x, y \in \mathrm{Path}_\Sigma$, $i, j \in \mathbb{N}$ with $path(x) = u$, $path(y) = v$, $y \neq \varepsilon$ and yj idempotent. If R_q^{xiy} is uniformized by a DTDT $\mathscr{T}$ such that $||\mathscr{T}(s_x \cdot s_y^n)||^{ui(vj)^n} \leq |ui|$ for each $t \in T_\Sigma^{xiy}$ and for each $n \in \mathbb{N}$, then R_q^{xiy} can be uniformized 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.

Now that we have completed all preparations, we present a decision procedure for the case of unbounded delay. Therefore, we consider a similar safety game as in the previous section on uniformization with bounded output delay. We only have to adapt the game graph if the input sequence is ahead K steps. Let $\mathcal{G}_{\mathscr{A}}^{K}$ denote the modified game. From each vertex $(q,\pi) \in V_{\mathsf{Out}}$ with $||\pi|| = K$ we add a move that allows Out to stay in this vertex if there exists a factorization of $\pi = xiyjz$ with $x,y,z \in \mathrm{Path}_\Sigma$, $i,j \in \mathrm{dir}$ and yj is idempotent such that R_q^{xiy} can be uniformized by a path-recognizable function without input validation. These changes to the game graph can be made, because if the input is K steps ahead, then there exists a factorization of the input sequence that contains an idempotent factor and Lemma 8 implies that it is decidable whether there exists a corresponding uniformization by a path-recognizable function.

Lemma 10 *R has a uniformization if, and only if, Out has a winning strategy in the safety game $\mathcal{G}_{\mathscr{A}}^{K}$.*

Proof. Assume that Out has a winning strategy in $\mathcal{G}_{\mathscr{A}}^{K}$, then there also exists a positional winning strategy for Out. To construct a DTDT $\mathscr{T}$ that uniformizes R, we proceed as presented in the proof of Lemma 5 with one addition. We construct for each $(q,\pi) \in V_{\mathsf{Out}}$ such that $||\pi|| = K$ and there is $\pi = xiyjz$ such that R_q^{xiy} can be uniformized by a path-recognizable function, a DTDT $\mathscr{T}_q^{xiy}$ that uniformizes R_q^{xiy}. In $\mathscr{T}$ we switch to $\mathscr{T}_q^{xiy}$ at the respective states.

For the other direction, assume that R is uniformized by some DTDT $\mathscr{T}$. Again, the proof is similar to the proof of Lemma 6. Thus, we only describe how the strategy is chosen if the output delay in $\mathscr{T}$ exceeds K. If the play reaches a vertex $(q,\pi) \in V_{\mathsf{Out}}$ with $||\pi|| = K$, there is a factorization of $\pi = xiyjz$ with $x,y,z \in \mathrm{Path}_\Sigma$, $i,j \in \mathrm{dir}$ such that yj is idempotent. Let $path(x) = u$, $path(y) = v$ and $path(z) = w$. Let $\mathscr{T}_s$ uniformize R_q and pick any $t \in T_\Sigma^\pi$, if $||out_{\mathscr{T}_s}(t^n = s_x \cdot s_y^n \cdot \hat{t}, ui(vj)^n)|| < ui(vj)^n$ for all $n \in \mathbb{N}$, then Lemma 9 implies that R^{xiy} can be uniformized by a path-recognizable function. In this case, Out stays in this vertex from then on and wins.

Otherwise, there exists $m \in \mathbb{N}$ such that $||out_{\mathscr{T}_s}(s_x \cdot s_y^m \cdot \hat{t}, ui(vj)^m)|| \geq ui(vj)^m$. Consider the factorization of $out_{\mathscr{T}_s}(s_x \cdot s_y^m \cdot \hat{t}, ui(vj)^m) = o_1 i o_2 j o_3$ such that $|o_1 i| = |xi|$ and $|o_2 j| = (yj)^m$. Since yj is idempotent we can choose some o of length K such that $\mathscr{A} : q \xrightarrow{xiy \otimes o_1 o}_j q'$ and $\mathscr{A} : q \xrightarrow{xi(yj)^{m-1}y \otimes o_1 o_2}_j q''$ with $q' = q''$. Then Out makes K moves according to o leading to some $(q',z) \in V_{\mathsf{Out}}$. From there, Out takes the transitions according to o_3. □

As a consequence of Lemma 10 and the fact that a winning strategy for Out in $\mathcal{G}_{\mathscr{A}}^{K}$ can effectively be computed we immediately obtain our main result.

Theorem 11 *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 TDT 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.

5 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 DTDT, but every such DTDT 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 12 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 $\{\big(f(b,t), f(t',b)\big) \mid \neg\exists u \in dom_t : val_t(u) = b\}$. Clearly, both R_1 and $dom(R_1)$ are D↓TA-recognizable. Intuitively, a DTDT $\mathscr{T}$ that uniformizes R_1 must read the whole right subtree $t|_2$ of an input tree t to verify that there is no occurrence of b. If an f in $t|_2$ is read and no output is produced, a DTDT can either continue to read left or right, but cannot verify both subtrees. Therefore, in order to verify $t|_2$, a DTDT has to produce an output symbol at each read inner node which results in an output tree of the same size. Clearly, the relation R_1 is uniformized by the following DTDT $\mathscr{T} = (\{q_0,q_1,q_2\},\Sigma,\Sigma,q_0,\Delta)$ with $\Delta =$

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

However, there exists no DTDT $\mathscr{T}'$ that uniformizes R_1 such that the read input sequence and the produced output are on the same path. Assume such a DTDT $\mathscr{T}'$ 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 $\mathscr{T}'_{q_2}$ must induce the relation $\{(t,b) \mid t \in T_\Sigma \wedge \neg\exists u \in dom_t : val_t(u) = b\}$. The only output that $\mathscr{T}'_{q_2}$ 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, $dom(R(\mathscr{T}'_{q_2}))$ 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 DTDT only contains rules of the form $q(f(x_1,\ldots,x_i)) \to w[q_1(x_{j_1}),\ldots,q_n(x_{j_n})]$, where $w \in \Gamma(X_n)$ i.e., read input symbol and correspondingly produced output begin always on the same tree level, it is possible to adapt the presented decision procedure from Section 3. We refer to this kind of DTDTs as DTDTs without delay.

Theorem 13 *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 $\mathscr{A}$ be a D↓TA for a relation and $\mathscr{B}$ be a D↓TA for its domain. The main differences to the previous section is that the vertices in the game graph keep track of the current state of $\mathscr{B}$ on the input sequence played by In and keep track of the state of $\mathscr{A}$ on the combined part of all possible input sequences and the current output sequence of Out which is not necessarily the same as the input sequence played by In. The move constraints for Out will be chosen such that it is guaranteed that the input sequence is valid, and the combined part of all possible input sequences together with her output sequence is valid. Details for this construction can be found in [14].

6 Conclusion

In this paper, we focused on synthesis of deterministic top-down tree transducers from deterministic top-down tree automaton-definable specifications. We have shown that is decidable whether a 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 the future, we want to investigate synthesis of tree transducers from general non-deterministic tree relations. It looks promising to use guidable tree automata [11] for the specifications. It seems that the presented decision procedure remains valid in case of uniformization without input validation.

References

[1] J. Büchi & L. Landweber (1969): *Solving sequential conditions by finite-state strategies*. *Transactions of the American Mathematical Society*. Available at `http://dx.doi.org/10.1090/S0002-9947-1969-0280205-0`.

[2] A. Carayol & C. Löding (2012): *Uniformization in Automata Theory*. To appear in: Logic, Methodology and Philosophy of Science. Proceedings of the Fourteenth International congress. P. Schroeder-Heister, G. Heinzmann, W. Hodges, P. Edouard Bour, eds., London: College Publications.

[3] A. Church (1962): *Logic, arithmetic and automata*. In: *Proceedings of the international congress of mathematicians*, pp. 23–35.

[4] Hu. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison & M. Tommasi (2007): *Tree Automata Techniques and Applications*. Available at `http://www.grappa.univ-lille3.fr/tata`. Release October, 12th 2007.

[5] J. Engelfriet (1978): *On Tree Transducers for Partial Functions*. *Inf. Process. Lett.* 7(4), pp. 170–172. Available at `http://dx.doi.org/10.1016/0020-0190(78)90060-1`.

[6] F. Gèsceg & M. Steinby (1984): *Tree automata*. *Akademiai Kiado*.

[7] E. Grädel, W. Thomas & T. Wilke, editors (2002): *Automata, Logics, and Infinite Games*. *Lecture Notes in Compter Science* 2500, Springer.

[8] Erich Grädel (2007): *Finite Model Theory and Descriptive Complexity*. In: *Finite Model Theory and Its Applications*, Springer, pp. 125–230. Available at `http://dx.doi.org/10.1007/3-540-68804-8_3`.

[9] M. Holtmann, Ł. Kaiser & W. Thomas (2010): *Degrees of lookahead in regular infinite games*. In: *Foundations of Software Science and Computational Structures*, Springer, pp. 252–266. Available at `http://dx.doi.org/10.1007/978-3-642-12032-9_18`.

[10] F. Hosch & L. Landweber (1972): *Finite Delay Solutions for Sequential Conditions*. In: *ICALP*, pp. 45–60.

[11] C. Löding (2009): *Logic and automata over infinite trees*. Habilitation Thesis, RWTH Aachen, Germany.

[12] T. Milo, D. Suciu & V. Vianu (2003): *Typechecking for XML transformers*. *J. Comput. Syst. Sci.* 66(1), pp. 66–97. Available at `http://dx.doi.org/10.1016/S0022-0000(02)00030-2`.

[13] F.P. Ramsey (1930): *On a problem of formal logic*. *Proceedings of the London Mathematical Society* 2(1), p. 264. Available at `http://dx.doi.org/10.1007/978-0-8176-4842-8_1`.

[14] S. Winter (2013): *Uniformization of Automaton Definable Tree Relations*. Masterthesis, RWTH Aachen, Germany.