RWTH Aachen University, Germany
Chair of Computer Science 7 – Logic and Theory of Discrete Systems
Prof. Dr. Martin Grohe, Prof. Dr. Dr.h.c. Wolfgang Thomas

Master Thesis

Uniformization of Automaton Definable Tree Relations

Revised Edition

Sarah Winter

Supervisors:
Priv.-Doz. Dr.rer.nat. Christof Löding
Prof. Dr. Dr.h.c. Wolfgang Thomas

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständing verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, den 28. November 2013

Abstract

We present results on uniformization of automatic tree relations by automaton definable functions. We look at two settings of uniformization by automata. We consider the case that an automaton is not required to verify the correctness of the input, and the case that an automaton is required to validate the input.

Concerning the first setting, we show that it is decidable whether relations recognized by deterministic top-down tree automata have a uniformization in the class of top-down tree transformations.

In the second setting, we present our results regarding the uniformization of deterministic top-down tree automata definable relations by top-down tree transducers and also the uniformization of bottom-up tree automata definable relations by bottom-up tree transducers.

Contents

1 Introduction **1**

2 Preliminaries **3**
- 2.1 Ranked Trees . . . 4
- 2.2 Automata on Ranked Trees . . . 5
- 2.3 Games . . . 9

3 Uniformization of Tree-Automatic Relations **13**
- 3.1 Uniformization Within the Same Class . . . 13

4 Uniformization by Top-Down Tree Transducers **17**
- 4.1 Top-Down Tree Transducer . . . 17
- 4.2 A Restricted Uniformization Case . . . 19
- 4.3 Notations and Definitions . . . 23
- 4.4 Uniformization Without Input Validation . . . 25
  - 4.4.1 Bounded Output Delay . . . 27
  - 4.4.2 Unbounded Output Delay . . . 38
- 4.5 Uniformization with Input Validation . . . 48
  - 4.5.1 Synchronous Input and Output . . . 49

5 Uniformization by Bottom-Up Tree Transducers **61**
- 5.1 Bottom-Up Tree Transducer . . . 61
- 5.2 A Restricted Uniformization Case . . . 62

6 Conclusion **67**

Bibliography **69**

Chapter 1

Introduction

A uniformization of a binary relation is a function that maps each element of the domain to a unique element of the image such that the pair is part of the relation. In connection to automata theory, relations and functions from classes defined by finite automata are considered. Uniformization problems which arise are:

- Investigate whether each relation R that comes from a class $\mathcal{C}$ has a uniformization in a class $\mathcal{C}'$.

- Given a definition of a relation R that comes from a class $\mathcal{C}$, decide whether R has a uniformization in a class $\mathcal{C}'$, and if possible construct one which realizes R in $\mathcal{C}'$.

The study of these questions can be motivated by viewing relations as specifications that relate input to allowed outputs. A uniformization of a relation can then be seen as a form of automatic program synthesis from specifications. A well-known example is Church's synthesis problem [Chu62], for a given input/output specification the question is asked whether there is a circuit which realizes the specification.

These questions have been studied in different settings. Concerning automatic relations as specifications, and the question whether an automatic relation has a uniformization in the same class, it was shown that automatic relations over finite and infinite words have automatic uniformizations [Sie75, CG99]. For finite trees the uniformization result was obtained by [CL07, KW11], but can also be obtained from earlier results in [Eng78]. In a variant of this setting, where it is asked for a uniformization in another class, in [BL69] it was shown for the case of infinite words, that it is decidable whether an ω-automatic specification admits a uniformization by synchronous deterministic sequential transducers. For finite words, it was shown in [CL12] that is it decidable whether an automatic relation has a uniformization by a subsequential transducer.

The aim of this thesis is to explore uniformization of tree-automatic relations by tree transducers.

Outline

Following this introduction, in Chapter 2 we introduce notations and definitions on trees, tree automata and games that we will use throughout this thesis.

In Chapter 3 we present a solution for the uniformization problem for automatic tree relations in the same class that is derived from a result of [Eng78].

In the next two chapters we are concerned with uniformization of tree-automatic relations in different classes.

In Chapter 4 we investigate whether the decision problem corresponding to the question whether a given deterministic top-down tree automata-recognizable relation has a uniformization is decidable in different settings. We are looking for a uniformization in the class of top-down tree transformations. Before turning to more general cases, we show that it is decidable whether a transducer can realize a uniformization of a relation by simply exchanging each input symbol with an output symbol in a given tree. In the extension of this restricted case, two distinctions are made. Primarily, we study the setting where a top-down tree transducer implementing a uniformization is not required to validate the input. In this setting, 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. We show that it is decidable whether a given relation has a uniformization in the class of top-down tree transformations. In the setting where a top-down tree transducer implementing a uniformization is required to validate the input we study the case where the transducer defining the uniformization produces one output symbol for each read input symbol.

In Chapter 5 we study a similar setting in case of automatic tree relations to the restricted setup studied in the previous chapter concerning the uniformization by bottom-up tree transducers.

Finally, in Chapter 6 we conclude this thesis by summarizing our results and give some ideas for future research based on the work of the previous chapters.

Acknowlegements

I would like to thank my supervisor Dr. Christof Löding for suggesting this topic and guiding me in writing this thesis.

Furthermore, I would like to thank Dr. Christof Löding and Prof. Wolfgang Thomas for reviewing my thesis.

Chapter 2

Preliminaries

In this chapter, we formalize the concepts used in this thesis and state some results. In the following sections, we present basic notations and definitions for trees, tree automata, and games. Beforehand, we fix some general notations.

General Notations

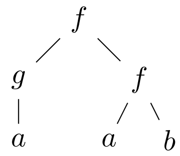
The set of natural numbers containing zero is denoted by $\mathbb{N}$. For a set S, the powerset of S is denoted by 2^S.

An *alphabet* is a finite non-empty set of symbols, called *letters*. Alphabets are usually denoted by Σ or Γ. A *word* is a finite or infinite sequence of letters. The set of finite words (resp. finite non-empty words) over Σ is denoted by Σ^* (resp. Σ^+). The set of infinite words over Σ is denoted by Σ^ω. 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., $w[i, j] = a_i \dots a_j$. A subset $L \subseteq \Sigma^*$ is called *language* over Σ.

The *concatenation* of two words u and v is the word $u \cdot v$, usually denoted uv. Furthermore, the *concatenation* of two languages $U \subseteq \Sigma^*$ and $V \subseteq \Sigma^*$ is the set $U \cdot V := \{uv \mid u \in U, v \in V\}$, and the *complement* of U is the set $\Sigma^* \setminus U$, shortly written as $\overline{U}$.

For all words $u, v \in \Sigma^*$, u is a *prefix* of v if there exists $w \in \Sigma^*$ such that $v = uw$. Given two words u and v, the *greatest common prefix* of u and v, denoted by $gcp(u, v)$, is the longest word that is a prefix of both u and v.

For a language $L \subseteq \Sigma^*$, let $L^0 := \{\varepsilon\}$ and L^{n+1} is recursively defined by $L^{n+1} := LL^n$. The *Kleene closure* of L is the language $L^* := \bigcup_{n \geq 0} L^n$. The set of *regular expressions* over Σ is built up inductively. Atomic expressions are $\emptyset$, a for all $a \in \Sigma$, and ε. For regular expressions r and s, the expressions $r \cdot s$, $r + s$, and r^* are also regular expressions denoting the concatenation, the union and the Kleene closure, respectively. Given a regular expression r, let $L(r)$ denote the language induced by r.



$f(g(a), f(a, b))$

graphical notation term notation

Figure 2.1: Representations of the tree from Example 2.1.

2.1 Ranked Trees

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 ranked alphabet Σ can be specified by the set $\bigcup_{i=0}^{m} \Sigma_i$ where m is the maximum arity in Σ.

Let $\Sigma = \bigcup_{i=0}^{m} \Sigma_i$ be a ranked alphabet. A finite Σ-*labeled ranked tree*, or tree for short, is a pair $t = (\mathrm{dom}_t, \mathrm{val}_t)$ with a set $\mathrm{dom}_t \subseteq \{1, \ldots, m\}^*$ and a mapping $\mathrm{val}_t : \mathrm{dom}_t \to \Sigma$, where

- dom_t is non-empty and finite with
  - dom_t is prefix-closed, i.e., if $ui \in \mathrm{dom}_t$, then $u \in \mathrm{dom}_t$
  - if $ui \in \mathrm{dom}_t$, then $uj \in \mathrm{dom}_t$ for all $1 \leq j < i$

  for all $u \in \mathbb{N}^*$ and $i \in \mathbb{N}$, and
- $\mathrm{val}_t(u) \in \Sigma_i$, if $u \in \mathrm{dom}_t$ has exactly i successors, i.e., $u1, \ldots, ui \in \mathrm{dom}_t$ and $u(i+1) \notin \mathrm{dom}_t$.

The set dom_t is called tree domain and each element in dom_t is called node. A leaf is a node u such that $\forall i \in \mathbb{N}, ui \notin \mathrm{dom}_t$. The root is the node ε. For $u, v \in \mathrm{dom}_t$ we speak of v as ith successor of u, if there is $i \in \mathbb{N}$ such that $v = ui$. Naturally, the tree domain is equipped with a prefix relation $\sqsubseteq$ (resp. strict prefix relation $\sqsubset$) on nodes, where $u \sqsubseteq v$ (resp. $u \sqsubset v$) for $u, v \in \mathrm{dom}_t$ if and only if there is $w \in \mathbb{N}^*$ (resp. $w \in \mathbb{N}^+$) such that $v = uw$.

The set of all Σ-labeled trees (also called trees over Σ) is denoted by T_Σ. A subset $T \subseteq T_\Sigma$ is called *tree language* over Σ. The *complement* of T is $T_\Sigma \setminus T$ written as $\overline{T}$.

The following example illustrates the tree representations used in this thesis.

Example 2.1 Given a ranked alphabet Σ by $\Sigma_2 = \{f\}$, $\Sigma_1 = \{g\}$, and $\Sigma_0 = \{a, b\}$. Let t be a ranked tree with $\mathrm{dom}_t = \{\varepsilon, 1, 2, 11, 21, 22\}$ and $\mathrm{val}_t(\varepsilon) = \mathrm{val}_t(2) = f, \mathrm{val}_t(1) = g, \mathrm{val}_t(11) = \mathrm{val}_t(21) = a$, and $\mathrm{val}_t(22) = b$. The graphical and the term representation are shown in Figure 2.1.

The *height* h of a tree t is the length of a longest path through t, i.e., $h(t) = max\{|p| \mid p \in \mathrm{dom}_t\}$.

A *subtree* $t|_u$ of a tree t at node u is defined by

- $\text{dom}_{t|_u} = \{i \in \mathbb{N}^* \mid ui \in \text{dom}_t\}$
- $\text{val}_{t|_u}(v) = \text{val}_t(v)$ for all $v \in \text{dom}_{t|_u}$

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 $u \in \text{dom}_t$, 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.

In Chapters 4 and 5 on uniformization by tree transducers we need the following notations.

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. Let $X = \bigcup_{n>0} X_n$. 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. 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 $C[x_1 \leftarrow t_1, \dots, x_n \leftarrow t_n]$.

2.2 Automata on Ranked Trees

A word language $L \subseteq \Sigma^*$ is called *regular* if L is recognizable by a finite automaton on words, see e.g. [HMU01] for an introduction to finite automata.

A *non-deterministic finite automaton* (NFA) over an alphabet Σ is of the form $\mathcal{A} = (Q, \Sigma, \Delta, q_0, F)$, where Q is a finite set of states, $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. A *run* of $\mathcal{A}$ on a word $w = a_1 \dots a_n \in \Sigma^*$ is a sequence of states $\rho = \rho_0 \dots \rho_n$, such that $\rho_0 = q_0$ and $(\rho_i, a_{i+1}, \rho_{i+1}) \in \Delta$ for $0 \leq i < n$. The run is successful if $\rho_n \in F$. A word $w \in \Sigma^*$ is accepted by $\mathcal{A}$ if, and only if, there exists a successful run of $\mathcal{A}$ on w. The language recognized by $\mathcal{A}$ is $L(\mathcal{A}) = \{w \in \Sigma^* \mid \mathcal{A} \text{ accepts } w\}$. An NFA $\mathcal{A} = (Q, \Sigma, \Delta, q_0, F)$ is *deterministic* (a DFA) if for each $a \in \Sigma$ and each $q \in Q$ there exists at most one transition $(q, a, q') \in \Delta$.

Tree automata can be viewed as a straightforward generalization of finite automata on finite words, when words are interpreted as trees over unary symbols. In the following we present bottom-up and top-down tree automata as well as some results for tree automata used in this thesis. For a detailed introduction to tree automata see e.g. [GS84] or [CDG+07].

Let $\Sigma = \bigcup_{i=1}^m \Sigma_i$ be a ranked alphabet. A *non-deterministic tree automaton* (NTA) over Σ is a tuple $\mathcal{A} = (Q, \Sigma, \Delta, F)$, where Q is a finite set of states,

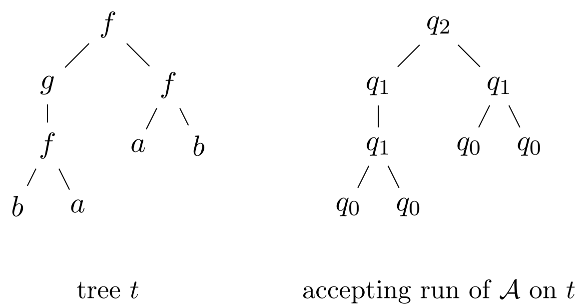


Figure 2.2: A run of the NTA $\mathcal{A}$ from Example 2.2.

$F \subseteq Q$ is a set of final states, and $\Delta \subseteq \bigcup_{i=0}^{m}(Q^i \times \Sigma_i \times Q)$ is the transition relation.

Let t be a tree and $\mathcal{A}$ be an NTA, a *run* of $\mathcal{A}$ on t is a mapping $\rho : \text{dom}_t \to Q$ compatible with Δ, i.e., for each node $u \in \text{dom}_t$, if $\text{val}_t(u) \in \Sigma_i$, then $(\rho(u1), \ldots, \rho(ui), \text{val}_t(u), \rho(u)) \in \Delta$. The run is *successful* if $\rho(\varepsilon)$ is a final state. A tree $t \in T_\Sigma$ is *accepted* if, and only if, there is a successful run of $\mathcal{A}$ on t. The tree language *recognized* by $\mathcal{A}$ is $T(\mathcal{A}) = \{t \in T_\Sigma \mid \mathcal{A} \text{ accepts } t\}$.

Example 2.2 Let Σ be a ranked alphabet given by $\Sigma_2 = \{f\}$, $\Sigma_1 = \{g\}$, and $\Sigma_0 = \{a, b\}$. Consider the NTA $\mathcal{A} = (\{q_0, q_1, q_2\}, \Sigma, \Delta, \{q_2\})$ with $\Delta =$

$$\begin{array}{rl} & \{(a, q_0), (b, q_0)\} \\ \cup & \{(q_i, g, q_i) \mid 0 \leq i \leq 2\} \\ \cup & \{(q_i, q_j, f, q_{min(2,i+j+1)})\}. \end{array}$$

The recognized tree language is the set of all trees with at least two occurrences of f. A tree t and an accepting run of $\mathcal{A}$ on t is depicted in Figure 2.2.

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

A special case of NTAs are deterministic tree automata (DTAs). A tree automaton $\mathcal{A} = (Q, \Sigma, \Delta, F)$ is *deterministic* if there are no two ambiguous rules. That is, for each $f \in \Sigma_i$ and each $q_1, \ldots, q_i \in Q$ there is at most one transition $(q_1, \ldots, q_i, f, q) \in \Delta$. In other words, each run of $\mathcal{A}$ on a tree t is unique. For instance, consider the automaton from Example 2.2 that is in fact deterministic. It should be noted, that for every NTA there exists an equivalent DTA that recognizes the same tree language. As for finite automata on words, the subset construction is used for determinization.

Sometimes it is convenient to consider tree automata in which the state set only consists of *productive* states, i.e., for an NTA $\mathcal{A}$ with state set Q, each $q \in Q$ is reachable and for each $q \in Q$ there exists a tree t such that t is accepted by $\mathcal{A}$ starting from q. Also, in some cases it is useful to consider tree automata such that there exists at least one run for every tree. Tree automata that fulfill this property are called *complete*.

Another automaton model that also recognizes the class of regular tree languages are top-down tree automata.

Let $\Sigma = \bigcup_{i=1}^{m} \Sigma_i$ be a ranked alphabet. A *non-deterministic top-down tree automaton* (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. The pairs $\Delta \cap (Q \times \Sigma_0)$ are called final combinations.

Let t be a tree and $\mathcal{A}$ be an N↓TA, a *run* of $\mathcal{A}$ on t is a mapping $\rho : \mathrm{dom}_t \to Q$ compatible with Δ, i.e., $\rho(\varepsilon) \in Q_0$ and for each node $u \in \mathrm{dom}_t$, if $\mathrm{val}_t(u) \in \Sigma_i$ with $i > 0$, then $(\rho(u), \mathrm{val}_t(u), \rho(u1), \ldots, \rho(ui)) \in \Delta$. A run is *successful* if for each leaf u, the pair $(\rho(u), \mathrm{val}_t(u))$ is a final combination. A tree $t \in T_\Sigma$ is *accepted* if, and only if, there is a successful run of $\mathcal{A}$ on t. The tree language *recognized* by $\mathcal{A}$ is $T(\mathcal{A}) = \{t \in T_\Sigma \mid \mathcal{A} \text{ accepts } t\}$.

It is easily seen that NTAs and N↓TAs are equivalent, by reversing the transitions and exchanging final and initial states one obtains an N↓TA from an NTA and vice versa.

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 $(q, f, q_1, \ldots, q_i) \in \Delta$.

However, unlike bottom-up tree automata, non-deterministic and deterministic top-down automata are not equally expressive. There are regular tree languages that cannot be recognized by a D↓TA. For example, consider the tree language $T = \{f(a, b), f(b, a)\}$ which is obviously regular, but not recognizable by a D↓TA. A D↓TA accepting $f(a, b)$ and $f(b, a)$ would also accept $f(a, a)$, thus there exists no D↓TA that recognizes T.

Intuitively, the class of D↓TA-recognizable tree languages is exactly the class of *path-closed languages*, see [Vir80]. Path-closure is defined as follows. The path language $\pi(t)$ of a tree $t \in T_\Sigma$ is defined inductively by:

- if $t \in \Sigma_0$, then $\pi(t) = t$, and
- if $t = f(t_1, \ldots, t_i) \in T_\Sigma$, then $\pi(t) = \bigcup_{j=1}^{j=i}\{fjw \mid w \in \pi(t_j)\}$.

Let T be a tree language over Σ, then $\pi(T) = \bigcup_{t \in T} \pi(t)$, the path-closure of T is defined by $\mathit{pathclosure}(T) = \{t \in T_\Sigma \mid \pi(t) \subseteq \pi(T)\}$. A tree language is path-closed if $\mathit{pathclosure}(T) = T$.

In [GS84] it was shown that it is decidable whether a regular tree language can be recognized by a deterministic top-down tree automaton. We present an alternative proof of this result using closure properties of finite tree automata.

Theorem 1 *It is decidable whether a regular tree language is D↓TA-recognizable.*

Proof. Let $\mathcal{A} = (Q, \Sigma, Q_0, \Delta)$ be a N↓TA. We assume that all states of $\mathcal{A}$ are productive. We define the D↓TA $\mathcal{A}' = (2^Q, \Sigma, \{Q_0\}, \Delta')$ by a top-down subset construction with:

- $(R, f, R_1, \ldots, R_i) \in \Delta'$ if $R_j = \{r_j \mid \exists r \in R \text{ with } (r, f, r_1, \ldots, r_i) \in \Delta\}$ for each $j \in \{1, \ldots, i\}$,

- $(R, a) \in \Delta'$ if there exists $r \in R$ with $(r, a) \in \Delta$.

We show that $T(\mathcal{A}') = T(\mathcal{A})$ holds if, and only if, $T(\mathcal{A})$ is D↓TA-recognizable. Clearly, if $T(\mathcal{A}') = T(\mathcal{A})$, then $T(\mathcal{A})$ is D↓TA-recognizable.

For the other direction, assume there exists a complete D↓TA $\mathcal{B} = (Q'', \Sigma, \{q_0\}, \Delta'')$ with $T(\mathcal{B}) = \mathcal{T}(A)$. We first show that $T(\mathcal{A}') \supseteq T(\mathcal{A})$. Let t be a tree and let ρ be a run of $\mathcal{A}$ on t and ρ' be the unique run of $\mathcal{A}'$ on t. We prove by induction over the height of a node u in a tree t that if $\rho(u) = r$, then $\rho'(u) = R$ with $r \in R$.

If the height of the node is 0, then $u = \varepsilon$. We have $\rho(\varepsilon) = q \in Q_0$, $\rho'(\varepsilon) = Q_0$ with $q \in Q_0$.

Let $n{+}1$ be the height of a node u' in the tree, assume that the claim holds for n. Let $u' = uj$ with $j \in \{1, \ldots, rk(\mathrm{val}_t(u))\}$, $\rho(u) = r$ and $\rho(u') = r_j$, then there exists $(r, \mathrm{val}_t(u), r_1, \ldots, r_i) \in \Delta$. By induction hypothesis it follows that $\rho'(u) = R$ with $r \in R$ and by construction of $\mathcal{A}'$ there exists $(R, \mathrm{val}_t(u), R_1, \ldots, R_i) \in \Delta'$ with $r_j \in R_j$. Hence $\rho'(u') = R_j$ with $r_j \in R_j$.

Let $t \in T(\mathcal{A})$. There exists an accepting run ρ of $\mathcal{A}$ on t, i.e., for each leaf $u \in \mathrm{dom}_t$ holds $(\rho(u), \mathrm{val}_t(u)) \in \Delta$. Consequently, for the unique run ρ' of $\mathcal{A}'$ on t holds for each leaf u that $(\rho'(u), \mathrm{val}_t(u)) \in \Delta'$, since $\rho(u) \in \rho'(u)$. Hence, $t \in T(\mathcal{A}')$.

Towards a contradiction, assume that there is a tree $t \in T(\mathcal{A}') \setminus T(\mathcal{B})$. We now construct a tree t' that is accepted by $\mathcal{A}$ and rejected by $\mathcal{B}$. Let ρ' resp. ρ'' denote the unique run of $\mathcal{A}'$ resp. $\mathcal{B}$ on t. Since $t \in T(\mathcal{A}') \setminus T(\mathcal{B})$, there exists a leaf $w \in \mathrm{dom}_t$ with $\mathrm{val}_t(w) = a$, $\rho''(w) = q$ and $\rho'(w) = R$ such that (q, a) is not accepted in $\mathcal{B}$, but (r, a) is accepted in $\mathcal{A}$ for some $r \in R$ $(*)$. Thus, $t|_u \in T(\mathcal{A}'_{\rho'(u)}) \setminus T(\mathcal{B}_{\rho''(u)})$ holds for each $u \sqsubseteq w$.

Let $w = uv$, $\rho''(u) = q$ and $\rho'(u) = R$. We show by induction on the length of v that there exists $r \in R$ such that there exists $t' \in T(\mathcal{A}_r) \setminus T(\mathcal{B}_q)$.

For the induction base let $|v| = 0$, then $u = w$. Let $t' = a$, then it follows directly from $(*)$ that $a \in T(\mathcal{A}_r) \setminus T(\mathcal{B}_q)$ for some $r \in R$.

For the induction step consider $|v| = n{+}1$. Assume the claim holds for n. Let $v = jv'$ with $\mathrm{val}_t(u) = f$ and $j \in \{1, \ldots, rk(f)\}$ and let the runs ρ' and ρ'' result in $\rho''(u) = q$, $\rho''(uj) = q_j$, $\rho'(u) = R$ and $\rho'(uj) = R_j$. From the existence of the run it follows that there are $(q, f, q_1, \ldots, q_i) \in \Delta''$ and $(R, f, R_1, \ldots, R_i) \in \Delta'$. From the induction hypothesis we know that there exists some $r_j \in R_j$ such that there exists a $t'_j \in T(\mathcal{A}_{r_j}) \setminus T(\mathcal{B}_{q_j})$. By construction of $\mathcal{A}'$ there is some $r \in R$ with $(r, f, r_1, \ldots, r_i) \in \Delta$. Since $\mathcal{A}$ only contains productive states there exist trees $t'_1 \in T(\mathcal{A}_{r_1}), \ldots, t'_{j-1} \in T(\mathcal{A}_{r_{j-1}}), t'_{j+1} \in T(\mathcal{A}_{r_{j+1}}), \ldots, t'_i \in T(\mathcal{A}_{r_i})$. Now let $t' = f(t'_1, \ldots, t'_i)$, then $t' \in T(\mathcal{A}'_r) \setminus T(\mathcal{B}_q)$.

Choosing $v = w$, we obtain $\rho''(\varepsilon) = q_0$ and $\rho'(\varepsilon) = Q_0$. This means there exists $t' \in T(\mathcal{A}) \setminus T(\mathcal{B})$, contradicting $T(\mathcal{B}) = \mathcal{T}(A)$.

Since it is decidable whether $\mathcal{A}$ and $\mathcal{A}'$ recognize the same language (cf. e.g. [CDG+07]), it is decidable whether $T(\mathcal{A})$ is D↓TA-recognizable.

□

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 $\Sigma_\perp = \Sigma \dot\cup \{\perp\}$, $\Gamma_\perp = \Gamma \dot\cup \{\perp\}$. 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

- $\mathrm{dom}_t = \mathrm{dom}_{t_1} \cup \mathrm{dom}_{t_2}$, and
- $\mathrm{val}_t(u) = (\mathrm{val}^\perp_{t_1}(u), \mathrm{val}^\perp_{t_2}(u))$ for all $u \in \mathrm{dom}_t$,

where $\mathrm{val}^\perp_{t_i}(u) = \mathrm{val}_{t_i}(u)$ if $u \in \mathrm{dom}_{t_i}$ and $\mathrm{val}^\perp_{t_i}(u) = \perp$ otherwise for $i \in \{1, 2\}$.

We define the *convolution of a tree relation* $R \subseteq T_\Sigma \times T_\Gamma$ to be the tree language $T_R := \{t_1 \otimes t_2 \mid (t_1, t_2) \in R\}$. The definition is easily generalized to tuples of trees, but is not needed within this thesis.

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.

Example 2.3 Let Σ be a ranked alphabet given by $\Sigma_2 = \{f, g\}$, and $\Sigma_0 = \{a, b\}$. Consider the following relation $R \subseteq T_\Sigma \times T_\Sigma$ that contains a pair of trees if there exists a position that has the same label in both trees. Formally,

$$R := \{(t_1, t_2) \mid \exists u \in \mathrm{dom}_{t_1} \cap \mathrm{dom}_{t_2} \text{ with } \mathrm{val}_{t_1}(u) = \mathrm{val}_{t_2}(u)\}.$$

The DTA $\mathcal{A} = (\{q, q_F\}, \Sigma_\perp \times \Sigma_\perp, \Delta, \{q_F\})$ with $\Delta =$

$$\begin{array}{rl} & \{\big((a,a), q_F\big), \big((b,b), q_F\big)\} \\ \cup & \{\big((a,\perp), q\big), \big((\perp,a), q\big), \big((b,\perp), q\big), \big((\perp,b), q\big)\} \\ \cup & \{\big(q, q, (f,f), q_F\big), \big(q, q, (g,g), q_F\big)\} \\ \cup & \{\big(q_1, q_2, (\sigma_1,\sigma_2), q_F\big) \mid (\sigma_1,\sigma_2) \in \Sigma_2 \times \Sigma_2 \text{ and } \{q_F\} \subseteq \{q_1, q_2\}\} \\ \cup & \{\big(q, q, (\sigma_1,\sigma_2), q\big) \mid (\sigma_1,\sigma_2) \in \Sigma_2 \times \Sigma_\perp \cup \Sigma_\perp \times \Sigma_2\} \end{array}$$

recognizes this relation. A pair of trees $(t_1, t_2) \in T_\Sigma \times T_\Sigma$ that belongs to the relation and its convolution $t_1 \otimes t_2$ accepted by $\mathcal{A}$, are depicted in Figure 2.3.

2.3 Games

It is natural to view synthesis problems as infinite games between two players, where one player supplies an input part and the other player reacts with an output part. In this setting, we will refer to the players as player In and player Out. For notational convenience, we assume that player In is male and player Out is female. The games are played on a graph that has a partition of the vertices into two sets; vertices of In, graphically represented as rectangles, and

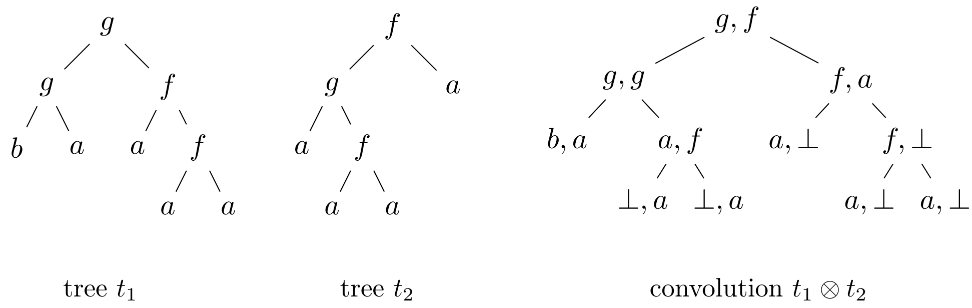


Figure 2.3: The convolution $t_1 \otimes t_2$ is accepted by $\mathcal{A}$ from Example 2.3.

vertices of Out, graphically represented as circles. The players move a token along the edges, if the token is on a vertex belonging to In, he can choose the next edge and vice versa for Out. Together with a winning condition, a game graph is turned into a game.

A finite *game graph* is of the form $G = (V_{\mathsf{In}}, V_{\mathsf{Out}}, E, c)$, where

- V_{In} is the set of vertices of player In,
- V_{Out} is the set of vertices of player Out,
- $E \subseteq V \times V$ with $V = V_{\mathsf{In}} \uplus V_{\mathsf{Out}}$ is the set of edges or moves, and
- $c : V \to C$ is a coloring of the vertices with colors from a finite set of colors C.

A *play* in G is a maximal sequence α of vertices compatible to the edges of the game graph. A sequence α is maximal if it is either infinite or it ends in a vertex without outgoing edges. The players move a token along the edges. In moves the token from vertices in V_{In}, Out moves the token from vertices in V_{Out}.

Given a play $\alpha \in V^*$ or $\alpha \in V^\omega$, the winner of the play is determined by the corresponding sequence $c(\alpha) \in C^*$ or $c(\alpha) \in C^\omega$. Therefore, we have to specify a *winning condition* by a set $Win \subseteq C^\omega \cup C^*$. Out wins a play α if $c(\alpha) \in Win$, otherwise In wins. A *Game* is a pair $\mathcal{G} = (G, Win)$ of a game graph and a winning condition as defined above.

The coloring $c : V \to C$ is only needed in some cases. Hence, if we specify a game graph without a coloring, we implicitly assume that $C = V$ and $c : V \to V$ is the identity function. In particular, in this thesis we do not consider games that need a vertex coloring, but we consider games that start in a specific initial vertex. In these cases, the initial vertex is specified as an additional component of the game graph. Given a game graph with an initial vertex v, a play is then a maximal sequence of vertices starting in v.

An example of a game graph without coloring is shown in Figure 2.4.

To describe a game play we introduce the notion of strategies. A strategy outputs the player's next move given a finite prefix of a play ending in a vertex of said player. A *strategy* for Out is a function

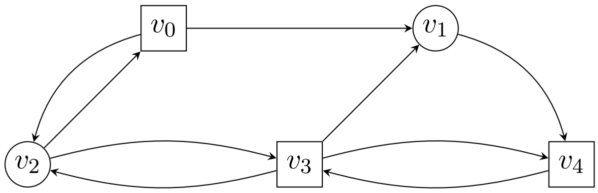


Figure 2.4: A game graph.

$$\sigma_{\mathsf{Out}} : V^* V_{\mathsf{Out}} \to V$$

such that $\sigma_{\mathsf{Out}}(xv) = v'$ with $x \in V^*$, $v, v' \in V$ implies that $(v, v') \in E$.

Given a play α of vertices, let $\alpha(i)$ denote the ith vertex in this sequence starting with 1. A play α is *played according to* σ_{Out} if $\sigma_{\mathsf{Out}}(\alpha(1)\ldots\alpha(i)) = \alpha(i+1)$ for all $i \in \mathbb{N}$ with $\alpha(i) \in V_{\mathsf{Out}}$.

Given a game $\mathcal{G} = (G, \mathit{Win})$ and a vertex $v \in V$. Let $\mathit{Out}(\sigma_{\mathsf{In}}, v_0)$ denote the set of all plays α starting in v. A strategy σ_{Out} for Out is a *winning strategy* from v if $c(\alpha) \in \mathit{Win}$ for all $\alpha \in \mathit{Out}(\sigma_{\mathsf{Out}}, v)$. That is, Out wins every play starting in v, no matter how In plays, if Out plays according to her strategy. The same definitions apply for In with V_{In} instead of V_{Out}.

The set $W_{\mathsf{Out}} \subseteq V$ from which Out has a winning strategy is called *winning region* of Out. Analogously, $W_{\mathsf{In}} \subseteq V$ from which In has a winning strategy is called the winning region of In. A game is called *determined* if we can partition V into the sets $W_{\mathsf{In}}, W_{\mathsf{Out}}$ such that $V = W_{\mathsf{In}} \uplus W_{\mathsf{Out}}$, i.e. from each vertex either In or Out has a winning strategy.

Restricted types of strategies are positional or memoryless strategies, meaning the strategy does not consider the previously seen vertices, but only the current vertex. More formally, a *positional strategy* for Out is a mapping $\sigma_{\mathsf{Out}} : V_{\mathsf{Out}} \to V$ such that $(v, \sigma_{\mathsf{Out}}(v)) \in E$ for all $v \in V_{\mathsf{Out}}$. Similarly for In.

A game is called *positionally determined* if from each vertex one of the players has a positional winning strategy.

In the following chapters, we consider so-called *safety games*. An important property of this type of game is the positional determinacy and the fact that a winning strategy for a player can effectively be computed in a safety game, see e.g. [GTW02].

A winning condition $\mathit{Win} \subseteq C^\omega \cup C^*$ is called a *safety condition* if there is $D \subseteq C$ such that $\alpha \in \mathit{Win}$ if, and only if, only colors from D occurs in α. The objective of Out is to stay inside a safe region of vertices with color D. Therefore, a *safety game* can fully be specified by the game graph $G = (V_{\mathsf{In}}, V_{\mathsf{Out}}, E)$ and a set $F \subseteq V$ of vertices that specify the safe region. The set $V \setminus F$ specifies the region that Out has to avoid in order to win the game.

Chapter 3

Uniformization of Tree-Automatic Relations

As already mentioned in the introduction, a *uniformization* of a relation $R \subseteq X \times Y$ is a function $f_R : X \to Y$ that maps every element of the domain of R to an element of Y that is in relation with this element. That is, a function with the same domain as R and $(x, f_R(x)) \in R$ for all $x \in dom(R)$.

In this chapter we consider the question whether a tree-automatic relation has a uniformization in the same class. More formally, let R be a tree-automatic relation. In this context, the *uniformization problem* is the decision problem whether there exists a uniformization of R whose graph defines a tree-automatic relation.

3.1 Uniformization Within the Same Class

This question was already considered in [CL12] with the result that every tree-automatic relation has a tree-automatic uniformization based on results from [CL07] and [KW11]. There, an equivalence relation is defined that makes it possible to construct a tree automaton that contains exactly one representative from each equivalence class.

However, the result can also be deduced from a proof of a result in [Eng78]. In [Eng78] it is shown that relations induced by non-deterministic top-down tree transducers have a uniformization by a deterministic top-down tree transducer with regular look-ahead. There, for a given ordering of the transitions, the deterministic transducer chooses the smallest transition that admits a successful run whereby the choice can be verified by checking the regular look-ahead.

We will now present a proof for the uniformization result on tree-automatic relations using a similar idea. According to some ordering of the transitions of a ↓TA, we will construct a N↓TA that guesses at each point in a run the smallest possible transition that admits a successful run on the remaining input. The choice is then verified in the remaining part of the run by tracking that no smaller transition would yield a successful run.

Theorem 2 *Every tree-automatic relation has a tree-automatic uniformization.*

Proof. Let $R \subseteq T_\Sigma \times T_\Gamma$ be a tree-automatic relation that is recognized by a N↓TA $\mathcal{A} = (Q, \Sigma_\perp \times \Gamma_\perp, Q_0, \Delta)$. Let $\mathcal{A}_q$ denote the N↓TA with results from $\mathcal{A}$ by replacing Q_0 with $\{q\}$. Our goal is to find a N↓TA that recognizes a relation $R' \subseteq R$ with $dom(R) = dom(R')$ such that there exists for each $t \in dom(R)$ exactly one $t' \in T_\Gamma$ with $(t, t') \in R'$.

Informally, the idea is to order the transitions with the same state and input symbol, and at each point in a run, when it is possible to use a smaller transition, the automaton non-deterministically pursues a path of the smaller run. If the path of the smaller run is also accepting, the automaton rejects the current input-output combination.

We assume that $Q_0 = \{q_0\}$ is a singleton set, which is not a restriction. Otherwise, we introduce a new initial state q_0, and for each transition $(q, (f, g), q_1, \ldots, q_n) \in \Delta$ such that $q \in Q_0$ we add the transition $(q_0, (f, g), q_1, \ldots, q_n)$ to Δ, and replace Q_0 by $\{q_0\}$.

Let $\prec$ denote a partial ordering on the transitions such that for each $q \in Q$ and each $f \in (\Sigma_\perp)_i$, $i \geq 0$ all transitions of the form $(q, (f, g), q_1, \ldots, q_n) \in \Delta$, $g \in (\Gamma_\perp)_j$, $j \geq 0$ and $q_1 \ldots, q_n \in Q$ are comparable. The ordering should fulfill the condition that for two comparable transition $r_1 = (q, (f, g_1), q_1, \ldots, q_m), r_2 = (q, (f, g_2), p_1, \ldots, p_n)$ with $g_1 \in (\Gamma_\perp)_j, g_2 \in (\Gamma_\perp)_k$ holds $r_1 \prec r_2$ if $j < k$. Consider two comparable transitions $r_1, r_2 \in \Delta$. We say r_1 is smaller than r_2 if $r_1 \prec r_2$.

Now we are ready to construct a N↓TA $\mathcal{B}$ that recognizes a uniformization of R. We define $\mathcal{B} = (Q', \Sigma_\perp \times \Gamma_\perp, Q'_0, \Delta')$, where

- $Q' \subseteq Q \times 2^Q$ is the set of states,

  The intuitive meaning of a state $(q, P) \in Q'$ is as follows. Given a node $u \in \mathrm{dom}_t$, consider a run $\rho_\mathcal{A}$ of $\mathcal{A}$ on $t \otimes t'$ consisting of an input tree t and an output tree t' such that $\rho_\mathcal{A}(u) = q$. If we have a run $\rho_\mathcal{B}$ of $\mathcal{B}$ on $t \otimes t'$ with $\rho_\mathcal{B}(u) = (q, P)$ there exists for each $p \in P$ a run $\rho'_\mathcal{A}$ on $t \otimes t''$ for an alternative output t'' with $\rho'_\mathcal{A}(u) = p$ such that $\mathcal{A}$ would have taken a smaller transition at some point $v \sqsubseteq u$ which then would lead to a state p.

- $Q'_0 = \{(q_0, \emptyset)\}$ is the singleton set of initial states, and

- Δ' is the transition relation:

  The new transition relation is built up from Δ with the following idea in mind. In addition to the original run, the automaton keeps track of runs that could have originated from choosing smaller applicable transitions. The automaton followed non-deterministically one path of an alternative smaller run. If a smaller run is not accepting, then the automaton can follow a path that does not lead to a final combination, but if a smaller run is accepting, then each path leads to a final combination. In this case, the automaton does not accept as there exists another possible smaller output.

  This property is captured by the following transition rules.

- Consider $(q, P) \in Q'$ and $(f, g) \in (\Sigma_\perp \times \Gamma_\perp)_i$, $i > 0$ such that $\neg\exists p \in P : (p, (f, g')) \in \Delta$. For each rule $r = (q, (f, g), q_1, \ldots, q_n) \in \Delta$ we add new rules of the form
$$\big((q, P), (f, g), (q_1, P_1), \ldots, (q_n, P_n)\big)$$
to Δ' for all $P_1, \ldots, P_n \subseteq 2^Q$ that fulfill three conditions:
  1. For each $r' = (q, (f, g'), p_1, \ldots, p_m) \in \Delta$ with $r' \prec r$ there is $p_j \in P_j$ for exactly one $j \in \{1, \ldots, m\}$, and
  2. for each $p \in P$ and for each $(p, (f, g'), p_1, \ldots, p_m) \in \Delta$ there is $p_j \in P_j$ for exactly one $j \in \{1, \ldots, n\}$ or there is $j \in \{n + 1, \ldots, rk((f, g'))\}$ with $\perp \notin dom(T(\mathcal{A}_{p_j}))$.

  The first condition ensures that whenever a smaller transition is applicable the automaton choses non-deterministically a path of an alternative smaller run. The second condition ensures that each already followed path is non-deterministically continued. Additionally, each $P_j, 1 \leq j \leq m$ should only contain states that could have originated from a smaller run. This is captured by the third condition.
  3. For each $p_j \in P_j$ for all $1 \leq j \leq m$ there exists either $(q, (f, g'), p_1, \ldots, p_m) \in \Delta$ or $(p, (f, g'), p_1, \ldots, p_m) \in \Delta$ such that $p \in P$.
- For each $(q, P) \in Q'$ we add $\big((q, P), (a, b)\big)$ as final combination to Δ' if and only if $(q, (a, b)) \in \Delta$ and for all $b' \in \Gamma_\perp : (p, (a, b')) \notin \Delta$ for all $p \in P$ and $(q, (a, b)) \prec (q, (a, b'))$ if $(q, (a, b')) \in \Delta$.

Let R' be the relation recognized by $\mathcal{B}$. It is obvious that $R' \subseteq R$ with $dom(R') \subseteq dom(R)$. We show $dom(R) \subseteq dom(R')$. Consider $t \in dom(R)$, let $t_1, \ldots, t_m$ denote the trees with $(t, t_i) \in R$ and accepting runs $\rho^i_\mathcal{A}$ on $t \otimes t_i$ for $1 \leq i \leq m$. We pick t_i from all possible outputs $t_1, \ldots, t_m$ matching t such that on each path through $t \otimes t_i$ holds: Let u be the deepest node on this path such that for all $v \sqsubset u$ holds the same transition was applied at v in all runs, and the applied transition at u in $\rho^i_\mathcal{A}$ was smaller then the applied transition at u in $\rho^j_\mathcal{A}$ for $j \neq i, 1 \leq j \leq m$.

We show by induction on the height of a node $u \in \mathrm{dom}_{t\otimes t_i}$ that there exists a run $\rho_\mathcal{B}$ of $\mathcal{B}$ on $t \otimes t_i$ such that:

$$\begin{aligned}&\forall u \in \mathrm{dom}_{t\otimes t_i} :\\ &\rho_\mathcal{A}(u) = q \Rightarrow \rho_\mathcal{B}(u) = (q, P) \text{ and } \forall p \in P : t_{|u} \notin dom(T(A_p))\end{aligned} \qquad (*)$$

For the induction base, we have $\rho_\mathcal{B}(\varepsilon) = (q_0, \emptyset) \in Q'_0$ and $\rho_\mathcal{A}(\varepsilon) = q_0 \in Q_0$. The claim holds directly.

For the induction step, consider a node u at height n such that the claim holds. Let $\rho^i_\mathcal{A}(u) = q$ and $\mathrm{val}_{t\otimes t_i}(u) = (f, g)$ with $rk(f, g) = k$, by induction hypothesis we obtain that there exists a run $\rho_\mathcal{B}$ with $\rho_\mathcal{B}(u) = (q, P)$ such that $\forall p \in P : t_{|u} \notin dom(T(A_p))$. From the run $\rho^i_\mathcal{A}$ on $t \otimes t_i$ it follows that $\big(q, (f, g), \rho^i_\mathcal{A}(u1), \ldots, \rho^i_\mathcal{A}(uk)\big) \in \Delta$. Since $\forall p \in P : t_{|u} \notin dom(T(A_p))$, it also holds that $\neg\exists p \in P : (p, (f, g')) \in \Delta$ for all $g' \in \Gamma_\perp$. Thus, by construction there exist transitions of the form $\big((q, P), (f, g), (\rho^i_{\mathcal{A}}(u1), P_1), \ldots, (\rho^i_{\mathcal{A}}(uk), P_k)\big) \in \Delta'$. Now, we have to pick a transition with suitable $P_1, \ldots, P_k$ such that for all $p_i \in P_i : t_{|ui} \notin dom(T(\mathcal{A}_{p_i})), 1 \leq i \leq k$.

We choose $P_1, \ldots, P_k$ as follows. Let r denote the transition applied at u in $\rho^i_{\mathcal{A}}$. To satisfy the first condition, for each transition r' with $r' \prec r$, we have to pursue a path of an alternative run on an alternative output for the given input, which is different from the considered output beginning at u. More formally, this corresponds to a tree $(t_{|u} \otimes t'_i) \cdot (t \otimes t_i[u/\circ])$ such that t'_i is compatible to r'. This tree is not one of the trees $t \otimes t_j$ with $(t, t_j) \in R$ for $1 \leq j \leq m$, because otherwise we would have chosen t_j instead of t_i from the possible output trees. Thus, there exists no successful run of $\mathcal{A}$ on $(t_{|u} \otimes t'_i) \cdot (t \otimes t_i[u/\circ])$. In particular, there is no successful run of $\mathcal{A}_q$ on $t_{|u} \otimes t'_i$ for all possible choices of t'_i. This means, there exists at least one path that does not lead to a final combination. Let $r' = (q, (f, g'), p_1, \ldots, p_n)$, if in this case the unsuccessful path is continued in the jth successor of u, add p_j to P_k.

To satisfy the second condition, we first remember that for each $p \in P$: $t_{|u} \notin dom(T(A_p))$. So, for each applicable rule $(p, (f, g'), p_1, \ldots, p_n \in \Delta$, there exists at least one $t_{|ui} \notin dom(T(\mathcal{A}_{p_i}))$ for $1 \leq i \leq n$. In case that $i \leq n$, we add p_i to P_i, otherwise we do not need to pursue the run any further.

By picking $P_1, \ldots, P_k$ as described above, the claim also holds for each node u at height $n + 1$. What is left to show is that the run $\rho_{\mathcal{B}}$ on $t \otimes t_i$ is indeed accepting, i.e., for each leaf u holds $(\rho_{\mathcal{B}}(u), \mathrm{val}_{t\otimes t_i}(u))$ is a final combination. Let $\mathrm{val}_{t\otimes t_i}(u) = (a, b)$ and $\rho_{\mathcal{B}}(u) = (q, P)$. Obviously, it follows directly from $(*)$ that there is no $p \in P$ such that $(p, (a, b') \in \Delta$ for all $b' \in \Gamma_\perp$. In addition, there is no $(q, (a, b')) \in \Delta$ with $(q, (a, b')) \prec (q, (a, b))$ for all $b' \in \Gamma_\perp$, because otherwise we would have chosen that output instead. Consequently, we have shown that $dom(R) \subseteq dom(R')$.

With the same argumentation, if we construct a run on $t \otimes t_j$ for another output t_j such that $i \neq j$ and $(t, t_j) \in R$, there exists at least one point in the run such that a smaller transitions belonging to the run $\rho^i_{\mathcal{A}}$ of $\mathcal{A}$ on $t \otimes t_i$ exists. Hence, at least a path of that run has to be pursued. No matter which path is chosen, it leads to a final combination in $\rho^i_{\mathcal{A}}$, because $(t, t_i) \in R$. So, $\mathcal{B}$ does not accept, because the run of $\mathcal{B}$ can either not be continued once a final combination in an alternative run occurs, or does not lead to a final combination by construction of Δ'.

From the above proof it is clear that for each $t \in dom(R)$ there exists exactly one t' such that $(t, t') \in R'$.

□

In the next chapters we consider the setting where the given relation is still tree-automatic, but we are looking for a uniformization in more restrictive classes.

Chapter 4

Uniformization by Top-Down Tree Transducers

In this chapter we investigate uniformization of tree-automatic relations in the class of top-down tree transformations. We restrict ourselves in the scope of this chapter to D↓TA-recognizable relations with D↓TA-recognizable domain. We previously asked whether a tree-automatic relation has a uniformization in the the same class, in this chapter we study the question: "Given a D↓TA-recognizable relation with D↓TA-recognizable domain, has the relation a uniformization in the class of top-down tree transformations?". However, we are mainly concerned with a variant of this uniformization problem. Thereby, we only require a transducer to realize a uniformization of a relation in the following way. 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. Our main result will be that the uniformization problem without input validation is decidable.

The chapter is structured as follows. We start by introducing the formal model of top-down tree transducers. Thereafter, we consider a special case of uniformization in the class of top-down tree transformations, namely the case that a uniformization of a D↓TA-recognizable relation with D↓TA-recognizable domain can be realized by a top-down transducer which simply re-labels the nodes of an input tree. Subsequently, we consider more general setups. Thereby we distinguish between uniformization with and without input validation.

4.1 Top-Down Tree Transducer

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 [CDG$^+$07].

Definition 4.1 (TDT). A *top-down tree transducer* 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, a set $Q_0 \subseteq Q$ of initial states, and Δ is a finite set of transition rules of the form

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

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

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

where u is an n-context over Γ and $q, q_1, \dots, q_n \in Q$.

For our purposes it is more convenient to use a non-standard formalization of configurations, as this will simplify the reading of the proofs.

Definition 4.2 (Configuration). 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 $\varphi : D_{t'} \to \mathrm{dom}_t$, where

- $\mathrm{val}_{t'}(u) \in \Gamma_i$ for each $u \in \mathrm{dom}_{t'}$ with $i > 0$ successors
- $\mathrm{val}_{t'}(u) \in \Gamma_0$ or $\mathrm{val}_{t'}(u) \in Q$ for each leaf $u \in \mathrm{dom}_{t'}$
- $D_{t'} \subseteq \mathrm{dom}_{t'}$ with $D_{t'} = \{u \in \mathrm{dom}_{t'} \mid \mathrm{val}_{t'}(u) \in Q\}$

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 $\to_\mathcal{T}$ on configurations by:

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

if a non-ε-transition was applied, or

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

if an ε-transition was applied.
Furthermore, let $\to_\mathcal{T}^*$ be the reflexive and transitive closure of $\to_\mathcal{T}$ and $\to_\mathcal{T}^n$ the reachability relation for $\to_\mathcal{T}$ in n steps.

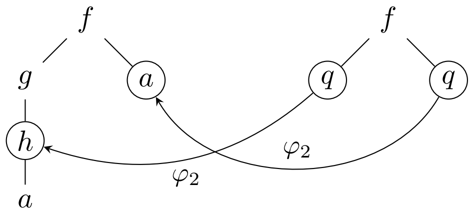


Figure 4.1: The configuration $c_2 = (t, f(q, q), \varphi_2)$ of $\mathcal{T}$ on t from Example 4.4.

A top-down tree transducer is *deterministic* (a DTDT) if it contains no ε-transition and there are no two rules with the same left-hand side. Additionally, the set of initial states is a singleton set.

Definition 4.3 (Semantics of TDTs). The relation $R(\mathcal{T}) \subseteq T_\Sigma \times T_\Gamma$ induced by a top-down tree transducer $\mathcal{T}$ is

$$R(\mathcal{T}) = \{(t, t') \mid (t, q_0, \varphi) \to^*_{\mathcal{T}} (t, t', \varphi') \text{ with } q_0 \in Q_0 \text{ and } \varphi(\varepsilon) = \varepsilon\}.$$

For a tree $t \in T_\Sigma$ let $\mathcal{T}(t) := \{t' \in T_\Gamma \mid (t, t') \in R(\mathcal{T})\}$.

The class of relations definable by TDTs is called the class of *top-down tree transformations*.

Example 4.4 Let Σ be a ranked alphabet given by $\Sigma_2 = \{f\}$, $\Sigma_1 = \{g, h\}$, and $\Sigma_0 = \{a\}$. Consider the TDT $\mathcal{T}$ given by $(\{q\}, \Sigma, \Sigma, \{q\}, \Delta)$ with $\Delta =$

$$\begin{array}{lrcll} \{ & q(a) & \to & a, & \\ & q(g(x_1)) & \to & q(x_1), & \\ & q(h(x_1)) & \to & h(q(x_1)), & \\ & q(f(x_1, x_2)) & \to & f(q(x_1), q(x_2)) & \}. \end{array}$$

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 $c_0 \to^5_{\mathcal{T}} c_5$ 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 4.1.

4.2 A Restricted Uniformization Case

In this section we deal with a special case of uniformization. We only consider D↓TA-recognizable relations with D↓TA-recognizable domain such that for each pair (t, t') in the relation holds that t and t' have the same domain. In this context, we are interested whether there exists a uniformization by a TDT such that the transducer passes through each node and thereby relabels the node according to the current state and input symbol at that node. This is captured in the following definition.

Definition 4.5 Let R be a D↓TA-recognizable relation with D↓TA-recognizable domain such that for each $(t, t') \in R$ holds $\mathrm{dom}_t = \mathrm{dom}_{t'}$. The *restricted uniformization problem* is the decision problem whether there exists a uniformization of R whose graph is recognizable by a deterministic TDT $\mathcal{T}$ such that only transitions of the following form are used:

$$q(f(x_1, \dots, x_i)) \to g(q_1(x_1), \dots, q_i(x_i)),$$

where $f \in \Sigma_i$, $g \in \Gamma_i$, and $q, q_1, \dots, q_i \in Q$.

For the remainder of this section, let $R \subseteq T_\Sigma \times T_\Gamma$ be a relation recognized by a D↓TA $\mathcal{A} = (Q_\mathcal{A}, \Sigma \times \Gamma, q_0^\mathcal{A}, \Delta_\mathcal{A})$ and let $dom(R)$ be recognized by a D↓TA $\mathcal{B} = (Q_\mathcal{B}, \Sigma, q_0^\mathcal{B}, \Delta_\mathcal{B})$ such that $\mathrm{dom}_t = \mathrm{dom}_{t'}$ holds for each $(t, t') \in R$.

We now describe a decision procedure for this problem. We consider a game 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 and Out can react with one output symbol.

The vertices in the game graph represented the current state of $\mathcal{B}$ on the input and the current state of $\mathcal{A}$ on the input combined with the output. A move of In corresponds to a direction and an input symbol, a move of Out corresponds to the chosen output.

Formally, the game graph $G_{\mathcal{A},\mathcal{B}}$ is of the form $(V_{\mathsf{In}}, V_{\mathsf{Out}}, E, v_0)$, where

- $V_{\mathsf{In}} \subseteq 2^{Q_\mathcal{A} \times Q_\mathcal{B}}$ is the set of vertices of player In,
- $V_{\mathsf{Out}} \subseteq V_{\mathsf{In}} \times \Sigma$ is the set of vertices of player Out.
- From a vertex of In the following moves are possible:
  - $P \to ((q, p), f)$ for each $(q, p) \in P$ and there exists $(p, f, p_1, \dots, p_i) \in \Delta_\mathcal{B}$ and $(q, (f, g), q_1, \dots, q_i) \in \Delta_\mathcal{A}$ with $g \in \Gamma$
- From a vertex of Out the following moves are possible:
  - $((q, p), f) \xrightarrow{r} P$ with $P = \bigcup_{j=1}^{i} \{(q_j, p_j)\}$ if there exists a transition $r = (q, (f, g), q_1, \dots, q_i) \in \Delta_\mathcal{A}$ and a transition $(p, f, p_1, \dots, p_i) \in \Delta_\mathcal{B}$
- The initial vertex v_0 is $\{(q_0^\mathcal{A}, q_0^\mathcal{B})\}$.

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 all $P \in V_{\mathsf{In}}$ such that there is $(q, p) \in P$ and $f \in \Sigma$ such that $(p, f, p_1, \dots, p_i) \in \Delta_\mathcal{B}$, but there exists no $(q, (f, g), q_1, \dots, q_i) \in \Delta_\mathcal{A}$ for some $g \in \Gamma$. If one of these vertices is reached during a play, Out loses the game. Let B denote the set of these bad vertices which Out should avoid. Hence, we specify the game $\mathcal{G}_{\mathcal{A},\mathcal{B}} = (G_{\mathcal{A},\mathcal{B}}, V \setminus B)$ as a safety game for Out.

The following example demonstrates the presented construction for a simple relation.

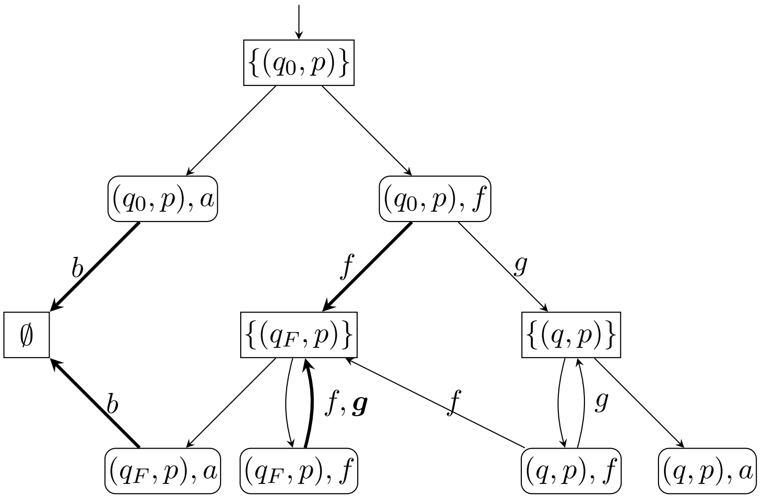


Figure 4.2: The game graph $G_{\mathcal{A},\mathcal{B}}$ constructed from the D↓TAs $\mathcal{A}$ and $\mathcal{B}$ from Example 4.6. A possible winning strategy for Out in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$ is emphasized in the graph.

Example 4.6 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 defined by

$$R := \{(t, t') \in T_\Sigma \times T_\Gamma \mid (\mathrm{dom}_t = \mathrm{dom}_{t'}) \wedge \forall u \in \mathrm{dom}_t \text{ with } \mathrm{val}_t(u) = a \wedge |u| > 0 : \exists v \sqsubset u \text{ s.t.} \mathrm{val}_{t'}(u) = f\}.$$

The relation contains exactly the pairs of trees (t, t') of same size such that on every path through t' occurs an f if $h(t') > 0$.

Obviously, the relation and the domain are D↓TA-recognizable. It is easy to see that the D↓TA $\mathcal{B} = (\{p\}, \Sigma, p, \Delta_\mathcal{B})$ with $\Delta_\mathcal{B} = \{(p, a), (p, f, p, p)\}$ recognizes *dom*(R) and the 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.

The corresponding game graph $G_{\mathcal{A},\mathcal{B}}$ is depicted in Figure 4.2.

We show that the game is constructed such that a TDT uniformizing R can be built up from a winning strategy of Out and vice versa.

Lemma 4.7 *The relation R can be uniformized by a restricted TDT in the sense as described above if, and only if,* Out *has a winning strategy in the safety game $\mathcal{G}_{\mathcal{A},\mathcal{B}} = (G_{\mathcal{A},\mathcal{B}}, V \setminus B)$.*

Proof. Assume Out has a winning strategy in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$, then Out has also a positional one. A positional winning strategy can be represented by a function $\sigma : V_{\mathsf{Out}} \to \Delta_\mathcal{A}$. However, it is more convenient to write $\sigma : v \overset{r}{\mapsto} w$ instead of $\sigma(v) = r$ for $(v, w) \in E$ and $r \in \Delta_\mathcal{A}$ as the former notation directly indicates the subsequently reached vertex of In. We construct a deterministic TDT $\mathcal{T} = (Q, \Sigma, \Gamma, q_0, \Delta)$ from σ as follows:

- $Q := V_{\mathsf{In}}$ is the set of states, with $q_0 := (q_0^{\mathcal{A}}, q_0^{\mathcal{B}})$ as initial state, and
- Δ is the transition relation build as follows:
  - For each $\sigma : ((q,p), f) \overset{r}{\mapsto} P$ with $r = (q, (f, g), q_1, \ldots, q_i) \in \Delta_{\mathcal{A}}$ and $(p, f, p_1, \ldots, p_i) \in \Delta_{\mathcal{B}}$ add $(q,p)(f(x_1, \ldots, x_i)) \rightarrow g((q_1, p_1)(x_1), \ldots, (q_j, p_j)(x_j))$ to Δ, and
  - for each $\sigma : ((q,p), a) \overset{r}{\mapsto} \emptyset$ with $r = (q, (a, b)) \in \Delta_{\mathcal{A}}$ and $(p, a) \in \Delta_{\mathcal{B}}$ add $(q,p)(a) \rightarrow b$ to Δ.

We now verify that $\mathcal{T}$ defines a uniformization of R. The transitions of $\mathcal{T}$ correspond to transitions of the product $\mathcal{A} \times \mathcal{B}$, so it is easy to see, that $R(\mathcal{T}) \subseteq R(\mathcal{A})$. We prove the other direction $R(\mathcal{A}) \subseteq R(\mathcal{T})$. Let $t \in dom(R)$ and ρ is the run of $\mathcal{B}$ on t. We show by induction on the height of a node $u \in \mathrm{dom}_t$ that there exists a configuration (t, t', φ) of $\mathcal{T}$ such that for each $u \in \mathrm{dom}_t$ that is a leaf in t' holds

$$\text{if } \rho(u) = p \in Q_{\mathcal{B}}, \text{ then } \varphi(u) = u \text{ and } \mathrm{val}_{t'}(u) = (q,p) \text{ for some } q \in Q_{\mathcal{A}}. \quad (*)$$

Clearly, from the structure of the transition rules of $\mathcal{T}$ we see, that for each configuration (t, t', φ) of $\mathcal{T}$ the function φ is the identity function. For the induction base, let $(t, (q_0^{\mathcal{A}}, q_0^{\mathcal{B}}), \varphi)$ be the initial configuration. We have $\rho(\varepsilon) = q_0^{\mathcal{B}}$ and $\mathrm{val}_{t'}(\varepsilon) = (q_0^{\mathcal{A}}, q_0^{\mathcal{B}})$, thus the claim holds.

For the induction step, assume the claim holds for n. We consider a node u at height n. Let $\rho(u) = p$, by induction hypothesis there exists a configuration $c = (t, t', \varphi)$ such that $\mathrm{val}_{t'}(u) = (q,p)$. From $(q,p) \in Q$ it follows from the construction of $\mathcal{T}$ that it is possible to reach a vertex P with $(q,p) \in P$ in the game graph if Out plays according to her winning strategy σ. Let $\mathrm{val}_t(u) = f \in \Sigma_i$, then there exists a transition $(p, f, p_1 \ldots, p_i) \in \Delta_{\mathcal{B}}$. Since σ is a winning strategy and P is not a bad vertex, a vertex $(q,p), f$ can be reached in a game played according to σ. Hence, there exists a transition $(q,p)(f(x_1, \ldots, x_i)) \rightarrow g((q_1, p_1)(x_1), \ldots, (q_i, p_i)(x_i)) \in \Delta$ for $\sigma(((q,p), f)) \overset{r}{\mapsto} P$ with $r = (q, (f, g), q_1, \ldots, q_n) \in \Delta_{\mathcal{A}}$ by construction. Let $c' = (t, t'', \varphi')$ denote the successor configuration of c that results from applying the above transition. Therefore, for the jth successor of u at height $n + 1$, it holds that $\rho(uj) = p_j$ and $val_{t''}(uj) = (q_j, p_j)$ and thus, the claim holds.

Now, for $t \in dom(R)$, let t' be an output tree such that (t, t', φ) is a configuration of $\mathcal{T}$ such that $(*)$ holds at each leaf of t. With the same reasoning as above in the induction step, there exists for each leaf u with $\mathrm{val}_t(u) = a$ a transition $(q,p)(a) \rightarrow b \in \Delta$. When we apply the respective transitions at the leaves, we obtain a tree t'' with $(t, t'') \in R(\mathcal{T})$. Hence, $\mathcal{T}$ defines a uniformization of R.

For the other direction, we assume that R is uniformized by some DTDT $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta)$ such that Δ only contains transitions with the restrictions introduced in Definition 4.5. The idea is that a winning strategy for Out reproduces the transformations of $\mathcal{T}$. The moves of In induce a path $x_1 \ldots x_n$ of vertices together with an input sequence $v_1 \dots v_n$ on that path through some input tree t. Only this part is needed to compute the output sequence $u_1 \dots u_n$ of the same length on the same path that $\mathcal{T}$ produces for the given input sequence. Let $\rho_\mathcal{A}$ be the unambiguous run of $\mathcal{A}$ on $t \otimes T(t)$ and $\rho_\mathcal{B}$ be the unambiguous run of $\mathcal{B}$ on t. Thus, the corresponding vertex in the game graph is $((q,p),v_n)$ with $q = \rho_\mathcal{A}(x_n)$ and $p = \rho_\mathcal{B}(x_n)$. Since $\mathcal{T}$ uniformizes R, there have to exist transitions $r = (q,(v_n,u_n),q_1,\dots,q_i) \in \Delta_\mathcal{A}$ and $(p,v_n,p_1,\dots,p_i) \in \Delta_\mathcal{B}$. Then, we define σ such that Out moves to $\bigcup_{j=1}^{i}(q_j,p_j)$ via the r-labeled edge that has to exist by construction of the game graph.

To show that this defines a winning strategy for Out we show that the next reached vertex of Out is not a bad vertex. The next move of In defines a new path segment x_{n+1} and an input symbol v_{n+1} leading to $((q_j,p_j),v_{n+1})$ for a $j \in \{1,\dots,i\}$. Since $\mathcal{T}$ uniformizes R, we know that there exists a tree $t \in dom(R)$ and output $\mathcal{T}(t)$ such that the path $x_1 \dots x_{n+1}$ through $t \otimes \mathcal{T}(t)$ is labeled with $(v_1,u_1)\dots(v_{n+1},u_{n+1})$ and $(t,\mathcal{T}(t)) \in R$. Since $\mathcal{A}$ recognizes R, we know that there exists a transition $\big(q_j,(v_{n+1},u_{n+1}),q_{j1},\dots,q_{ji}\big) \in \Delta_\mathcal{A}$. Hence, the vertex $((q_j,p_j),v_{n+1})$ is not in B.

□

In the following example we present a TDT that uniformizes the relation R from Example 4.6, obtained from the winning strategy for Out in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$, shown in Figure 4.2.

Example 4.8 The TDT $\mathcal{T}$ with initial state (q_0,p) that is obtained from the winning strategy of Out in Figure 4.2, by using the construction from Lemma 4.7, has the following transition rules:

$$\begin{array}{rcl} (q_0,p)(a) & \to & b, \\ (q_0,p)\big(f(x_1,x_2)\big) & \to & f\big((q_F,p)(x_1),(q_F,p)(x_2)\big), \\ (q_F,p)(a) & \to & b, \\ (q_F,p)\big(f(x_1,x_2)\big) & \to & g\big((q_F,p)(x_1),(q_F,p)(x_2)\big). \end{array}$$

Obviously, $\mathcal{T}$ uniformizes R from Example 4.6.

Together with Lemma 4.7 and the fact that a winning strategy for Out can be effectively computed in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$, we obtain the following result.

Theorem 3 *The restricted uniformization problem in the class of top-down tree transformations is decidable.*

4.3 Notations and Definitions

In this section, we fix some notations and definitions that will simplify the presentation of the proofs in the following sections.

Given $\Sigma = \bigcup_{i=0}^{m} \Sigma_i$, let $\mathrm{dir}_\Sigma = \{1,\dots,m\}$ be the set of directions compatible with Σ. Usually, we write dir when Σ is clear from the context.

For $\Sigma = \bigcup_{i=0}^{m} \Sigma_i$, the set Path_Σ of labeled path over Σ is defined inductively by:

- ε is a labeled input path
- 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 $path(\pi) = \varepsilon$, $lbls(\pi) = \pi$
- 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

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

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 $u \in \mathbb{N}^*$, let $||t||^u := max\{v \mid u \sqsubseteq v \vee v \sqsubseteq u\}$ be the length of a maximal path through t along u. In particular, for $u = \varepsilon$ it holds that $||t|| = ht(t)$.

For $R \subseteq T_\Sigma \times T_\Gamma$ and $\pi \in \mathrm{Path}_\Sigma$ let

$$R^\pi := \{(t, t') \in R \mid t \in T_\Sigma^\pi\}.$$

For a D↓TA $\mathcal{A}$ with state set Q and $q \in Q$, let $\mathcal{A}_q$ be the automaton that results from $\mathcal{A}$ by using q as single initial state.

Let $R \subseteq T_\Sigma \times T_\Gamma$ be recognized by a D↓TA $\mathcal{A}$, for $q \in Q$ we define

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

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

Since we will consider labeled paths through trees, it is convenient to define the notion of convolution for labeled paths. For a labeled path $x \in \mathrm{Path}_\Sigma$ with $||x|| > 0$, let $\mathrm{dom}_x := \{u \in \mathrm{dir}^* \mid u \sqsubseteq path(x)\}$ and $\mathrm{val}_x : \mathrm{dom}_x \to \Sigma$, where $\mathrm{val}_x(u) = lbls(x)[i]$ if $u \in \mathrm{dom}_x$ with $|u| = i + 1$.

Let Σ, Γ be ranked alphabets and 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

$\mathrm{dom}_{x\otimes y} = \mathrm{dom}_x \cup \mathrm{dom}_y$, and $\mathrm{val}_{x\otimes y}(u) = (\mathrm{val}_x^{\perp}(u), \mathrm{val}_y^{\perp}(u))$ for all $u \in \mathrm{dom}_{x\otimes y}$, where $\mathrm{val}_x^{\perp}(u) = \mathrm{val}_x(u)$ if $u \in \mathrm{dom}_x$ and $\mathrm{val}_x^{\perp}(u) = \perp$ otherwise, analogously defined for $\mathrm{val}_y^{\perp}(u)$.

Furthermore, it is useful to relax the notion of runs to labeled paths. Let $\mathcal{B} = (Q_{\mathcal{B}}, \Sigma, q_0^{\mathcal{B}}, \Delta_{\mathcal{B}})$ be a D↓TA that recognizes a tree language over an alphabet Σ and $\mathcal{A} = (Q_{\mathcal{A}}, \Sigma_{\perp} \times \Gamma_{\perp}, q_0^{\mathcal{A}}, \Delta_{\mathcal{A}})$ be a D↓TA that recognizes a tree relation over an input alphabet Σ and an output alphabet Γ. For $\mathcal{B}$ and $x \in \mathrm{Path}_{\Sigma}$, let $\rho_{\mathcal{B}} : \mathrm{dir}^* \to Q_{\mathcal{B}}$ be the partial function with the property that

- $\rho_{\mathcal{B}}(\varepsilon) = q_0^{\mathcal{B}}$, and
- for each $u \in \mathrm{dom}_x$: if $p := \rho_{\mathcal{B}}(u)$ is defined and there is a transition $(p, \mathrm{val}_x(u), p_1, \ldots, p_i) \in \Delta_{\mathcal{B}}$, then $\rho_{\mathcal{B}}(u.j) = p_j$ for all $j \in \{1, \ldots, i\}$.

Similarly for $\mathcal{A}$ and $x \in \mathrm{Path}_{\Sigma}$, $y \in \mathrm{Path}_{\Gamma}$ such that $x \otimes y$ is defined. Let $i \in \mathrm{dir}$, $x \in \mathrm{Path}_{\Sigma}$, $y \in \mathrm{Path}_{\Gamma}$ such that $x \otimes y$ is defined with $path(x) = u$ and $path(x \otimes y) = v$. We write

$$\mathcal{B} : q_0^{\mathcal{B}} \xrightarrow{x}_i p,$$

if $p := \rho_{\mathcal{B}}(ui)$ is defined. Analogously, we write

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

if $q := \rho_{\mathcal{A}}(vi)$ is defined. We write $\mathcal{B} : q_0^{\mathcal{B}} \xrightarrow{x} F_{\mathcal{B}}$ resp. $\mathcal{A} : q_0^{\mathcal{A}} \xrightarrow{x\otimes y} F_{\mathcal{A}}$ if $(\rho_{\mathcal{B}}(u), \mathrm{val}_x(u)) \in \Delta_{\mathcal{B}}$ resp. $(\rho_{\mathcal{A}}(v), \mathrm{val}_{x\otimes y}(v)) \in \Delta_{\mathcal{A}}$ to indicate that the (partial) run on the respective labeled path is accepting.

Let $R \subseteq T_{\Sigma} \times T_{\Gamma}$ be the relation recognized by a DTDT $\mathcal{T}$. Sometimes, we are interested in the output that $\mathcal{T}$ produces for an arbitrary tree, which might e.g., be part of a valid input tree. Therefore, for a tree t, we redefine $\mathcal{T}(t)$. Given an arbitrary tree $t \in T_{\Sigma}$ or $t \in S_{\Sigma}$, then let

$$\mathcal{T}(t) := \{t' \mid (t, q_0^{\mathcal{T}}, \varphi_0) \to_{\mathcal{T}} (t, t', \varphi) \wedge \neg\exists (t, t'', \varphi') \text{ such that } (t, t', \varphi) \to_{\mathcal{T}} (t, t'', \varphi')\}$$

be the final transformed output of $\mathcal{T}$ for the input tree t. If $t \in dom(R)$, then $\mathcal{T}(t)$ is t' with $(t, t') \in R$ as previously defined in the introduction to TDTs.

Also, 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 $u \in \mathrm{dir}^*$, we define

$$out_{\mathcal{T}}(t, u) := \{\pi \in \mathrm{Path}_{\Gamma} \mid \mathcal{T}(t) \in T_{\Gamma}^{\pi} \wedge (path(\pi) \sqsubseteq u \vee u \sqsubseteq path(\pi))\}$$

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

4.4 Uniformization Without Input Validation

As mentioned in the introduction to this chapter, here we consider the question whether a relation has a uniformization by a top-down tree transducer such that the transducer may behave arbitrarily on trees that are not in the domain of the considered relation.

Definition 4.9 Let $R \subseteq T_\Sigma \times T_\Gamma$ denote a D↓TA-recognizable relation with D↓TA-recognizable domain. The *uniformization problem without input validation* is the decision problem whether there exists a TDT $\mathcal{T}$ such that $R(\mathcal{T}) \cap R$ defines a uniformization of R.

The goal of this section is to provide a decision procedure for the aforementioned question. Again, we will reduce this question to the existence of winning strategies in a safety game. To begin with, we investigate the connection between input and (delayed) output under the assumption that the input tree is valid. Afterwards, we consider relations that have a uniformization by transducers with bounded output delay and then extend our results to transducers with unbounded output delay.

While in the previous section the considered relations as well as the considered transducers were very restricted, we now consider a more general setup. That means, a transducer possibly reaches configurations where read input symbol and produced output symbol are not only of different arity, but might also occur on different nodes. In these cases, as the transducer can not read the whole input, because of either the different arity or the introduced delay, it is necessary that fixed output trees exist matching all possible unread input variants.

Later, we will frequently use the following result that shows that it is decidable for a given regular set of input trees whether there exits a single fixed matching output tree.

Lemma 4.10 *Let $R \subseteq T_\Sigma \times T_\Gamma$ denote a D↓TA-recognizable relation with D↓TA-recognizable domain. 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, and let $\mathcal{B} = (Q_\mathcal{B}, \Sigma, q_0^\mathcal{B}, \Delta_\mathcal{B})$ be a D↓TA that recognizes dom(R) and let $q \in Q_\mathcal{A}, p \in Q_\mathcal{B}$. It is decidable whether the following holds:*

(a) $\forall t \in T(\mathcal{B}_p) : t \otimes \perp \in T(\mathcal{A}_q)$,

(b) $\exists t' \in T_\Gamma : \perp \otimes t' \in T(\mathcal{A}_q)$,

(c) $\exists t' \in T_\Gamma \forall t \in T(\mathcal{B}) : (t, t') \in R$.

Proof.

Part (a). Let $\mathcal{A}_q^{\Sigma\times\perp}|_{T_\Sigma} = (Q_\mathcal{A}, \Sigma, q, \Delta'_\mathcal{A})$ be the automaton that results from $\mathcal{A}_q$ by using $\big(\bigcup_{i=1}^m (Q_\mathcal{A} \times (\Sigma \times \perp) \times Q_\mathcal{A}^i)\big) \cap \Delta_\mathcal{A}$ projected onto $\bigcup_{i=1}^m (Q_\mathcal{A} \times \Sigma \times Q_\mathcal{A}^i)$ as new transition set $\Delta'_\mathcal{A}$. It holds that $\forall t \in T(\mathcal{B}_p) : t \otimes \perp \in T(\mathcal{A}_q)$ if, and only if, $T(\mathcal{B}_p) \subseteq T(\mathcal{A}_q^{\Sigma\times\perp}|_{T_\Sigma})$. Note, if (q, p) is a reachable state in the product automaton $\mathcal{A} \times \mathcal{B}$, then it always holds that $dom(T(\mathcal{A}_q)) \subseteq T(\mathcal{B}_p)$. In this case, we can test whether $T(\mathcal{B}_p) = T(\mathcal{A}_q^{\Sigma\times\perp}|_{T_\Sigma})$ holds.

Part (b). Let $\mathcal{A}_q^{\perp\times\Gamma}|_{T_\Gamma} = (Q_\mathcal{A}, \Gamma, q, \Delta'_\mathcal{A})$ be the automaton that results from $\mathcal{A}_q$ by using $\big(\bigcup_{i=1}^m (Q_\mathcal{A} \times (\perp \times \Gamma) \times Q_\mathcal{A}^i)\big) \cap \Delta_\mathcal{A}$ projected onto $\bigcup_{i=1}^m (Q_\mathcal{A} \times \Gamma \times Q_\mathcal{A}^i)$ as new transition set $\Delta'_\mathcal{A}$. It holds that $\exists t' \in T_\Gamma : \perp \otimes t' \in T(\mathcal{A}_q)$ if, and only if, $T(\mathcal{A}_q^{\perp\times\Gamma}|_{T_\Sigma}) \neq \emptyset$.

Part (c). First, we construct an N↓TA $\bar{\mathcal{A}}$ for $\overline{R}$. Secondly, we define the N↓TA $\mathcal{C} = (Q_{\bar{\mathcal{A}}} \times (Q_{\mathcal{B}} \cup p_\bot), \Sigma_\bot \times \Gamma_\bot, (q_0^{\mathcal{B}}, q_0^{\bar{\mathcal{A}}}), \Delta_{\mathcal{C}})$ as the product automaton of $\bar{\mathcal{A}} \times \mathcal{B}$ with $\Delta_{\mathcal{C}}$ constructed as follows:

- For $p \in Q_{\mathcal{B}}$, $q \in Q_{\bar{\mathcal{A}}}$, and $f \in \Sigma$ such that $(p, f, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$ and $(q, (f, g), q_1, \dots, q_n) \in \Delta_{\bar{\mathcal{A}}}$

$$\big((q, p), (f, g), (q_1, p_1), \dots, (q_i, p_i), (q_{i+1}, p_\bot), \dots, (q_n, p_\bot)\big) \in \Delta_{\mathcal{C}},$$

- for $p_\bot$, $q \in Q_{\bar{\mathcal{A}}}$ such that $(q, (\bot, g), q_1, \dots, q_n) \in \Delta_{\mathcal{A}}$

$$\big((q, p_\bot), (\bot, g), (q_1, p_\bot), \dots, (q_n, p_\bot)\big) \in \Delta_{\mathcal{C}}.$$

We obtain $R(\mathcal{C}) = \{(t, t') \in T_\Sigma \times T_\Gamma \mid t \in T(\mathcal{B}) \wedge (t, t') \notin R\}$. Let $\mathcal{C}|_{T_\Gamma}$ be the automaton that results from $\mathcal{C}$ by projection of $\Delta_{\mathcal{C}}$ onto $\bigcup_{i=1}^{m}(Q_{\mathcal{C}} \times \Gamma \times Q_{\mathcal{C}}^i)$. $T(\mathcal{C}|_{T_\Gamma}) = \{t' \in T_\Gamma \mid \exists t \in T(\mathcal{B}) : (t, t') \notin R\}$. Thirdly, construct a tree automaton $\overline{\mathcal{C}|_{T_\Gamma}}$ that recognizes the complement $\overline{T(\mathcal{C}|_{T_\Gamma})} = \{t' \in T_\Gamma \mid \forall t \in T(\mathcal{B}) : (t, t') \in R\}$. It holds that $\exists t' \in T_\Gamma \, \forall t \in T(\mathcal{B}) : (t, t') \in R$ if, and only if, $T(\overline{\mathcal{C}|_{T_\Gamma}}) \neq \emptyset$.

□

4.4.1 Bounded Output Delay

Let $\mathcal{T}$ be an arbitrary TDT and let $c = (t, t', \varphi)$ be a configuration of $\mathcal{T}$. Consider a node $u \in D_{t'}$ with $\varphi(u) = v$. For arbitrarily formed transitions, it can occur that $|u| \neq |v|$. If $|u| < |v|$, then the transducer has read an input sequence and produced a shorter output sequence. In this case, we say the transducer has an *output delay* of $|v| - |u|$. If no configuration is reachable such that the delay is greater than some $n \in \mathbb{N}$, then the output delay is bounded to n. In addition, if configurations such that $|u| > |v|$ occur, then the transducer has produced a longer output sequence than the read input sequence.

The following lemma shows that we can focus on the construction of TDTs where the read input and produced output lie on the same path if only valid input trees are considered. The basic idea behind the proof is that D↓TAs can not compare information on divergent paths.

Lemma 4.11 *Let R be a D↓TA-definable relation. If R is uniformized by a TDT $\mathcal{T}$ without input validation in which the output delay is bounded, then R can be uniformized by a TDT $\mathcal{T}'$ without input validation in which the output delay is bounded such that for each $t \in dom(R)$ and each reachable configuration (t, t', φ) holds:*

$$\forall u \in D_{t'} \text{ with } \varphi(u) = v : u \sqsubseteq v.$$

Proof. Let $\mathcal{A}$ be a D↓TA that recognizes R. The proof is split in two parts. First, we show that we can keep track of the current state of $\mathcal{A}$ on the combined part of input and output in the state set of $\mathcal{T}$ as long as input and output are on the same path. Secondly, we show that if we can not keep track of the current state of $\mathcal{A}$, because the path of input and output diverge, then it is also not necessary to keep track. Instead a fixed output tree can be produced.

We assume that in $\mathcal{T}$ the output delay is bounded to K. Furthermore, we assume that the output never overtakes the input, i.e., no configuration with $|u| > |v|$ is reached. Otherwise, if the length of the output that can become ahead of the input is bounded in $\mathcal{T}$, then the output can always be postponed until a leaf is reached as described in Remark 4.12. If the length of the output that can become ahead of the input is unbounded, then it becomes possible to drop postponed output such that input and output are one same level again. However, we will not present this case, because we will later see that it is not necessary to consider TDTs in which the output overtakes the input.

As long as $u \sqsubseteq v$ holds, it is possible to keep track in $\mathcal{T}$ in which state $\mathcal{A}$ is at the vertex u. For that, we define a cross-product of $\mathcal{T}$ and $\mathcal{A}$ inductively. Since $\mathcal{T}$ can introduce output delay, the states of the cross-product also have to keep track of the part of the input currently ahead. Starting in $\mathcal{T}$ from s_0 and in $\mathcal{A}$ from q_0 each rule

$$s_0(f(x_1,\dots,x_i)) \to g(w_1,\dots,w_j),$$

where $w_k \in T_\Gamma$ or $w_k = s_k(x_k), s_k \in Q_\mathcal{T}$ for all $1 \leq k \leq j$ is replaced by

$$(s_0,q_0)(f(x_1,\dots,x_i)) \to g(w'_1,\dots,w'_j),$$

where $w'_k = w_k$ if $w_k \in T_\Gamma$, or $w'_k = (s_k,q_k)(x_k)$ if $w_k = s_k(x_k)$ and $(q_0,(f,g),q_1,\dots,q_n) \in \Delta_\mathcal{A}$, and each rule

$$s_0(f(x_1,\dots,x_i)) \to s'(x_k)$$

is replaced by

$$(s_0,q_0)(f(x_1,\dots,x_i)) \to (s',q_{0_f})(x_k).$$

Generally, for some state $(s,q_{f_1 j_1 \dots f_m})$ each rule

$$\begin{aligned} s(f(x_1,\dots,x_i)) \to& \\ &g_1(t_{11},\dots,t_{1(j_1-1)},\circ,t_{1(j_1+1)},\dots,t_{1n_1}) \cdot \ldots \cdot \\ &\quad g_k(t_{k1},\dots,t_{k(j_k-1)},s'(x_{j_m}),t_{k(j_k+1)},\dots,t_{kn_k}), \end{aligned}$$

where $t_{uv} \in T_\Gamma$ for all u,v such that $1 \leq u \leq k, 1 \leq v \leq n_u$ is replaced by

$$\begin{aligned} (s,q_{f_1 j_1 \dots f_m})(f(x_1,\dots,x_i)) \to& \\ &g_1(t_{11},\dots,t_{1(j_1-1)},\circ,t_{1(j_1+1)},\dots,t_{1n_1}) \cdot \ldots \cdot \\ &\quad g_k(t_{k1},\dots,t_{k(j_k-1)},(s',q'_{f_{k+1} j_{k+1} \dots f_m j_m f_n})(x_{j_m}),t_{k(j_k+1)},\dots,t_{kn_k}), \end{aligned}$$

where $(q^{i-1}_{j_{i-1}},(f_i,g_i),q^i_1,\dots,q^i_{j_i},\dots,q^i_{n_i}) \in \Delta_\mathcal{A}$ for all $1 \leq i \leq k$ such that $q^0_{j_0} = q$, $q^k_{j_k} = q'$ and $f_n = f$.

Note, if $k = m+1$, then the read input and produced output are on the same tree level afterwards. This means, that any successor of g_k can be of the form $s''(x_l)$ for some $s'' \in Q_\mathcal{T}$ if it is the lth successor instead of a fixed tree t_{kl}. Furthermore, each rule

$$s(f(x_1,\dots,x_i)) \to s'(x_k)$$

is replaced by

$$(s, q_{f_1 j_1 \dots f_m})(f(x_1,\dots,x_i)) \to (s', q_{f_1 j_1 \dots f_m j_m f_n})(x_k),$$

where $j_m = k$ and $f_n = f$.

Obviously, if the right-hand side of a transition is a fixed tree, only the left-hand side has to be changed.

Now we consider the case that at some point $u \sqsubseteq v$ does not hold. We show that we can construct a transducer such that $u \sqsubseteq v$ always holds by defining the cross-product $\mathcal{T} \times \mathcal{A}$ in the following way if it cannot be constructed as above.

Let $\mathcal{T}$ uniformize R such that the property is not fulfilled. Then there exists $t \in dom(R)$ such that $(t, s_0, \varphi) \to_{\mathcal{T}}^{*} (t, t_1, \varphi_1) \to_{\mathcal{T}} (t, t_2, \varphi_2)$ and for (t, t_2, φ_2) the property does not hold for the first time. Thus, there exists $u \in \mathrm{dom}_{t_1}$ with $\varphi_1(u) = v$ and $u \sqsubseteq v$, and in the next step there exists some $u' \in \mathrm{dom}_{t_2}$ with $\varphi_2(u') = v'$ and $u \sqsubset u'$ such that $u' \not\sqsubseteq v'$.

Let the successor configuration (t, t_2, φ_2) of (t, t_1, φ_1) be the result of applying the following transition

$$s(f(x_1,\dots,x_n)) \to w[\dots, s_i(x_{j_i}), \dots],$$

where $f \in \Sigma_i$, w is a context over Γ, $s, s_i \in Q_{\mathcal{T}}$ and $j_i \in \{1,\dots,n\}$ such that $s_i(x_{j_i})$ causes the problem, i.e., $\varphi_2(u') = v'$ with $v' = v.j_i$ and $\mathrm{val}_{t_2}(u') = s_i$.

Let $t' := \mathcal{T}(t)$ be the final output of $\mathcal{T}$ for the given input t with $(t, t') \in R$. Since $u \sqsubseteq v$, we can compute the output of $\mathcal{T} \times \mathcal{A}$ for t with the already given rules up to u. This means there is a configuration (t, t_1', φ_1') of $\mathcal{T} \times \mathcal{A}$ reachable that only differs from (t, t_1, φ') in the states at the leaves in t_1 which also contain the information of $\mathcal{A}$ and the part of the input ahead. Let $\mathrm{val}_{t_1'}(u) = (s, q_{f_1 j_1 \dots f_m})$, for this state we use the modified rule

$$s(f(x_1,\dots,x_n)) \to w[\dots, t'|_{u'}, \dots],$$

to construct the corresponding rule in $\mathcal{T} \times A$ by using $t'|_{u'}$ as fixed output instead of $s_i(x_{j_i})$. We now show the correctness of this construction. Since $\mathrm{val}_{t_1'}(u) = (s, q_{f_1 j_1 \dots f_m})$, we know that the unique run ρ of $\mathcal{A}$ on $t \otimes t'$ yields $\rho(u) = q$ and thus we can also compute $\rho(u.j) = q'$ with $u.j \sqsubseteq u'$ and $j \neq j_1$. Towards a contradiction, assume it is not possible to choose a fixed output for the above rule. Then there exists some $t_{alt} \in T_\Sigma$ such that $t[\circ/u'] \cdot t_{alt} \in dom(R)$ and $(t|_{uj}[\circ/u'] \cdot t_{alt}) \otimes (t'|_{uj}[\circ/u'] \cdot t'|_{u'}) \notin T(\mathcal{A}_{q'})$. Consider the modified input tree t_{in} that results from t by replacing $t|_{u'}$ by t_{alt}, i.e., $t_{in} := t[\circ/u'] \cdot t_{alt}$ which is also in the domain of R. Due to $\mathrm{dom}_{t_{v'}} \cap \mathrm{dom}_{t_{u'}} = \emptyset$, we can conclude that the original automaton $\mathcal{T}$ on t_{in} still produces $t'|_{u'}$ which is then mapped to u', i.e., $\mathcal{T}(t_{in})|_{u'} = \mathcal{T}(t)|_{u'} = t'|_{u'}$. Since $\mathcal{T}$ uniformizes R, we obtain $(t|_{uj}[\circ/u'] \cdot t_{alt}) \otimes (t'|_{uj}[\circ/u'] \cdot t'|_{u'}) \in T(\mathcal{A}_{q'})$. This is a contradiction.

The construction of $\mathcal{T} \times \mathcal{A}$ can be continued as described above. Eventually, it is finished because all components of $\mathcal{T}$ and $\mathcal{A}$ are finite, and additionally the delay of $\mathcal{T}$ is bounded to K which guarantees that the state space of $\mathcal{T} \times \mathcal{A}$ is also finite.

We have seen that we can construct a transducer $\mathcal{T}' := \mathcal{T} \times \mathcal{A}$ as described above such that for each $t \in dom(R)$ and each reachable configuration (t, t', φ) of $\mathcal{T}'$ holds $\forall u \in D_{t'}$ with $\varphi(u) = v : u \sqsubseteq v$.

□

The following remark shows that if at one point the output becomes ahead of the input, then it is possible to postpone the output such that it is produced at some point later in time. We will see that postponing the output is enough for our purposes.

Remark 4.12 *Given $n \in \mathbb{N}$ and a TDT $\mathcal{T}$ such that for each $t \in T_\Sigma$ and each reachable configuration (t, t', φ) of $\mathcal{T}$ holds:*

$$\forall u \in D_{t'} \text{ with } \varphi(u) = v : u \sqsubseteq v \text{ or } v \sqsubseteq u.$$

Then, there also exists an equivalent TDT $\mathcal{T}'$ such that for each $t \in T_\Sigma$ and each reachable configuration (t, t'', φ') of $\mathcal{T}'$ holds:

$$\forall u \in D_{t''} \text{ with } \varphi'(u) = v : u \sqsubseteq v.$$

Proof. Let $\mathcal{T} = (Q, \Sigma, \Gamma, q_0, \Delta)$ be a DTDT and let $c_1 = (t, t_1, \varphi_1)$, $c_2 = (t, t_2, \varphi_2)$ be configurations of $\mathcal{T}$ with $c_1 \to_{\mathcal{T}} c_2$ resulting from the application of a transition r. Thereby producing output at $u \in D_{t_1}$ with $\varphi_1(u) = v$ such that $u \sqsubseteq v$ and there exists at least one $u' \in D_{t_2}$ such that $u \sqsubset u'$ and $\varphi_2(u') \sqsubset u'$ with

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

where $f \in \Sigma_i$, w is an n-context over Γ, $j_1, \dots, j_n \in \{1, \dots, n\}$ and $q, q_1, \dots, q_n \in Q$. Consider $j \in \{1, \dots, n\}$ and $f \in \Sigma_i$ $(i \geq 0)$, and let

$$r_{jf} = q_j(f(x_1, \dots, x_i)) \to z_{jf}[\dots] \in \Delta$$

be all transitions that possibly could be applied afterwards to reach a successor configuration of c_2.

The idea is to split the output of r such that at first only the output up to v is produced and the rest of the output is passed on to the successor transitions. Without loss of generality, we assume that for each $u' \in D_{t_2}$ with $u \sqsubset u'$ holds $\varphi_2(u') \sqsubset u'$. For the formal construction, consider r from above. By assumption, the right-hand side can be written in the following form

$$s \cdot g\big(w_1[q_{11}(x_{j_1}), \dots, q_{1i_1}(x_{j_1})], w_2[q_{21}(x_{j_2}), \dots, q_{2i_2}(x_{j_2})], \dots \\ \dots, w_m[q_{m1}(x_{j_m}), \dots, q_{mi_m}(x_{j_m})]\big),$$

where $s \in S_\Gamma$, $g \in \Gamma_m$, $w_1, \dots, w_m$ are contexts over Γ, and $q_{11}(x_{j_1}), \dots, q_{mi_m}(x_{j_m}) = q_1(x_{j_1}), \dots, q_n(x_{j_n})$ such that for $s' := t_1[\circ/u] \cdot s$ holds $\mathrm{val}_{s'}(v) = \circ$.

Note that a right-hand side that does not follow this form leads to some $u' \in D_{t_2}$ with $u' \not\sqsubseteq \varphi_2(u')$, but $\varphi_2(u') \not\sqsubset u'$. We replace r by a new rule r' build as follows.

$$r' := q(f(x_1, \ldots, x_i)) \to s \cdot g(q_{w_1}(x_{j_1}), \ldots, q_{w_m}(x_{j_m})),$$

with $\{q_{w_1}, \ldots, q_{w_m}\} \cup Q$, and for each newly added state q_{w_k} and each $f \in \Gamma_i$ $(i \geq 0)$ we introduce a new rule

$$r'_{w_k f} := q_{w_k}(f(x_1, \ldots, x_i)) \to w_k[z_{(k1)f}[\ldots], \ldots, z_{(ki_k)f}[\ldots]].$$

For the correctness of the construction, let $c_3 = (t, t_3, \varphi_3)$ be the configuration with $c_1 \to_{\mathcal{T}} c_2 \to^*_{\mathcal{T}} c_3$ that is reached by:

1. Given u, v such that v is on the same tree level or deeper as u with $\varphi(u) = v$, $\text{val}_{t_1}(u) = q$, and $\text{val}_t(v) = f$ holds, then the transition r is applied and produces output at u. This yields c_2.
2. For each u', v', such that $u \sqsubset u'$ and $\varphi_2(u') = v'$, $\text{val}_{t_2}(u) = q_j$, and $\text{val}_t(v') = f$ holds, the transition r_{jf} is applied producing output at u'. This eventually yields c_3.

It is easy to see that in the altered transducer $\mathcal{T}'$ still $c_1 \to^*_{\mathcal{T}'} c_3$ holds. Now, first r' is applied, thereby only output up to v is produced. That is for each u' with $u \sqsubset u', \varphi_2(u') = v'$ as above holds $v' = u'$. Afterwards it is only possible to use a newly added transition $r'_{w_k f}$ to produce output at u' for the corresponding v' with $\text{val}_t(v') = f$. By applying the respective transitions $r'_{w_k f}$ the configuration c_3 is eventually reached.

By replacing r as presented above, we have postponed the output. The same method can be used on the transitions $r'_{w_k f}$ to postpone the output further. This can be continued as long as desired in order to postpone the output further by adding new rules to the transition relation. Eventually, as the transition relation has to be finite, the postponed output is produced.

□

The following example demonstrates this procedure.

Example 4.13 Let Σ be a ranked alphabet given by $\Sigma_1 = \{f, g\}$, and $\Sigma_0 = \{a\}$. Consider the TDT $\mathcal{T}$ given by $(\{s\}, \Sigma, \Sigma, \{s\}, \Delta)$ with $\Delta =$

$$\begin{array}{lrcl} \{ & s(a) & \to & a, \\ & s(g(x_1)) & \to & g(s(x_1)), \\ & s(f(x_1)) & \to & f(g(s(x_1))) \quad \}. \end{array}$$

By applying the last transition the output becomes ahead of the input. Assume we want to postpone the output once, then the construction presented in Remark 4.12 results in the following modified set of transitions with an additional new state s':

$$\begin{array}{rcl}\{ \quad s(a) & \to & a, \\ s(g(x_1)) & \to & g(s(x_1)), \\ s(f(x_1)) & \to & f(s'(x_1)), \\ s(f(x_1)) & \to & f(g(s(x_1))) \\ s'(a) & \to & g(a), \\ s'(g(x_1)) & \to & g(g(s(x_1))), \\ s'(f(x_1)) & \to & g(f(g(s(x_1)))) \quad \}\end{array}$$

Now that we have presented some results as preparation, we are ready to prove the following theorem.

Theorem 4 *Given $K > 0$, it is decidable whether a given D↓TA-recognizable relation with D↓TA-recognizable domain has a uniformization without input validation by a top-down tree transducer with output delay bounded to K.*

To solve the above decision problem, we again consider a safety game between In and Out. The procedure is similar to a decision procedure presented in [CL12], where the question whether a uniformization of an automatic word relation by a subsequential transducer exists, is reduced to the existence of winning strategies in a safety game. Let us recall the safety game presented in Section 4.2, we basically extend the game graph such that the vertices can additionally keep track of the input that is ahead, which is bounded to K. Player In can play valid inputs and Out can either react with output, or delay the output and react with a direction in which In should continue with his input sequence.

Given alphabets Σ, Γ, let $R \subseteq T_\Sigma \times T_\Gamma$ denote a relation recognized by a D↓TA $\mathcal{A} = (Q_\mathcal{A}, \Sigma_\perp \times \Gamma_\perp, q_0^\mathcal{A}, \Delta_\mathcal{A})$ and let the domain of R be recognized by a D↓TA $\mathcal{B} = (Q_\mathcal{B}, \Sigma, q_0^\mathcal{B}, \Delta_\mathcal{B})$. Formally, the game graph $G^K_{\mathcal{A},\mathcal{B}}$ is constructed as follows.

- $V_{\mathsf{In}} \subseteq \{\big((q,p),\pi j\big) \in (Q_\mathcal{A} \times Q_\mathcal{B}) \times \mathrm{Path}_\Sigma \cdot \mathrm{dir}_\Sigma \mid \|\pi\| \leq K, \pi \in \mathrm{Path}_\Sigma, j \in \mathrm{dir}_\Sigma\} \cup 2^{Q_\mathcal{A} \times Q_\mathcal{B}}$ is the set of vertices of player In
- $V_{\mathsf{Out}} \subseteq \{\big((q,p),\pi\big) \in (Q_\mathcal{A} \times Q_\mathcal{B}) \times \mathrm{Path}_\Sigma \mid \|\pi\| \leq K\}$ is the set of vertices of player Out
- From a vertex of In the following moves are possible:
  - $\big((q,p),\pi j\big) \to \big((q,p),\pi j f\big)$ for each $f \in \Sigma$ such that $\mathcal{B} : p \xrightarrow{\pi}_j p'$ and there exists $(p', f, p_1, \dots, p_i) \in \Delta_\mathcal{B}$ if $\|\pi\| < K$
  - $P \to \big((q,p),f\big)$ for each $(q,p) \in P$ and there exists $(p, f, p_1, \dots, p_i) \in \Delta_\mathcal{B}$ and $(q,(f,g),q_1,\dots,q_n) \in \Delta_\mathcal{A}$ with $g \in \Gamma_\perp$
- From a vertex of Out the following moves are possible:
  - $\big((q,p),f\big) \xrightarrow{r} P$ if $f \in \Sigma$ is i-ary, $g \in \Gamma_\perp$ is j-ary and there exists a transition $r = (q,(f,g),q_1,\dots,q_n) \in \Delta_\mathcal{A}$ and a transition $(p, f, p_1, \dots, p_i) \in \Delta_\mathcal{B}$ such that if
    - $j \leq i$: for all $t \in T(\mathcal{B}_{p_k}) : t \otimes \perp \in T(\mathcal{A}_{q_k})$ for all $i < k \leq j$, then $P = \{(q_1,p_1),\dots,(q_j,p_j)\}$, and if

  * $j > i$: there exist trees $t_{i+1}, \dots, t_j \in T_{\Gamma_\perp}$ such that $\perp \otimes t_k \in T(\mathcal{A}_{q_k})$ for all $i < k \leq j$, then $P = \{(q_1, p_1), \dots, (q_i, p_i)\}$

- $\big((q,p), \pi j' f'\big) \xrightarrow{r} \big((q',p'), \pi' j' f'\big)$ for each $g \in \Gamma_\perp$ such that $\pi = f j \pi'$, there is $(p, f, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$ with $p' = p_j$ and there is $r = (q, (f,g), q_1, \dots, q_n) \in \Delta_{\mathcal{A}}$ with $q' = q_j$, and for each $k \neq j$ with $k \in \{1, \dots, n\}$ holds
  * if $k \leq rk(f), rk(g)$, then $\exists t' \in T_\Gamma \forall t \in T(\mathcal{B}_{p_k}) : t \otimes t' \in T(\mathcal{A}_{q_k})$
  * if $rk(g) < k \leq rk(f)$, then $\forall t \in T(\mathcal{B}_{p_k}) : t \otimes \perp \in T(\mathcal{A}_{q_k})$
  * if $rk(f) < k \leq rk(g)$, then $\exists t' \in T_\Gamma : \perp \otimes t' \in T(\mathcal{A}_{q_k})$
- $\big((q,p), \pi j f\big) \to \big((q,p), \pi j f j'\big)$ for each $j' \in \{1, \dots, i\}$ for $f \in \Sigma_i$ if $\|\pi j f\| < K$

- The initial vertex is $\{(q_0^{\mathcal{A}}, q_0^{\mathcal{B}})\}$

Note that the game graph can effectively be constructed, because Lemma 4.10 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 $P \in V_{\mathsf{In}}$ such that there is $(q,p) \in P$ and $f \in \Sigma$ such that $(p, f, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$, but there exists no $(q, (f,g), q_1, \dots, q_n) \in \Delta_{\mathcal{A}}$ for some $g \in \Gamma_\perp$, and
- 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 $\mathcal{G}^K_{\mathcal{A},\mathcal{B}} = (G^K_{\mathcal{A},\mathcal{B}}, V \setminus B)$ as safety game for Out.

Example 4.14 Let Σ be given by $\Sigma_1 = \{f, g\}$ and $\Sigma_0 = \{a\}$. We consider the relation $R = T_\Sigma \times T_\Sigma$ and construct a game graph for R without output delay, i.e., $K = 1$. Therefore, let $\mathcal{B}$ recognize $dom(R)$ with only one state $\{p\}$, and R is recognized by $\mathcal{A} = (\{q_0, q_1, q_2\}, \Sigma_\perp \times \Sigma_\perp, q_0, \Delta_{\mathcal{A}})$ with $\Delta_{\mathcal{A}} =$

$$
\begin{aligned}
& \{\big(q_0, (a,a)\big), \big(q_0, (\sigma_1, \sigma_2), q_0\big), \big(q_0, (\sigma_1, a), q_1\big), \big(q_0, (a, \sigma_1), q_2\big) \mid \sigma_1, \sigma_2 \in \Sigma_1\} \\
\cup\ & \{\big(q_1, (f, \perp), q_1\big), \big(q_1, (g, \perp), q_1\big), \big(q_1, (a, \perp)\big)\} \\
\cup\ & \{\big(q_2, (\perp, f), q_2\big), \big(q_2, (\perp, g), q_2\big), \big(q_2, (\perp, a)\big)\}.
\end{aligned}
$$

The result of the construction is shown in Figure 4.3.

The following two lemmata show that from the existence of a winning strategy a top-down tree transducer without input validation that uniformizes the relation can be obtained and vice versa.

Lemma 4.15 *If Out has a winning strategy in $\mathcal{G}^K_{\mathcal{A},\mathcal{B}}$, then the relation R has a uniformization without input validation by a TDT in which the output delay is bounded to K.*

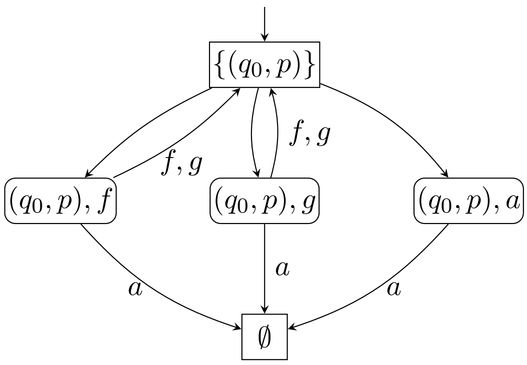


Figure 4.3: The game graph $G^1_{\mathcal{A},\mathcal{B}}$ obtained from the D↓TAs $\mathcal{A}$ and $\mathcal{B}$ from Example 4.14.

Proof. Assume that Out has a winning strategy in the safety game $\mathcal{G}^K_{\mathcal{A},\mathcal{B}}$, then there is also a positional one. We can represent a positional winning strategy by a function $\sigma : V_{\mathsf{Out}} \to \Delta_{\mathcal{A}} \cup \mathrm{dir}$, 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 TDT $\mathcal{T} = (Q, \Sigma, \Gamma, q_0, \Delta)$ from such a positional winning strategy σ as follows:

1. $\big((q,p), \pi j\big) \in Q$ for each $\big((q,p), \pi j f\big) \in V_{\mathsf{Out}}$
2. $q_0 := (q_0^{\mathcal{A}}, q_0^{\mathcal{B}})$
3. For each $\sigma : \big((q,p), f\big) \overset{r}{\mapsto} P$ with $r = (q, (f,g), q_1, \dots, q_n) \in \Delta_{\mathcal{A}}$ and $(p, f, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$:
   (a) add $(q,p)(f(x_1,\dots,x_i)) \to g((q_1,p_1)(x_1),\dots,(q_j,p_j)(x_j))$ to Δ if $j \leq i$, or
   (b) add $(q,p)(f(x_1,\dots,x_i)) \to g((q_1,p_1)(x_1),\dots,(q_i,p_i)(x_i), t_{i+1},\dots,t_j)$ 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 $\big((q,p), f\big)$.
4. For each $\sigma : \big((q,p), \pi j f\big) \mapsto \big((q,p), \pi j f j'\big)$ add $\big((q,p), \pi j\big)(f(x_1,\dots,x_i)) \to \big((q,p), \pi j f j'\big)(x'_j)$ to Δ.

If the strategy σ defines a sequence of moves of Out inside vertices of V_{Out}, 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 over Γ dependent on the used transition r. Eventually, the strategy defines a move of Out to a node of V_{In}, otherwise σ is not a winning strategy. These parts of the strategy are transformed as follows:

5. For each $\big((q,p),\pi j f\big) \overset{r_1}{\to} \dots \overset{r_{k-1}}{\to} \big((q',p'),\pi' j f\big) \overset{r_k}{\to} \big((q',p'),\pi' j f j'\big)$ add $\big((q,p),\pi j\big)(f(x_1,\dots,x_i)) \to s_1 \cdot \dots \cdot s_{k-1} \cdot \big((q',p'),\pi' j f j'\big)(x'_j)$ to Δ, where each $s_i \in S_\Gamma$ is a special tree corresponding to the edge constraints in the ith move.

6. For each $\big((q,p),\pi j f\big) \overset{r_1}{\to} \dots \overset{r_{k-1}}{\to} \big((q',p'),\pi' j f\big) \overset{r_k}{\to} P$ add $\big((q,p),\pi j\big)(f(x_1,\dots,x_i)) \to s_1 \cdot \dots \cdot s_{k-1} \cdot t$ to Δ, where each $s_i \in S_\Gamma$ is a special tree corresponding to the edge constraints in the ith move and t is a tree constructed as described in step 3.

We now verify that $\mathcal{T}$ defines a uniformization without input validation of R. Let $t \in dom(R)$. We show by induction on the number of steps needed to reach a configuration from the initial configuration (t, q_0, φ_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'.

For the induction base, we consider the initial configuration (t, q_0, φ_0) with $q_0 = (q_0^\mathcal{A}, q_0^\mathcal{B})$ and let $\mathrm{val}_t(\varepsilon) = f \in \Sigma_i$. That means, in the game In can move from the initial vertex $\{(q_0^\mathcal{A}, q_0^\mathcal{B})\}$ to $\big((q_0^\mathcal{A}, q_0^\mathcal{B}), f\big)$. Since σ is a winning strategy, the vertex is not in the set of bad vertices and $\sigma\big(\big((q_0^\mathcal{A}, q_0^\mathcal{B}), f\big)\big)$ defines the next move. By construction there exists a rule with left-hand side $(q_0^\mathcal{A}, q_0^\mathcal{B})(f(x_1,\dots,x_i))$ in Δ. Thus, a successor configuration is reachable and the claim holds.

For the induction step, consider a configuration $c_n = (t, t_n, \varphi_n)$ such that $(t, q_0, \varphi_0) \to^n_\mathcal{T} (t, t_n, \varphi_n)$. Assume the claim holds for n. By induction hypothesis the claim is true for c_n, hence there exists a successor configuration $c_{n+1} = (t, t_{n+1}, \varphi_{n+1})$. Let $D_{t_{n+1}} \neq \emptyset$, then there exists u, v with $\varphi(u) = v$ and $\mathrm{val}_{t_{n+1}}(u) = \big((q,p),\pi j\big) \in Q$, $\mathrm{val}_t(v) = f \in \Sigma_i$. By construction of Δ it follows that in a game played according to the winning strategy σ a vertex $\big((q,p),\pi j f\big) \in V_\mathsf{Out}$ is reachable. Thus σ defines the next move, consequently there exists a corresponding transition with left hand side $\big((q,p),\pi j\big)(f(x_1,\dots,x_i))$ in Δ. Thus, there exists c_{n+2} with $c_{n+1} \to_\mathcal{T} c_{n+2}$ and the claim holds.

From the above proof it follows that $(t, q_0, \varphi_0) \to^*_\mathcal{T} (t, t', \varphi)$ with $t' \in T_\Gamma$, i.e., $t' = \mathcal{T}(t)$, because in each step $c \to_\mathcal{T} c'$ one input symbol is read. Eventually, a leaf is reached and the output is a tree over Γ.

To show that $(t, t') \in R$, we can show by induction on the number of steps needed to reach a configuration $c = (t, t'', \varphi)$ such that there exists $u \in D_{t''}$ with $\mathrm{val}_{t''}(u) = \big((q,p),\pi j\big)$ that the unique run $\rho_\mathcal{A}$ of $\mathcal{A}$ on $t \otimes t'$ yields $\rho_\mathcal{A}(u) = q$, the unique run $\rho_\mathcal{B}$ of $\mathcal{B}$ on t yields $\rho_\mathcal{B}(u) = p$ and there exists a transition with left-hand side $(q, \mathrm{val}_{t \otimes t'}(u)) \in \Delta_\mathcal{A}$. Further, if in the next computation step $\mathcal{T}$ produces a fixed tree as output at the ith child of u, that is the output is $t'|_{ui}$, then $t|_{ui} \otimes t'|_{ui} \in T(\mathcal{A}_{q_i})$ for $q_i := \rho_\mathcal{A}(ui)$.

□

We now show the other direction.

Lemma 4.16 *If the relation R has a uniformization without input validation by a TDT in which the output delay is bounded to K, then* Out *has a winning strategy in $\mathcal{G}^K_{\mathcal{A},\mathcal{B}}$.*

Proof. Assume that R has a uniformization without input validation by some TDT $\mathcal{T}$. 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. To construct a winning strategy for Out, it is sufficient to visit each vertex of Out only once. Thus, we can assume that we do not have to consider configurations of $\mathcal{T}$ where the output is ahead, because we can postpone the output as described in Remark 4.12. This has to be done at most $|V_{\mathsf{Out}}|$ times, since we only need to consider plays in order to construct the strategy such that each vertex of Out is visited at most once. Then, Lemma 4.11 implies that R can be uniformized without input validation by some TDT $\mathcal{T}' = (Q, \Sigma, \Gamma, s_0, \Delta)$ such that for a read input sequence (up to length $|V_{\mathsf{Out}}|$) the produced output sequence lies on the same path, i.e., for each reachable configuration $c = (t, t', \varphi)$ holds $\varphi(u) = v$ with $u \sqsubseteq v$ for all $u \in D_{t'}$.

We will construct the strategy inductively. In a play, a vertex $\big((q,p), yjf\big)$ is reached by a sequence of moves that describe a path $xiyjf \in \mathrm{Path}_\Sigma$ with $x, y \in \mathrm{Path}_\Sigma$, $i, j \in \mathrm{dir}$, and $f \in \Sigma$. The strategy in $G^K_{\mathcal{A},\mathcal{B}}$ can be chosen such that in every play according to the strategy for each reached vertex $\big((q,p), yjf\big)$ directly by a move of In holds that if $\mathcal{T}'$ reaches a configuration (t, t', φ) for some $t \in T^{xiyjf}_\Sigma \cap dom(R)$ with $(t, s_0, \varphi_0) \to^*_{\mathcal{T}'} (t, t', \varphi)$ such that there exists $u := path(xi) \in D_{t'}$ with $v := path(xiyjf)$ and $\varphi(v) = u$ such that the following property holds:

- The deterministic run $\rho_\mathcal{A}$ of $\mathcal{A}$ on $t \otimes \mathcal{T}'(t)$ yields $\rho_\mathcal{A}(u) = q$, and
- the deterministic run $\rho_\mathcal{B}$ of $\mathcal{B}$ on t yields $\rho_\mathcal{B}(u) = p$.

We show this by induction on the number of moves played by In. The initial node in $G^K_{\mathcal{A},\mathcal{B}}$ is $\{(q_0^\mathcal{A}, q_0^\mathcal{B})\}$, from there In moves to some $\big((q_0^\mathcal{A}, q_0^\mathcal{B}), f\big)$. Obviously, for the initial configuration (t, s_0, φ_0) with $\varphi_0(\varepsilon) = \varepsilon$ for some $t \in T^f_\Sigma \cap dom(R)$ the claim holds.

For the induction step, assume the claim holds for n. After n moves of In we reach a vertex $\big((q,p), yjf\big)$ of Out. Assume that there exits a configuration $c = (t, t', \varphi)$ of $\mathcal{T}'$ for some $t \in T^{xiyjf}_\Sigma \cap dom(R)$ with $u \in D_{t'}$ and $\varphi(u) = v$ as described above. The induction hypothesis yields $\rho_\mathcal{A}(u) = q$ and $\rho_\mathcal{B} = p$. We distinguish two cases.

(i) Consider $yj = \varepsilon$, then $u = v$. In the next step, that leads to the successor configuration (t, t'', φ'), the transducer can produce output and advance in the input or just advance in the input without producing output.

  (a) In the former case, let $val_{t''}(u) = g$. Then, there exists a transition $r = (q, (f,g), q_1, \dots, q_n) \in \Delta_\mathcal{A}$ with $f \in \Sigma_i, g \in \Gamma_j$. If $j < i$, we have $t \otimes \bot \in T(\mathcal{A}_k)$ for all $k > j$. If $j > i$, the output produced for some $t \in T(\mathcal{B}_{p_k})$ for all $k > i$ is a fixed tree since $\mathcal{T}'$ satisfies the property of Lemma 4.11. Thus, we can choose $\sigma\big(((q,p),f)\big) = r$. The play advances to some $P \in V_{\mathsf{In}}$, then In moves to some $[(q_k, p_k), f]$. The transition r used to reach P was of the form $s(f(x_1,\dots,x_i)) \to g(\dots, s_k(x_k), \dots)$. Therefore, in (t, t'', φ') we have $\varphi'(uk) = vk$, $\rho_{\mathcal{A}}(vk) = q_k$ and $\rho_{\mathcal{B}}(vk) = p_k$. Thus, the claim holds.

(b) In the latter case, the applied transition was of the form $s(f(x_1,\dots,x_i)) \to s'(x_k)$ with $k \in \{1,\dots,i\}$. Then we set $\sigma\big(((q,p),f)\big) = k$. The edge has to exist, since $\mathcal{T}'$ has a delay of maximal K. The next vertex of Out that is reached is $\big((q,p), fkf'\big)$ for some $f' \in \Sigma$. Obviously, in the successor configuration (t, t'', φ) holds $\varphi(u) = vk$, and $\rho_{\mathcal{A}}(u) = q$ and $\rho_{\mathcal{B}} = p$. Consequently, the claim holds.

(ii) Consider $yj \neq \varepsilon$, then $u \sqsubset v$. Again, $\mathcal{T}'$ can either produce output or just advance in the input.

(a) In the next step $\mathcal{T}'$ produces output starting at u. Lemma 4.11 implies that $u' \sqsubseteq v'$ holds for all $u' \in \mathrm{dom}_{t''}$ with $\varphi'(u') = v'$ in the successor configuration (t, t'', φ'). In particular, it follows that every output divergent from the path from u to v is a fixed tree. Therefore, we define the strategy such that Out always chooses the move according to the output produced along v. These moves have to exist since $\mathcal{T}'$ uniformizes R. For each additionally to the symbol at u processed output symbol the delay between output and input decreases by one. In the process, the delay can either be reduced or completely diminished.

In the first case, after processing the last output symbol, Out has reached a vertex $\big((q',p'), y'jf\big)$ and $u' \sqsubset v'$. Let $v' = vk$, $k \in \mathrm{dir}$, then Out has to move to $\big((q',p'), y'jfk\big)$, because the transducer continues to read the input from vk. Subsequently, In moves to a vertex $\big((q',p'), y'jfkf'\big)$ for some $f' \in \Sigma$. Corresponding to this vertex, we have $\rho_{\mathcal{A}}(u') = q'$ and $\rho_{\mathcal{B}}(u') = p'$ and a configuration (t, t'', φ') with $\varphi'(u') = v'$. Thus, the claim holds.

In the second case, the output catches up. Then also an output symbol at v is produced, meaning that Out moves from some $\big((q',p'), f\big)$ such that $\rho_{\mathcal{A}}(v) = q'$ and $\rho_{\mathcal{A}}(v) = p'$ to a vertex P via a transition of the from $(q', (f, \mathrm{val}_{t''}(v)), q_1, \dots, q_n) \in \Delta_{\mathcal{A}}$. From there, In moves to some $\big((q'',p''), f'\big)$ such that $v' = vk$ with $k \in \{1,\dots,i\}$, $\rho_{\mathcal{A}}(v') = q''$, $\rho_{\mathcal{B}}(v') = p''$ with $q'' = q_k$ and $p'' = p_k$. Hence, the claim holds.

(b) Otherwise, if no output is produced, the applied transition was of the from $s(f(x_1,\dots,x_i) \to s_k(x_k)$. Then we set $\sigma\big(((q,p), yf)\big) = k$. The edge has to exist, since $\mathcal{T}'$ has a delay of maximal K. Obviously, the claim is true for (t, t'', φ') with $\varphi'(u) = vk$.

We can conclude that the statement always holds.

We will now describe the strategy if the sequence of moves induce a labeled path $xiyjf$ such that there is no configuration (t, t', φ) with $\varphi(v) = u$ and $u \sqsubseteq v$.

Since in $\mathcal{T}'$ the read input sequence and the produced output sequence are on the same path, this is only the case if $\mathcal{T}'$ reads a prefix of $xiyjf$ and produces a fixed tree as output. From this point on, Out chooses the move corresponding to the output produced along the said path. If the input sequence is longer than the output sequence, then, after all output is processed, the moves corresponding to $\perp$ are taken.

To show that this defines a winning strategy for Out we have to show that no bad vertex is reached. For each vertex of Out that is reached, we defined the strategy. Thus, each reached vertex of Out has outgoing edges. So it suffices to verify that no vertex P is reached that contains a pair (q_j, p_j) such that the input can be continued from p_j, but not from q_j. This is never the case, since for every preceding vertex of Out which is of the form $\big((q, p), f\big)$, we have a configuration (t, t', φ) of $\mathcal{T}'$ such that the run $\rho_\mathcal{A}$ of $\mathcal{A}$ on $t \otimes \mathcal{T}'(t)$ yields $\rho_\mathcal{A}(v) = q$ and the run $\rho_\mathcal{B}$ of $\mathcal{B}$ on t yields $\rho_\mathcal{B}(v) = p$. Thus, there is a transition $(p, f, p_1, \ldots, p_i) \in \Delta_\mathcal{B}$ and if a transition $(p_j, f', p_{j_1}, \ldots, p_{j_i}) \in \Delta_\mathcal{B}$ exists, then also a transition $(q_j, (f', g'), q_{j_1}, \ldots, q_{j_n}) \in \Delta_\mathcal{A}$ must exist.

□

As a consequence of Lemma 4.15 and Lemma 4.16 together with the fact that a winning strategy for Out can effectively be computed in $\mathcal{G}^K_{\mathcal{A},\mathcal{B}}$ we immediately obtain Theorem 4.

Recall the relation R defined in Example 4.14, the TDT $\mathcal{T}$ from Example 4.13 uniformizes the relation. In following example, we present a possible winning strategy for Out in the corresponding game shown in Figure 4.3 that is obtained from $\mathcal{T}$.

Example 4.17 Each time $\mathcal{T}$ from Example 4.13 reads an f it produces two output symbols. To obtain a winning strategy for Out in the game shown in Figure 4.3 each vertex of Out only has to be visited once. Thus, we can use the modified TDT $\mathcal{T}'$ from Example 4.13 to construct the winning strategy σ by simply transferring the outputs $\mathcal{T}'(g) = g$, $\mathcal{T}'(f) = f$, and $\mathcal{T}'(a) = a$ to moves of Out as shown in Lemma 4.16. The result is $\sigma : \big((q_0, p), g\big) \mapsto (q_0, (g, g), q_0) \in \Delta_\mathcal{A}$, $\big((q_0, p), f\big) \mapsto (q_0, (f, f), q_0) \in \Delta_\mathcal{A}$, and $\big((q_0, p), a\big) \mapsto (q_0, (a, a)) \in \Delta_\mathcal{A}$.

4.4.2 Unbounded Output 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 [CL12] 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 formally by introducing the term path-recognizable function. Path-recognizable is meant in the sense that at each point in a path starting from the root for each input symbol there exists exactly one direction in which the path has to be continued to select a matching output tree for the entire input tree.

A function $f_\Pi : T_\Sigma \to T_\Gamma$ is called *path-recognizable function*

$$:\Leftrightarrow \begin{cases} \exists \text{ regular set } \Pi \subseteq \text{Path}_\Sigma \text{ such that } \forall \pi \in \Pi : \pi \in (\Sigma\text{dir})^*\Sigma_0, \\ \forall \pi, \pi' \in \Pi : gcp(\pi, \pi') \in (\Sigma\text{dir})^*, \\ \exists \text{ regular sets } \Pi_1, \dots, \Pi_n \subseteq \Pi \text{ with } \dot{\bigcup}_{i=1}^n \Pi_i = \Pi, \\ \exists\, t_1, \dots, t_n \in T_\Gamma \text{ such that } f_\Pi : t \mapsto t_i \,\forall t \in T_{\Pi_i}, 1 \leq i \leq n. \end{cases}$$

A relation $R \subseteq T_\Sigma \times T_\Gamma$ is said to be *uniformizable by a path-recognizable function without input validation* if $\bigcup_{i=1}^n (T_{\Pi_i} \cap dom(R)) \times \{t_i\}$ defines a uniformization of R.

If a relation $R \subseteq T_\Sigma \times T_\Gamma$ is uniformizable by a path recognizable function without input validation, then R has a uniformization without input validation by a TDT that first reads an unambiguous path of the input tree and then outputs a matching output tree.

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, as preparation, we introduce a partial function that yields the state transformations on a path induced by the input sequence of said path together with some output sequence on the same path of same or smaller length.

For the rest of this section, let $R \subseteq T_\Sigma \times T_\Gamma$ be a relation recognized by a D↓TA $\mathcal{A} = (Q_\mathcal{A}, \Sigma_\perp \times \Gamma_\perp, q_0^\mathcal{A}, \Delta_\mathcal{A})$ and let $dom(R)$ be recognized by a D↓TA $\mathcal{B} = (Q_\mathcal{B}, \Sigma, q_0^\mathcal{B}, \Delta_\mathcal{B})$. Consider a labeled path x over Σ and output y over Γ of same or smaller length on the same path. More formally, for $x \in \text{Path}_\Sigma$, $y \in \text{Path}_\Gamma$ and a direction $i \in \text{dir}_\Sigma$ such that $path(y) \sqsubseteq path(x)$, we define the partial function $\tau_{xi,y} : Q_\mathcal{A} \times Q_\mathcal{B} \to Q_\mathcal{A}$ with:

- $\tau_{xi,y}(q,p) := q'$ if $\mathcal{A} : q \xrightarrow{x \otimes y}_i q'$ and for each uj with $u \in \text{dom}_x : uj \not\sqsubseteq path(xi)$ and $j \in \{1, \dots, rk\big((\text{val}_x^\perp(u), \text{val}_y^\perp(u))\big)\}$ holds
  - if $r := \rho_\mathcal{A}(uj)$ and $s := \rho_\mathcal{B}(uj)$ are defined, then there exists $t' \in T_\Gamma$ such that for all $t \in T(\mathcal{B}_s)$ holds $t \otimes t' \in T(\mathcal{A}_r)$, and
  - if $r := \rho_\mathcal{A}(uj)$ is defined and $\rho_\mathcal{B}(uj)$ is undefined, then there exists $t' \in T_\Gamma$ such that $\perp \otimes t' \in T(\mathcal{A}_r)$,

  where $\rho_\mathcal{A}$ is the run of $\mathcal{A}_q$ on $x \otimes y$ and $\rho_\mathcal{B}$ is the run of $\mathcal{B}_p$ on x.

Lemma 4.10 implies, that it is decidable whether $\tau_{xi,y}(q)$ is defined. Basically, if $q' := \tau_{xi,y}(q,p)$ is defined, then there exists a fixed (partial) output tree $s' \in S_\Gamma^{yi\circ}$ such that for each input tree $t \in T_\Sigma^x \cap dom(R)$ there exists some $t' \in T_\Gamma$ such that $t \otimes (s' \cdot t') \in T(\mathcal{A}_q)$.

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

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

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

The following remark shows that there exists a bound K such that each labeled path of length at least K contains an idempotent factor.

Remark 4.18 *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 [Ram30] 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. [Die00].

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 $j_1, \dots, j_{n-1} \in \mathrm{dir}$. Consider the complete graph $G = (V, E, col)$ with edge-coloring $col : E \to Cols$, where $V := \{f_i j_i \mid 1 \leq i < n\}$, $E := V \times V$, $Cols$ is the finite set of profiles and $col(e) := P_{f_i j_i \dots f_k j_k}$ if $e = (f_i j_i, f_k j_k)$ for all $e \in E$. If there exist $i, j, k \in \mathbb{N}$ with $i < j < k \leq n$ such that the edges $(f_i j_i, f_k j_k)$, $(f_i j_i, f_j j_j)$ and $(f_j j_j, f_k j_k)$ 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.

□

For the rest of this section 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

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

for $n \in \mathbb{N}$.

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

Lemma 4.19 *For $q \in Q_\mathcal{A}$ $p \in Q_\mathcal{B}$ with $dom(R_q) = T(\mathcal{B}_p)$, $x, y \in \mathrm{Path}_\Sigma$, $i, j \in \mathbb{N}$ with $y \neq \varepsilon$ and yj idempotent, it is decidable whether R_q^{xiy} can be uniformized by a path-recognizable function f_Π without input validation such that $xiyj$ is a prefix of each $\pi \in \Pi$.*

Proof. Let $path(x) = u$, $path(y) = v$. First, we show if R_q^{xiy} is uniformized by a path-recognizable function without input validation that is recognized by a TDT $\mathcal{T}$ without input validation, then R_q^{xiy} can also be uniformized by a path-recognizable function without input validation recognized by a TDT $\mathcal{T}'$ without input validation such that $||\mathcal{T}'(t)||^{uiv} \leq |uiv|$ for all $t \in dom(R_q^{xiy})$.

Let $f_\Pi = \bigcup_{k=1}^{n}(T_{\Pi_k} \cap dom(R_q^{xiy})) \times \{t_k\}$ uniformize R_q^{xiy} such that $xiyj$ is a prefix of each $\pi \in \Pi$. For any $k \in \{1,\dots,n\}$ such that $||t_k||^{uiv} > |uiv|$ consider for an arbitrary $xiyjz \in \Pi_k \cap T(\mathcal{B}_p)$ with $path(z) = w$ an arbitrary tree $t \in T_\Sigma^{xiyjz} \subseteq T_{\Pi_k}$. Since yj is idempotent, $t^n \in dom(R_q^{xy})$. We choose n such that $||\mathcal{T}(t^n)||^{ui(vj)^n w} < |ui(vj)^n|$. This is always possible, because for each $t \in dom(R_q^{xiy})$ holds $||\mathcal{T}(t)|| \leq max\{||t_1||,\dots,||t_n||\}$. Let $o = out_{\mathcal{T}}(t^n, ui(vj)^n w)$, and let $o = o'o''$ such that $||o'|| = ||x||$. Using this factorization, we now construct some $t'_k \in T_\Gamma$ with $||t'_k||^{uivj} \leq |uivj|$. Since yj is idempotent, $P_{yi} = P_{(yi)^n}$, i.e., there exists some $m \in \mathrm{Path}_\Gamma$ with $||m|| \leq ||y||$ such that $\tau_{y,m} = \tau_{y^n,o''}$. For $o'm$, let t'_k be the corresponding tree in $T_\Gamma^{o'm}$. Thus, the run ρ_q of $\mathcal{A}_q$ on $t^n \otimes \mathcal{T}(t^n)$ results in the same state at $ui(vj)^n$ as the run ρ_q of $\mathcal{A}$ on $t \otimes t'_k$ at $uivj$. Consequently, $(t, t'_k) \in R_q^{xiy}$.

We now show that $(t', t'_k) \in R_q^{xiy}$ for any other $t' \in T_{\Pi_k}$ and thus it suffices to replace t_k with some suitable t'_k to obtain a uniformization of R_q^{xiy} such that $||t'_k||^{uivj} \leq |uivj|$ for each $k \in \{1,\dots,n\}$. If Π is of the form $L(xi(vj)^+) \cdot \Pi'$, then $\mathcal{T}(t_d) = \mathcal{T}(t_d^n)$ for all $t_d \in dom(R_q^{xiy})$. In this case, since $\mathcal{T}(t) = \mathcal{T}(t')$ we obtain $\mathcal{T}(t^n) = \mathcal{T}(t'^n)$ and it follows $(t', t'_k) \in R_q^{xiy}$. Otherwise, if Π is not of this form, there exists $m \in \mathbb{N}$ such that for each $n > m$ and each tree $t_d \in T_\Sigma^{xiy}$ holds that a configuration of $\mathcal{T}$ for t_d^n is reachable with $\varphi(v) = \varepsilon$ and $v \not\sqsubseteq xi(yj)^n$ with $v \in \mathrm{dom}_{t_d^n}$. This means $\mathcal{T}(t_d^n)$ is independent of $\hat{t}_d$ for each $n > m$. Wlog, we can assume that we have chosen $n > m$ in order to find t'_k as alternative output for $t = s_x \cdot s_y \cdot \hat{t}$. Consider z' with $xiyjz' \in \Pi_k$ and an arbitrary tree $t' = s'_x \cdot s'_y \cdot \hat{t}' \in T_\Sigma^{xiyjz'} \subseteq T_{\Pi_k}$. Since $\mathcal{T}\left(s_x \cdot s_y^n \cdot \hat{t}\right) = \mathcal{T}\left(s_x \cdot s_y^n \cdot \hat{t}'\right)$, it follows directly that $(s_x \cdot s_y \cdot \hat{t}', t'_k) \in R_q^{xiy}$, but then also $(s'_x \cdot s'_y \cdot \hat{t}', t'_k) = (t', t'_k) \in R_q^{xiy}$.

Secondly, we describe a process how to check whether there exists such a uniformization. Our requirement is that for an arbitrary tree $t \in dom(R_q^{xiy}) \subseteq T_\Sigma^{xiy}$ only labeled paths beginning with $xiyj$ are relevant. Furthermore, we know that if such a uniformization exists, then it is sufficient to consider output trees in which the length on the path uiv is restricted to $|uiv|$. Therefore, let

$$O := \{o \in \mathrm{Path}_\Gamma \mid path(o) \sqsubseteq path(xiy) \wedge \tau_{xiyj,o}(q,p) = r \text{ for some } r \in Q_\mathcal{A}\}$$

be the set of possible outputs for xiy of same or shorter length that fulfill this requirement. For each $o \in O$, let t_o denote a corresponding output tree. The set O identifies a finite set of possible output trees. We first check if for each $t \in dom(R_q^{xiy})$ there exists an $o \in O$ such that $(t, t_o) \in R_q^{xiy}$. Hence, let

$$T := \bigcup_{o\in O}\{t \in T_\Sigma \mid (t, t_o) \in R_q^{xiy}\} = \bigcup_{o\in O}\left(R_q^{xiy} \cap (T_\Sigma \times \{t_o\})\right)$$

be the set of trees that are possible input trees for some output tree, and check whether $T = dom(R_q^{xiy})$ holds. If this is the case, we can continue, otherwise there exists no uniformization by a path-recognizable function. This is decidable, because a tree automaton for T can be constructed from an automaton for R_q^{xiy} (using that regular tree languages are closed under intersection and union).

Since the length of the output is restricted, we know that for each t with $(t, t_o) \in R_q^{xiy}$ for some $o \in O$ holds that $(t \times t_o)|_{uivj}$ is of the form $t|_{uivj} \times \bot$ and since $\tau_{xiyj,o}(q,p) = r \in Q_{\mathcal{A}}$ we have $t|_{uivj} \times \bot \in T(\mathcal{A}_r)$. Let $\mathcal{A}_r^{\bot}$ denote the automaton that results from $\mathcal{A}_r$ by removing every transition in which the second component of the letter is not $\bot$. Let

$$P := \bigcup_{o \in O} \{r \in Q_{\mathcal{A}} \mid \tau_{xiyj,o}(q,p) = r\}$$

be the set of states that $\mathcal{A}$ can reach at $uivj$ on the input xiy together with some output $o \in O$.

We now consider two cases, first let $dom(R_q) \neq T_\Sigma$, and secondly let $dom(R_q) = T_\Sigma$.

(i) Let $dom(R_q) \neq T_\Sigma$. We define the D↓TA $\mathcal{A}' = (2^Q_{\mathcal{A}}, \Sigma, P, \Delta')$ by a top-down subset construction with:

- $(R, f, R_1, \ldots, R_i) \in \Delta'$ if $R_j = \{r_j \mid \exists r \in R \text{ with } (r, (f, \bot), r_1, \ldots, r_i) \in \Delta_{\mathcal{A}}\}$ for each $j \in \{1, \ldots, i\}$.

If for each transition $(R, f, R_1, \ldots, R_i) \in \Delta'$ holds that there exists at most one $j \in \{1, \ldots, i\}$ such that there exists $r, r' \in R_j$ with $T(\mathcal{A}_r^{\bot}) \neq T(\mathcal{A}_{r'}^{\bot})$, then the output was dependent on an unambiguous path through the input. From the fact that it is decidable whether two tree automata recognize the same tree language, it follows that it is decidable whether R_q^{xiy} is uniformizable by a path-recognizable function without input validation.

(ii) Let $dom(R_q) = T_\Sigma$, then it is not necessary to construct a subset automaton. Instead, it holds that if each transition in $\mathcal{A}_r^{\bot}$ has at most one non-trivial successor state, then the output was dependent on an unambiguous path through the input. In this case, let $\Pi_r(\mathcal{A}_r^{\bot})$ denote the induced language of unambiguous labeled paths. If for each $r \in P$ this is the case and also the union of all induced labeled paths languages still is a language of unambiguous labeled path, then R_q^{xiy} is uniformizable by a path-recognizable function. $\mathcal{A}_r^{\bot}$ has at most one non-trivial successor state in each transition if, and only if, $T(\mathcal{A}_r^{\bot})$ and $\overline{T(\mathcal{A}_r^{\bot})}$ are D↓TA-recognizable (see Remark 4.20). We obtain that R_q^{xiy} can be uniformized by a path-recognizable function such that $xiyj$ is a prefix of each labeled path if

- $T(\mathcal{A}_r^{\bot})$ and $\overline{T(\mathcal{A}_r^{\bot})}$ are D↓TA-recognizable for all $r \in P$,
- For a DFA $\mathcal{C} = (Q_{\mathcal{C}}, \Sigma \cup \mathrm{dir}, \Delta_{\mathcal{C}}, F_{\mathcal{C}})$ that recognizes $\bigcup_{r \in P} \Pi_r(\mathcal{A}_r^{\bot})$ holds for all $q \in Q_{\mathcal{C}}$ and for all $i, j \in \mathrm{dir}$:

$$\text{If } (q, i, q') \in \Delta_{\mathcal{C}} \wedge (q, j, q'') \in \Delta_{\mathcal{C}}, \text{ then } i = j \wedge q = q'.$$

This is decidable, because D↓TA-recognizability for regular tree-languages is decidable.

A path-recognizable function can then be obtained from the induced language of unambiguous paths in part (i) resp. part (ii).

□

The next Remark connects path-recognizable functions to the form of a D↓TA. Intuitively, if a D↓TA recognizes a set of unambiguous labeled paths (that defines the domain of a path-recognizable function), then the transition structure of the D↓TA is very simple.

Remark 4.20 *Let $\mathcal{A}$ be a D↓TA that recognizes a tree language over Σ. It holds that $T(\mathcal{A})$ and $\overline{T(\mathcal{A})}$ are D↓TA-recognizable if, and only if, $\mathcal{A}$ has at most one non-trivial successor state in each transition.*

Proof. It is easy to see that if $\mathcal{A}$ has at most one non-trivial successor state in each transition, then $T(\mathcal{A})$ and $\overline{T(\mathcal{A})}$ are D↓TA-recognizable. The idea is that if $\mathcal{A}$ has at most one non-trivial successor state in each transition, then the transitions induce a set of unambiguous labeled paths. An automaton for $\overline{T(\mathcal{A})}$ only has to test the same set of labeled path in order to accept some $t \notin T(\mathcal{A})$, since $\mathcal{A}$ can only fail on these unambiguous labeled paths.

We prove the other direction by contraposition. Assume that there exists a transition with more than one non-trivial successor state, i.e., $(q, f, q_1, \dots, q_i) \in \Delta$ such that there are $p, p' \in \{q_1, \dots, q_i\}$ with $t \notin T(\mathcal{A}_p), t' \notin T(\mathcal{A}_{p'})$ and $\hat{t} \in T(\mathcal{A}_p), \hat{t}' \in T(\mathcal{A}_{p'})$. Then, there exists a tree $s \in T_\Sigma$ such that $s \in T(\mathcal{A})$ and for the run ρ of $\mathcal{A}$ on s holds for an $u \in \mathrm{dom}_t : (\rho(u), \mathrm{val}_t(u), \rho(u1), \dots, \rho(ui)) \in \Delta$ with $\rho(u) = q$, $\rho(uj) = q_j$ for each $j \in \{1, \dots, i\}$ and $\mathrm{val}_t(u) = f$. Let $p := q_j$ and $p' := q_k$, then $(s[\circ/uj] \cdot \hat{t})[\circ/uk] \cdot t'$ and $(s[\circ/uj] \cdot t)[\circ/uk] \cdot \hat{t}'$ are in $\overline{T(\mathcal{A})}$. The path-closure of $\overline{T(\mathcal{A})}$ also contains $(s[\circ/uj] \cdot \hat{t})[\circ/uk] \cdot \hat{t}' \notin \overline{T(\mathcal{A})}$. Thus, $\overline{T(\mathcal{A})}$ is not D↓TA-recognizable.

□

The following Lemma establishes the connection between long output delay and path-recognizable functions.

Lemma 4.21 *Let $q \in Q_\mathcal{A}$, $p \in Q_\mathcal{B}$ with $dom(R_q) = T(\mathcal{B}_p)$, $x, y \in \mathrm{Path}_\Sigma$, $i, j \in \mathbb{N}$ with $path(x) = u$, $path(y) = v$, $y \neq \varepsilon$ and yj idempotent. If R_q^{xiy} is uniformized by a TDT $\mathcal{T}$ without input verification such that $||\mathcal{T}(s_x \cdot s_y^n)||^{ui(vj)^n} \leq |ui|$ 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 without input verification.*

Proof. Consider an arbitrary tree $t \in T_\Sigma^{xiy}$. Since $||\mathcal{T}(s_x \cdot s_y^n)||^{ui(vj)^n} \leq |ui|$ for each n, we can choose n such that $||\mathcal{T}(t^n)||^{ui(vj)^n} < |ui(vj)^n|$. With the same argumentation as used in the proof of Lemma 4.19, we can show that there exists $t' \in T_\Gamma$ such that $||t'||^{uiv} \leq |uiv|$ and $(t, t') \in R_q^{xiy}$. There are only finitely many of these t's. Now, we show that there exists a path-recognizable function which is a uniformization without input validation of R_q^{xiy} that maps the input trees to a matching t'.

In Lemma 4.11 we have seen, that if the transducer is not required to validate the input tree, the output at a node is only dependent on the input read so far, on the same path. We use this characteristic to construct a DFA that recognizes the set of unambiguous labeled paths through an input tree t which are relevant to determine a suitable t'.

To choose a short output tree t' for some input tree t the idempotent factor is repeated as many times as necessary in t such that the produced output is shorter than the iteration of the idempotent factor. Then, t' is chosen according to the output on that path. Hence, we are only interested in the labeled path through t that is relevant for the output which is mapped to the iteration of the idempotent factor.

The idea behind the DFA is that if the idempotent factor is repeated often enough, say n times, then the output (on $xi(yj)^n$) is shorter than the repetition and thus, for some $t = s_x \cdot s_y \cdot \hat{t}$ we can keep track which path of $\hat{t}$ is read to produce output on $xi(yj)^n$ in $s_x \cdot s_y^n$. Therefore, we encode in the state space of the DFA the current state of $\mathcal{T}$ at the read input symbol in $\hat{t}$, the current state of $\mathcal{A}$ as well as the current position in x resp. in the repetition of y to which the next output of $\mathcal{T}$ is mapped. Since the current position in x resp. (in a repetition of) y is known, the direction in which the input has to be pursued can be determined from the applied transition of $\mathcal{T}$.

Now, for the construction, let $\mathcal{T} = (Q_\mathcal{T}, \Sigma, \Gamma, q_0^\mathcal{T}, \Delta_\mathcal{T})$. Furthermore, we assume that for every tree $t \in T_\Sigma^{xi(yj)^n y}$ there is a configuration $c = (t, t', \varphi)$ of $\mathcal{T}$ reachable such that there is $u \in \mathrm{dom}_{t'}$ with $\varphi(u) = ui(vj)^n$ for every $n \in \mathbb{N}$. (If this is not the case, for each t the output of t^n for all $n \geq m$ for some $m \in \mathbb{N}$ is independent of $\hat{t}$ as seen in the proof of Lemma 4.19. This means, we can choose the same output tree for all possible input trees.) In the DFA we will use an initial state of the form $(s_s, q_s, \pi_s d_s, p_s, \diamond)$ indicating that $\mathcal{T}$ is in the state s_s after reading $xi(yj)^n$, $\mathcal{A}$ is in the state q_s, the next output that is produced has to be mapped onto $\pi_s d_s$, the next input symbol that is read by $\mathcal{T}$ has to be compatible to $p_s \in Q_\mathcal{B}$ and $\diamond$ indicates that the next input symbol is currently unknown. Let $xiyj$ be of the form $x_1 i_1 \ldots x_m i y_1 j_1 \ldots y_n j$.

- Let s_n denote the state that $\mathcal{T}$ reaches after reading $ui(vj)^n$, $n \in \mathbb{N}$. Since $Q_\mathcal{T}$ is finite there has to be a state that occurs again and again in the sequence $s_1 s_2 \ldots$. Let s be such a state, and we set s_s to $s \in Q_\mathcal{T}$.
- Since $|out_\mathcal{T}(s_x \cdot s_y^n, ui(vj)^n)| \leq ui$ for each $n \in \mathbb{N}$ there exists some m such that $out_\mathcal{T}(s_x \cdot s_y^n, ui(vj)^n) = out_\mathcal{T}(s_x \cdot s_y^{n+1}, ui(vj)^{n+1})$ for all $n \geq m$. Let $||out_\mathcal{T}(s_x \cdot s_y^m, ui(vj)^m)|| = i$ and x' is a prefix of x such that $||x'|| = i$, and choose $q_s \in Q_\mathcal{A}$ with $\mathcal{A} : q \xrightarrow{x' \otimes out_\mathcal{T}(s_x \cdot s_y^m, ui(vj)^m)}_{i_i} q_s$.
- $\pi_s d_s := x'' j \in \mathrm{Path}_\Sigma \cdot \mathrm{dir}$ with $x'' = x_{i+1} i_{i+1} \ldots x_m$
- $p_s \in Q_\mathcal{B}$ with $\mathcal{B} : p \xrightarrow{x'}_{i_i} p_s$

We are ready for the formal construction. We define the DFA $\mathcal{C} = (Q, \Sigma \cup \mathrm{dir}, q_s, \Delta, F)$ with

- state set $Q := \{Q_{\mathcal{T}} \times Q_{\mathcal{A}} \times P \cdot \mathrm{dir} \times Q_{\mathcal{B}} \times (\Sigma \cup \{\diamond\})\}$ with $P = \{\pi \in \mathrm{Path}_\Sigma \mid ||\pi|| \leq ||xiy||\}$,
- the initial state $q_s := (s_s, q_s, \pi_s d_s, p_s, \diamond)$ as defined above,
- a set of final states $F := \{(s, q, \pi d, p, a) \in Q \mid a \in \Sigma_0\}$,

and the transition relation Δ constructed as follows:

- For each $(s, q, \pi d, p, \diamond) \in Q$: $((s, q, \pi d, p, \diamond), f, (s, q, \pi d, p, f)) \in \Delta$ if there exists $(p, f, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$.
- For each $(s, q, \pi d, p, f) \in Q$ with $s(f(x_1, \dots, x_i)) \to s'(x_{j_1}) \in \Delta_{\mathcal{T}}$: $((s, q, \pi d, p, f), j_1, (s', q, \pi d, p_{j_1}, \diamond)) \in \Delta$ if there exists $(p, f, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$.
- For each $(s, q, \pi d, p, f) \in Q$ such that πd is of the form $x'iyj$ with x' is a suffix of x:

  $$((s, q, \pi' d' \pi'' d'', p, f), j_i, (s', q', \pi'' d'' \pi''' d''', p_{j_1}, \diamond)) \in \Delta$$

  if the following constraints are satisfied:

  - there is $(p, f, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$,
  - there is $s(f(x_1, \dots, x_i)) \to w[\dots, s'(x_{j_i}), \dots] \in \Delta_{\mathcal{T}}$ with $w[\dots, s'(x_{j_i}), \dots] \in T^{z \cdot s'(x_{j_i})}_{\Sigma \cup Q_{\mathcal{T}}}(X)$ such that $path(z) \sqsubseteq path(\pi)$,
  - there is a factorization $\pi d = \pi' d' \pi'' d''$ such that $|\pi'| = |z|$ and $d', d'' \in \mathrm{dir}$ with $\mathcal{A} : q \xrightarrow{\pi' \otimes z}_{j_i} q'$,
  - if $\pi'' d''$ is of the form $x'' iyj$, then $\pi''' d''' = \varepsilon$, else $\pi''' d''' \pi'' d'' = yj$.

- For each $(s, q, \pi d, p, f) \in Q$ such that there exists a factorization $\pi d = z' z''$ with $z'' z' = yj$:

  $$((s, q, \pi' d' \pi'' d'', p, f), j_i, (s', q', \pi'' d'' \pi' d', p_{j_1}, \diamond)) \in \Delta$$

  if the following constraints are satisfied:

  - there is $(p, f, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$,
  - there is $s(f(x_1, \dots, x_i)) \to w[\dots, s'(x_{j_i}), \dots] \in \Delta_{\mathcal{T}}$ with $w[\dots, s'(x_{j_i}), \dots] \in T^{z \cdot s'(x_{j_i})}_{\Sigma \cup Q_{\mathcal{T}}}(X)$ such that $path(z) \sqsubseteq path(\pi)$,
  - there is a factorization $\pi d = \pi' d' \pi'' d''$ such that $|\pi'| = |z|$ and $d', d'' \in \mathrm{dir}$ with $\mathcal{A} : q \xrightarrow{\pi' \otimes z}_{j_i} q'$.

We consider only the reachable part of $\mathcal{C}$. Let $r_1, \ldots, r_n$ be the reachable states of F and let $\mathcal{C}_k$ denote the DFA that results from $\mathcal{C}$ by using r_k as only final state for each $k \in \{1, \ldots, n\}$. $\mathcal{C}$ accepts a set of unambiguous paths. Let $\Pi = \{xiyj\} \cdot L(\mathcal{C})$ and $\Pi_k = \{xiyj\} \cdot L(\mathcal{C}_k)$ for each $r_k \in F$. We show that there exist trees $t'_1, \ldots, t'_n \in T_\Gamma$ such that $\bigcup_{k=1}^{n} (T_{\Pi_k} \cap T(\mathcal{B}_p)) \times \{t'_k\}$ defines a uniformization of R_q^{xiy}.

Consider an arbitrary $\pi_k \in \Pi_k$ and an arbitrary tree $t_k \in \left(T_\Sigma^{\pi_k} \cap T(\mathcal{B}_p)\right)$. Choose n_1 such that $||\mathcal{T}(t_k^{n_1})|| \leq |ui(vj)^{n_1}|$ and a configuration $c = (t_k^{n_1}, t', \varphi)$ of $\mathcal{T}$ such that $\varphi(path(\pi_s)d_s) = ui(vj)^{n_1}$ and $\mathrm{val}_{t'}(path(\pi_s)d_s) = s_s$ is reached. Let t'_k be a matching tree as described above with $||t'_k||^{uiv} \leq |uiv|$. It holds that $(t_k, t'_k) \in R_q^{xiy}$.

We now prove that $(t, t'_k) \in R_q^{xiy}$ for each $t \in T_{\Pi_k} \cap T(\mathcal{B}_p)$. Let $x = x' i_i x''$ with $i_i \in \mathrm{dir}$. Let $y = y_1 j_1 \ldots j_{n-1} y_n$ with $y_1, \ldots, y_n \in \Sigma$ and $j_1, \ldots, j_{n-1} \in \mathrm{dir}$. For $r_k = (s, q_2, y_{i+1} j_{i+1} \ldots y_i j_i, p, a)$ and $path(\pi_k) = w$ consider the factorization $out_{\mathcal{T}}(t_k^{n_1}, ui(vj)^{n_1} w) = o_1 o_2 o_3 o_4$ such that $|o_1| = |x'|$, $|o_2| = |x''|$, and $|o_3| = |(yj)^{n_1 - 1} y_1 \ldots y_i|$. The run of $\mathcal{A}$ on $t_k^{n_1} \otimes \mathcal{T}(t_k^{n_1})$ has the following property:

$$\mathcal{A} : q \xrightarrow{x' \otimes o_1}_{i_i} q_s \xrightarrow{x'' \otimes o_2}_{i} q_1 \xrightarrow{(yj)^{n_1-1} y_1 \ldots y_i \otimes o_3}_{j_i} q_2 \xrightarrow{y_{i+1} \ldots y_n \otimes o_4}_{j} q_3 \xrightarrow{\pi_k \otimes \varepsilon} F_{\mathcal{A}}.$$

Recall, that t'_k is constructed by choosing some $\tau_{yj,o}$ such that $\tau_{(yj)^{n_1}, o_3 o_4} = \tau_{yj,o}$. Therefore, the run of $\mathcal{A}$ on $t_k \otimes t'_k$ looks as follows:

$$\mathcal{A} : q \xrightarrow{x' \otimes o_1}_{i_i} q_s \xrightarrow{x'' \otimes o_2}_{i} q_1 \xrightarrow{y \otimes o}_{j} q_3 \xrightarrow{\pi_k \otimes \varepsilon} F_{\mathcal{A}}.$$

Now, for an arbitrary $t \in T_{\Pi_k} \cap T(\mathcal{B}_p)$ with $t \in T_\Sigma^{xiyj\pi}$ we choose a suitable n_2 such that $||\mathcal{T}(t^{n_2})|| \leq |ui(vj)^{n_2}|$ and a configuration $c = (t^{n_2}, t', \varphi)$ of $\mathcal{T}$ such that $\varphi(path(\pi_s)d_s) = ui(vj)^{n_2}$ and $\mathrm{val}_{t'}(path(\pi_s)d_s) = s_s$ is reached. Let $path(\pi) = w'$, and consider the output of $\mathcal{T}$ on the path $ui(vj)^{n_2} w'$. The beginning and the end of the output on this path are the same as for $t_k^{n_1}$, because in $\mathcal{C}$ reading π leads to the same state as π_k, which means $\mathcal{T}$ produces the same final output at the leaf. Thus, we have $out_{\mathcal{T}}(t^{n_2}, ui(vj)^{n_2} w') = o_1 o'_2 o'_3 o_4$ with $|o'_2| = |o_2|$. Meaning the run of $\mathcal{A}$ on $t^{n_2} \otimes \mathcal{T}(t^{n_2})$ looks as follows:

$$\mathcal{A} : q \xrightarrow{x' \otimes o_1}_{i_i} q_s \xrightarrow{x'' \otimes o'_2}_{i} q'_1 \xrightarrow{(yj)^{n_2-1} y_1 \ldots y_i \otimes o'_3}_{j_i} q_2 \xrightarrow{y_{i+1} \ldots y_n \otimes o_4}_{j} q_3 \xrightarrow{\pi \otimes \varepsilon} F_{\mathcal{A}}.$$

We see, that $(x'' i (yj)^{n_2 - 1} y, o'_2 o'_3 o_4)$ induces the same state transformation on $\mathcal{A}$ from q_s as $(x'' iy, o_2 o)$. Hence, the run of $\mathcal{A}$ on $t \otimes t'_k$ results in

$$\mathcal{A} : q \xrightarrow{x' \otimes o_1}_{i_i} q_s \xrightarrow{x'' \otimes o_2}_{i} q_1 \xrightarrow{y \otimes o}_{j} q_3 \xrightarrow{\pi \otimes \varepsilon} F_{\mathcal{A}},$$

i.e., $(t, t'_k) \in R_q^{xiy}$.

□

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 question "Given a D↓TA-recognizable relation with D↓TA-recognizable domain, has the relation a uniformization by a TDT without input validation?". 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 $G'^K_{\mathcal{A},\mathcal{B}}$ denote the modified game, the modification is explained in the proof of Lemma 4.22.

Lemma 4.22 *Let $R \subseteq T_\Sigma \times T_\Gamma$ be relation that is recognized by a D↓TA $\mathcal{A}$ and its domain is recognized by a D↓TA $\mathcal{B}$. R has a uniformization without input validation if, and only if,* Out *has a winning strategy in the modified safety game $\mathcal{G}'^K_{\mathcal{A},\mathcal{B}} = (G'^K_{\mathcal{A},\mathcal{B}}, V \setminus B)$.*

Proof. The proof is similar to the proofs of Lemma 4.15 and Lemma 4.16. However, we need to make a small adjustment to the game graph $G^K_{\mathcal{A},\mathcal{B}}$ to obtain $G'^K_{\mathcal{A},\mathcal{B}}$. From each vertex $((q,p),\pi) \in V_{\mathsf{Out}}$ 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 yi is idempotent such that R^{xiy}_q can be uniformized by a path-recognizable function without input validation. The set of bad vertices B for Out remains unchanged.

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 4.19 implies that it is decidable whether there exists a corresponding uniformization without input validation.

Assume that Out has a winning strategy in $\mathcal{G}'^K_{\mathcal{A},\mathcal{B}}$, then there also exists a positional winning strategy for Out. To construct a TDT $\mathcal{T}$ that uniformizes R without input validation, we proceed as presented in the proof of Lemma 4.15 with one addition. We construct for each $((q,p),\pi) \in V_{\mathsf{Out}}$ such that $||\pi|| = K$ and there is $\pi = xiyjz$ such that R^{xiy}_q can be uniformized by a path-recognizable function without input validation, a TDT $\mathcal{T}^{xiy}_q$ that uniformizes R^{xiy}_q. In $\mathcal{T}$ we switch to $\mathcal{T}^{xiy}_q$ at the respective states. The correctness proof of the construction is similar to the correctness proof in the case of bounded delay and therefore omitted.

For the other direction, assume that R is uniformized by some TDT $\mathcal{T}$ without input validation. Again, the proof is similar to the proof of Lemma 4.16. Thus, we only describe how the strategy is chosen if the output delay in $\mathcal{T}$ exceeds K and omit a complete proof.

If the play reaches a vertex $((q,p),\pi) \in V_{\mathsf{Out}}$ 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 yi is idempotent. Let $path(x) = u, path(y) = v$ and $path(z) = w$. We assume that $\mathcal{T}$ produces no output in the next computation step, and continues to read the input, say in direction k. Before the current vertex is reached, the play has to visit the vertex $((q,p),\varepsilon) \in V_{\mathsf{In}}$ reached by a sequence of moves of In that induce a path $\pi \in \mathrm{Path}_\Sigma$ starting from the initial vertex. Then, there is a configuration $c = (t, t', \varphi)$ for each $t \in T^\pi_\Sigma$ of $\mathcal{T}$ reachable such that $\varphi(path(\pi)) = path(\pi)$. Let $s = \mathrm{val}_{t'}(path(\pi))$. This means, that $\mathcal{T}$ starting from s uniformizes R_q.

Consider $out_{\mathcal{T}_s}(s_x \cdot s_y^n \cdot s_z, ui(vj)^n)$ for $n \in \mathbb{N}$ and some $s_z \in S_\Sigma^{z \cdot k \cdot \circ}$. We can distinguish two cases.

If $||out_{\mathcal{T}_s}(s_x \cdot s_y^n \cdot \hat{t}, ui(vj)^n)|| < ui(vj)^n$ for all $n \in \mathbb{N}$, then Lemma 4.21 implies that R^{xiy} can be uniformized by a path-recognizable function without input validation. In this case, Out stays in this vertex from then on and wins.

Otherwise, there exists $m \in \mathbb{N}$ such that $||out_{\mathcal{T}_s}(s_x \cdot s_y^m \cdot s_z, ui(vj)^n)|| \geq ui(vj)^n$. Consider the factorization of $out_{\mathcal{T}_s}(s_x \cdot s_y^m \cdot s_z, ui(vj)^n) = o_1 i o_2 j o_3$ such that $|o_1 i| = |xi|$ and $|o_2 j| = (yj)^m$. Since yi is idempotent we can choose some o of length K such that $\mathcal{A} : q \xrightarrow{xiy \times o_1 o}_j q'$ and $\mathcal{A} : q \xrightarrow{xi(yj)^{m-1} y \times o_1 o_2}_j q''$ with $q' = q''$. Then Out makes K moves according to o leading to some $((q', p'), z) \in V_{\mathsf{Out}}$. From there, Out takes the transitions according to o_3. If $|o_3| < z$, then Out makes the move corresponding to taking direction k afterwards, otherwise the next move has to be played by In after processing o_3.

□

As a consequence of Lemma 4.22 and the fact that a winning strategy for Out in $\mathcal{G}'^K_{\mathcal{A},\mathcal{B}}$ can effectively be computed we immediately obtain our main result.

Theorem 5 *It is decidable whether a D↓TA-recognizable relation with D↓TA-recognizable domain has a uniformization without input validation by a top-down tree transducer.*

Before we turn to the next section on uniformization with input validation, we give a short note on the domain of a given relation. Consider a D↓TA-recognizable relation $R \subseteq T_\Sigma \times T_\Gamma$ such that the domain is total, i.e., $dom(R) = T_\Sigma$. Every TDT that implements a uniformization of R without input validation in fact realizes a uniformization of R in the classical sense. Furthermore, if the domain is total, the presented constructions in this chapter can be simplified by leaving out the components introduced by a given domain automaton.

4.5 Uniformization with 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.

First, we will see that in this case it is necessary that read input and produced output may take divergent paths. Secondly, we ask whether a given D↓TA-recognizable relation with D↓TA-recognizable domain has a uniformization by a top-down tree transducer with synchronous input and output, i.e., the transducer produces one output symbol per read input symbol. Therefore, we provide a decision procedure – again a safety game – that takes into account that read input and produced output may take divergent paths. Lastly, we will see that the presented decision procedure is not suitable if we allow asynchronous 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 top-down tree transducer, but not by a top-down-tree transducer such that in every configuration the position to which the produced output is mapped is a prefix of the position of the input symbol under consideration.

Example 4.23 Let Σ be a ranked alphabet given by $\Sigma_2 = \{f\}$ and $\Sigma_0 = \{a, b\}$. Consider the following relation $R_1 \subseteq T_\Sigma \times T_\Sigma$ given by $\{\big(f(b,t), f(t',b)\big) \mid \neg\exists u \in \mathrm{dom}_t : \mathrm{val}_t(u) = b\}$.

Both R_1 and $dom(R_1)$ are D↓TA-recognizable. The domain of the relation is recognized by the D↓TA $\mathcal{B}_1 = (\{p_0, p_1, p_2\}, \Sigma, p_0, \{(p_0, f, p_1, p_2), (p_1, b), (p_2, f, p_2, p_2), (p_2, a)\})$, and the relation is recognized by the D↓TA $\mathcal{A}_1 = (\{q_0, q_1, q_2, q_3, q_4\}, \Sigma_\perp \times \Gamma_\perp, q_0, \Delta_{\mathcal{A}_1})$ with $\Delta_{\mathcal{A}_1} =$

$$\begin{array}{ll} & \{\big(q_0, (f,f), q_1, q_2\big)\} \\ \cup & \{\big(q_1, (b,a)\big), \big(q_1, (b,b)\big), \big(q_1, (b,f), q_3, q_3\big)\} \\ \cup & \{\big(q_2, (a,b)\big), \big(q_2, (f,b), q_4, q_4\big)\} \\ \cup & \{\big(q_3, (\perp,a)\big), \big(q_3, (\perp,b)\big), \big(q_3, (\perp,f), q_3, q_3\big)\} \\ \cup & \{\big(q_4, (a,\perp)\big), \big(q_4, (f,\perp), q_4, q_4\big)\}. \end{array}$$

Intuitively, a TDT $\mathcal{T}$ 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, and therefore has to produce output of the same size. Clearly, the relation R is uniformized by the following TDT $\mathcal{T} = (\{q_0, q_1, q_2\}, \Sigma, \Sigma, q_0, \Delta)$ with $\Delta =$

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

However, there exists no TDT $\mathcal{T}'$ that uniformizes R such that the read input sequence and the produced out are on the same path. Assume such a TDT $\mathcal{T}'$ 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 $\{(t,b) \mid t \in T_\Sigma \wedge \neg\exists u \in \mathrm{dom}_t : \mathrm{val}_t(u) = b\}$. 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, $dom(R(\mathcal{T}'_{q_2}))$ must also contain trees with occurrences of b.

It follows directly from the above example that Lemma 4.11 is invalid if the domain of a considered relation is not total.

4.5.1 Synchronous Input and Output

Before presenting the decision procedure, we define more formally what we mean by synchronous input and output.

Definition 4.24 Consider a TDT $\mathcal{T} = (Q, \Sigma, \Gamma, Q_0, \Delta)$. We say input and output are synchronous in $\mathcal{T}$ if Δ only contains transitions of the form

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

where $f \in \Sigma_i$, $g \in \Gamma_n$, $q, q_1, \dots, q_n \in Q$ and $j_1, \dots, j_n \in \{1, \dots, i\}$.

Now, for the rest of this section, let $\Sigma = \bigcup_{i=0}^{m} \Sigma_i$ be an input alphabet and $\Gamma = \bigcup_{i=0}^{m'} \Gamma_i$ be an output alphabet. Consider a relation $R \subseteq T_\Sigma \times T_\Gamma$ such that R and $dom(R)$ are D↓TA-definable. 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 and $\mathcal{B} = (Q_\mathcal{B}, \Sigma, q_0^\mathcal{B}, \Delta_\mathcal{B})$ be a D↓TA that recognizes $dom(R)$.

In the following we present a game between In and Out such that the game graph besides the transition structure of $\mathcal{A}$ and $\mathcal{B}$ also takes the possibility into account that a TDT can reach configurations where the produced output symbol is not mapped to the read input symbol.

The main differences to the previous section is that the vertices in the game graph keep track of the current state of $\mathcal{B}$ on the input sequence played by In and keep track of the state of $\mathcal{A}$ 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.

For the second part, we define a D↓TA $\mathcal{C}$ that is the cross-product of $\mathcal{B} \times \mathcal{A}$ with an additional state to model that there is no input symbol. Let $\mathcal{C} = (Q_\mathcal{C}, \Sigma_\perp \times \Gamma_\perp, q_0^\mathcal{C}, \Delta_\mathcal{C})$ consist of

- a state set $Q_\mathcal{C} := (Q_\mathcal{B} \cup p_\perp) \times Q_\mathcal{A}$,
- an initial state $q_0^\mathcal{C} := \left(q_0^\mathcal{B}, q_0^\mathcal{A}\right)$, and
- a transition relation $\Delta_\mathcal{C}$ constructed as follows:
  - For $(p, q) \in Q_\mathcal{C}$, $p \in Q_\mathcal{B}$, and $f \in \Sigma$ such that $(p, f, p_1, \dots, p_i) \in \Delta_\mathcal{B}$ and $(q(f, g), q_1, \dots, q_n) \in \Delta_\mathcal{A}$ add

    $$\big((p, q), (f, g), (p_1, q_1), \dots, (p_i, q_i), (p_\perp, q_{i+1}), \dots, (p_\perp, q_n)\big)$$

    to $\Delta_\mathcal{C}$, and
  - for $(p_\perp, q) \in Q_\mathcal{C}$ such that $(q, (\perp, g), q_1, \dots, q_n) \in \Delta_\mathcal{A}$ add

    $$\big((p_\perp, q), (f, g), (p_\perp, q_1), \dots, (p_\perp, q_n)\big)$$

    to $\Delta_\mathcal{C}$.

Furthermore, let $\Delta_\mathcal{B}^\perp$ be the transition relation that is obtained from $\Delta_\mathcal{C}$ by removing the second letter from the transitions.

We are ready to define the game graph $G_{\mathcal{A},\mathcal{B}}$ as follows:

- $V_{\mathsf{In}} \subseteq \bigcup_{i=0}^{m'} \big(Q_{\mathcal{B}} \times (Q_{\mathcal{A}} \cup 2^{Q}_{\mathcal{C}})\big)^{i}$ is the set of vertices of player In.
- $V_{\mathsf{Out}} \subseteq Q_{\mathcal{B}} \times (Q_{\mathcal{A}} \cup 2^{Q}_{\mathcal{C}}) \times \Sigma$ is the set of vertices of player Out.
- From a vertex of In the following moves are possible:
  - $\big((p_1, P_1), \ldots, (p_n, P_n)\big) \rightarrow \big((p_j, P_j), f\big)$ for each $f \in \Sigma$ and each j, $1 \leq j \leq n$ such that there is a rule with left-hand side $(p_j, f) \in \Delta_{\mathcal{B}}$.
- From a vertex $\big((p, q), f\big)$ with $q \in Q_{\mathcal{A}}$ of Out the following moves are possible:
  - From $\big((p, q), f\big)$ with $f \in \Sigma_i$, $i > 0$, for each $g \in \Gamma_j$, $j \geq 0$ and each $j_1, \ldots, j_j \in \{1, \ldots, i\}$

    $$\big((p, q), f\big) \xrightarrow{g} \big((p_{j_1}, P_1), \ldots, (p_{j_j}, P_j)\big),$$

    such that the following constraints are satisfied:

    (i) $\exists\, (p, f, p_1, \ldots, p_i) \in \Delta_{\mathcal{B}}$

    (ii) $\forall k$, $1 \leq k \leq i$ with $k \notin \{j_1, \ldots, j_j\}$ holds $T(\mathcal{B}_{p_k}) = T_{\Sigma}$

    (iii) $\exists\, (q, (f, g), q_1, \ldots, q_n) \in \Delta_{\mathcal{A}}$

    (iv) $\forall k$, $j < k \leq i$ holds $s \times \bot \in T(\mathcal{A}_{q_k})$ for all $s \in T(\mathcal{B}_{p_k})$

    (v) $\forall k$, $1 \leq k \leq j$ holds $P_k = q_k$ if $j_k = k$ else $P_k = \{(p_k, q_k)\}$
  - From $\big((p, q), f\big)$ with $f \in \Sigma_0$, for each $g \in \Gamma_j$, $j \geq 0$ such that $(q, (f, g), q_1, \ldots, q_j) \in \Delta_{\mathcal{A}}$ and there exist trees $t'_1, \ldots, t'_j \in T_{\Gamma}$ such that $\bot \otimes t'_k \in T(\mathcal{A}_{q_k})$ for all $1 \leq k \leq n$:

    $$\big((p, q), f\big) \xrightarrow{g} (\,).$$
- From a vertex $\big((p, P), f\big)$ with $P \in 2^{Q}_{\mathcal{C}}$ of Out the following moves are possible:
  - From $\big((p, P), f\big)$ with $f \in \Sigma_i$, $i > 0$, for each $g \in \Gamma_j$, $j \geq 0$ and each $j_1, \ldots, j_j \in \{1, \ldots, i\}$

    $$\big((p, P), f\big) \xrightarrow{g} \big((p_{j_1}, P_1), \ldots, (p_{j_j}, P_j)\big),$$

    such that (i), (ii) and the following constraints are satisfied:

    (vi) $\forall p'$ s.t. $\exists (p', q) \in P$:
      * If $p' \in Q_{\mathcal{B}}$, $f' \in \Sigma_k$ and $\exists\, (p', f', p'_1, \ldots, p'_k) \in \Delta_{\mathcal{B}}$, then $\exists\, ((p', q), (f', g), (p'_1, q_1), \ldots, (p'_l, q_l)) \in \Delta_{\mathcal{C}}$
      * If $p' = p_{\bot}$, then $\exists\, ((p', q), (\bot, g), (p'_1, q_1), \ldots, (p'_j, q_j)) \in \Delta_{\mathcal{C}}$

    (vii) $\forall p'$ s.t. $\exists (p', q) \in P$:
    If $\exists\, ((p', q), (f', g), (p'_1, q_1), \ldots, (p'_l, q_l)) \in \Delta_{\mathcal{C}}$, then $\forall k$, $j < k \leq l : s \times \bot \in T(\mathcal{A}_{q_k})\ \forall\, s \in T(\mathcal{B}_{p'_k})$

    (viii) $\forall k$, $1 \leq k \leq j$:
    $P_k = \{(p'_k, q_k) \mid \exists\, (p', q) \in P \wedge \exists\, ((p', q), (f', g), (p'_1, q_1), \ldots, (p'_l, q_l)) \in \Delta_{\mathcal{C}}\}$

– From $\big((p,P),f\big)$ with $f \in \Sigma_0$, , for each $g \in \Gamma_j$, $j \geq 0$

$$\big((p,P),f\big) \xrightarrow{g} (\,),$$

such that (vi), (vii) and the following constraint is satisfied

(ix) $\exists\, t'_1,\dots,t'_j \in T_\Gamma$ s.t. $\forall\,(p'_k,q_k) \in P_k$ as defined in (vii): $\forall k,\ 1 \leq k \leq j : s \times t'_k \in T(\mathcal{A}_{q_k})$ for all $s \in T(\mathcal{B}_{p'_k})$

- The initial vertex is $\big((q_0^{\mathcal{B}}, q_0^{\mathcal{A}})\big)$.

The game graph can be constructed, because Lemma 4.10 implies that it is decidable whether the edge constraints hold.

The winning strategy for player Out should express that Out loses the game if she can not continue to produce valid output. This is represented in the game graph by all vertices of Out without outgoing edges. If one of these vertices is reached during a play, Out loses the game. Let B be the set of these bad vertices for Out, we define the game $\mathcal{G}_{\mathcal{A},\mathcal{B}} = (G_{\mathcal{A},\mathcal{B}}, V \setminus B)$ as safety game for Out.

Recall the relation R_1 from Example 4.23. The game graph $G_{\mathcal{A}_1,\mathcal{B}_1}$ constructed from the D↓TAs $\mathcal{A}_1$ and $\mathcal{B}_1$ from Example 4.23 is depicted in Figure 4.4. Out has a winning strategy in the corresponding safety game $\mathcal{G}_{\mathcal{A}_1,\mathcal{B}_1}$ from which a transducer that uniformizes R_1 can be constructed as shown in the next Lemma.

In the following we show that the question whether there exists a uniformization by a TDT with synchronous input and output can be reduced to the question of the existence of a winning strategy for Out in the above mentioned game.

Lemma 4.25 *The relation R can be uniformized by a TDT with synchronous input and output if Out has a winning strategy in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$.*

Proof. Assume Out has a winning strategy in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$, then Out also has a positional winning strategy in the game. Let σ denote such a strategy, each move of Out is uniquely determined by the output symbol and the reached vertex of In.

We construct a deterministic TDT $\mathcal{T} = (Q, \Sigma, \Gamma, q_0, \Delta)$ from σ as follows:

- $Q \subseteq Q_{\mathcal{B}} \times (Q_{\mathcal{A}} \cup 2^{Q_\mathcal{C}})$ is the set of states, with $q_0 := (q_0^{\mathcal{B}}, q_0^{\mathcal{A}})$ as initial state, and
- Δ is the transition relation build as follows:
  – For each $\sigma : \big((p,P),f\big) \overset{g}{\mapsto} \big((p_{j_1},P_1),\dots,(p_{j_n},P_n)\big)$ with $f \in \Sigma_i$, $i > 0$ and $g \in \Gamma_n$, $n \geq 0$ add
  $$(p,P)(f(x_1,\dots,x_i)) \to g((p_{j_1},P_1)(x_{j_1}),\dots,(p_{j_n},P_n)(x_{j_n}))$$
  to Δ, and
  – for each $\sigma : \big((p,P),f\big) \overset{g}{\mapsto} (\,)$ with $f \in \Sigma_0$ and $g \in \Gamma_n$, $n \geq 0$ add
  $$(p,P)(a) \to g(t'_1,\dots,t'_n)$$
  to Δ with $t'_1,\dots,t'_n \in T_\Gamma$ such that $t'_1,\dots,t'_n$ satisfy the constraints of the considered g-edge.

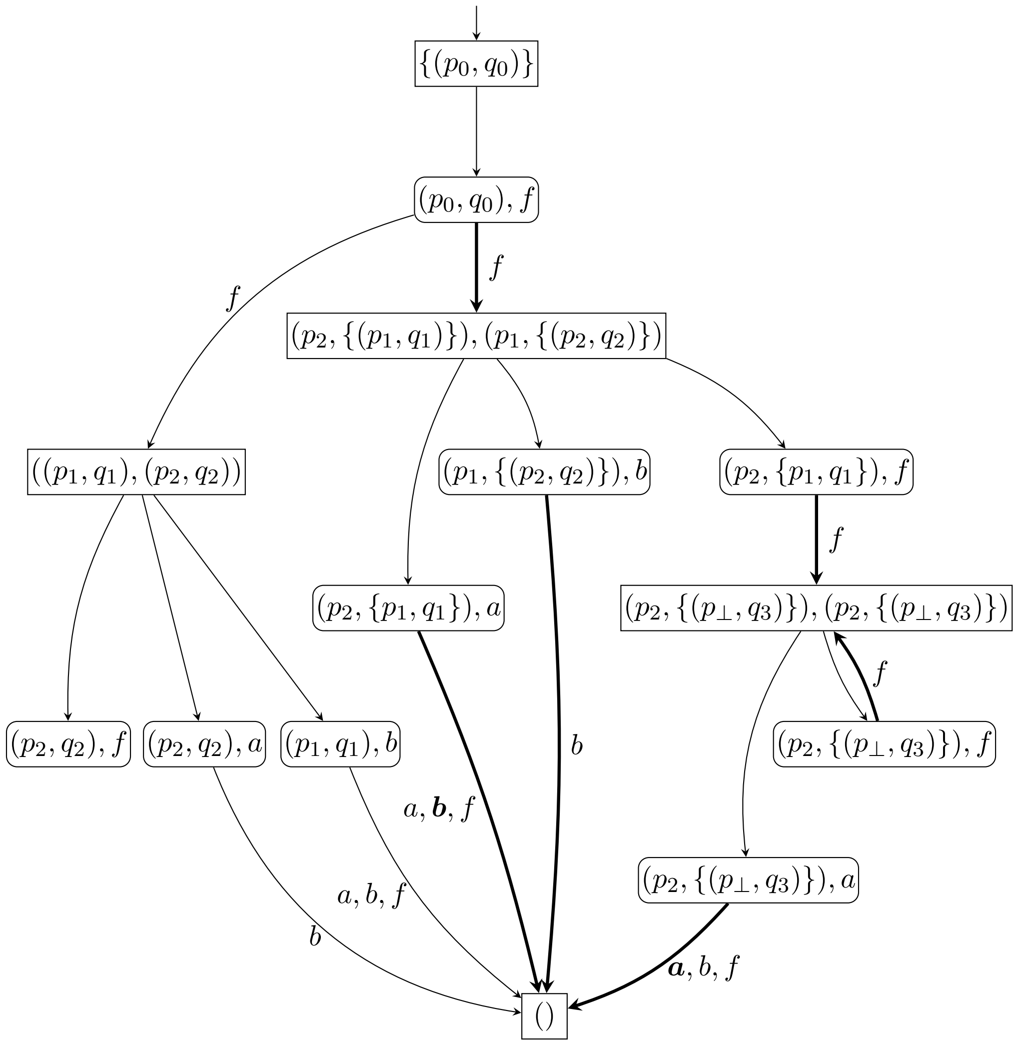


Figure 4.4: The game graph $G_{\mathcal{A}_1,\mathcal{B}_1}$ constructed from $\mathcal{A}_1$ and $\mathcal{B}_1$ from Example 4.23 that recognize R_1 resp. $dom(R_1)$ from Example 4.23. To simplify the graph edges with the same source and destination but with different labels are combined to one edge with multiple labels. A possible winning strategy for Out in $\mathcal{G}_{\mathcal{A}_1,\mathcal{B}_1}$ is emphasized in the graph.

We now verify that $\mathcal{T}$ defines a uniformization of R. First, we prove $dom(R(\mathcal{T})) \subseteq dom(R)$. Let $t \in dom(R(\mathcal{T}))$, we show by induction on the height of a node $v \in \mathrm{dom}_t$ that there exists a configuration (t, t', φ) of $\mathcal{T}$ with $(t, q_0, \varphi_0) \to^*_{\mathcal{T}} (t, t', \varphi)$ such that if there exists $u \in \mathrm{dom}_{t'}$ with $\varphi(u) = v$ and $u = (p, P) \in Q$, then the run ρ of $\mathcal{B}$ on t yields $\rho(v) = p$.

For the induction base, consider $v = \varepsilon$. In the initial configuration (t, q_0, φ_0) we obtain $\varphi_0(\varepsilon) = \varepsilon$ with $\mathrm{val}_{q_0}(\varepsilon) = (q_0^{\mathcal{B}}, q_0^{\mathcal{A}})$ by construction of $\mathcal{T}$. Obviously, $\rho(\varepsilon) = q_0^{\mathcal{B}}$, thus the claim holds.

For the induction step from n to $n + 1$, consider v at height n. Assume that the claim holds n, i.e., the induction hypothesis is true at v. Then, there exists a configuration (t, t', φ) such that $(t, q_0, \varphi_0) \to^*_{\mathcal{T}} (t, t', \varphi)$ with $u \in \mathrm{dom}_{t'}$ such that $\varphi(u) = v$ and $u = (p, P) \in Q$. By induction hypothesis we obtain $\rho(v) = p$. Choose $j, k \in \mathbb{N}$ such that there exists a successor configuration (t, t'', φ') of (t, t', φ) with $uk \in \mathrm{dom}_{t''}$ and $\varphi''(uk) = vj$. Let $\mathrm{val}_t(v) = f \in \Sigma_i$, then the successor configuration was reached by applying a transition of the form $(p, P)\big(f(x_1, \ldots, x_i)\big) \to g((p_{j_1}, P_1)(x_{j_1}), \ldots, (p_{j_n}, P_n)(x_{j_n}))$ and $\mathrm{val}_{t''}(uk) = (p_{j_k}, P_k)$ with $j_k = j$. Hence, by construction of $\mathcal{T}$ we know that in a play according to σ the vertex $\big((p, P), f\big)$ with an outgoing g-edge to $\big((p_{j_1}, P_1), \ldots, (p_{j_n}, P_n))\big)$ is reachable. Thus, from the construction of the game graph it follows that $(p, f, p_1, \ldots, p_i) \in \Delta_{\mathcal{B}}$. Therefore $\rho(vj) = p_j$ and the claim holds.

The above induction implies that for each leaf v of t, such that there exists a configuration (t, t', φ) of $\mathcal{T}$ with $\varphi(u) = v$ for some $u \in \mathrm{dom}_{t''}$ with $\mathrm{val}_{t'}(u) = (p, P)$, holds that $\rho(v) = p$ and $(p, \mathrm{val}_t(v))$ is a final combination. For each leaf v, such that there exists no such configuration, holds that there is a configuration (t, t', φ) with $v' \sqsubseteq v$ and $\varphi(u) = v'$ for some $u \in \mathrm{dom}_{t''}$. Let $\mathrm{val}_t = f \in \Sigma_i$, $\mathrm{val}_{t'} = (p, P)$, and $v = v'jv''$ with $j \in \mathbb{N}, v'' \in \mathbb{N}^*$, then there is a successor configuration reachable by applying a transition of the form $(p, P)\big(f(x_1, \ldots, x_i)\big) \to g((p_{j_1}, P_1)(x_{j_1}), \ldots, (p_{j_n}, P_n)(x_{j_n}))$ such that $j \notin \{j_1, \ldots, j_n\}$. Since $\mathcal{T}$ is constructed according to σ, we obtain that there is a transition $(p, f, p_1, \ldots, p_i) \in \Delta_{\mathcal{B}}$ and furthermore $T(\mathcal{B}_{p_j}) = T_\Sigma$. Consequently, $t|_{v'j} \in T(\mathcal{B}_{p_j})$, and thus also $(\rho(v), \mathrm{val}_t(v))$ is a final combination. Hence, $t \in dom(R)$.

Secondly, we show that $t \in dom(R)$ implies $(t, \mathcal{T}(t)) \in R$. We show by induction on the number of steps n needed to reach a configuration $c = (t, t', \varphi)$ with $D_{t'} \neq \emptyset$ that for each $u \in D_{t'}$ holds:

(a) Let $\rho_{\mathcal{B}}$ be the run of $\mathcal{B}$ on t and $\rho_{\mathcal{A}}$ be the run of $\mathcal{A}$ on $t \otimes t'$. If $\mathrm{val}_{t'}(u) = (p, q)$, then $\rho_{\mathcal{B}}(u) = p$ and $\rho_{\mathcal{A}}(u) = q$. If $\mathrm{val}_{t'}(u) = (p, P)$ with $P \in \in^{Q_C}$, then for some $(p', q) \in P$ holds $\rho_{\mathcal{A}}(u) = q$ and if $u \in \mathrm{dom}_t$, then $\rho_{\mathcal{B}}(u) = p'$ otherwise $p' = p_\perp$.

(b) Let $\rho_{\mathcal{A}}(u) = q$. There exists $c' = (t, t'', \varphi')$ with $c \to_{\mathcal{T}} c'$ and $u \notin D_{t''}$ such that $\big(q, (\mathrm{val}_t(u), \mathrm{val}_{t''}(u)), q_1, \ldots, q_n\big) \in \Delta_{\mathcal{A}}, q_1, \ldots, q_n \in Q_{\mathcal{A}}$ and for $(t \otimes t'')|_{ui}$ holds if $(t \otimes t'')|_{ui} \in T_{\Sigma_\perp \cup \Gamma_\perp}$, then $(t \otimes t'')|_{ui} \in T(\mathcal{A}_{q_i})$ for each i, $1 \leq i \leq n$.

For the induction base, we consider the initial configuration (t, t_0, φ_0) with $t_0 = (q_0^{\mathcal{B}}, q_0^{\mathcal{A}})$ and $D_{t_0} = \{\varepsilon\}$. For part (a), it holds that $\text{val}_{t_0}(\varepsilon) = (q_0^{\mathcal{B}}, q_0^{\mathcal{A}})$, and each run of $\mathcal{A}$ starts in $q_0^{\mathcal{A}}$, each run of $\mathcal{B}$ starts in $q_0^{\mathcal{B}}$, i.e., $\rho_{\mathcal{A}}(\varepsilon) = q_0^{\mathcal{A}}$ and $\rho_{\mathcal{B}}(\varepsilon) = q_0^{\mathcal{B}}$. For part (b), since each play starts in $\big((q_0^{\mathcal{B}}, q_0^{\mathcal{A}})\big)$ and σ is a winning strategy for Out, the vertex $\big((q_0^{\mathcal{B}}, q_0^{\mathcal{A}}), \text{val}_t(\varepsilon)\big)$ is reachable and Out can make her next move by taking some g-edge according to σ. Let $\text{val}_t(\varepsilon) = f \in \Sigma_i$, $g \in \Gamma_j$ with $i, j \geq 0$. Thus, by construction of $\mathcal{T}$, there is a transition $(q_0^{\mathcal{B}}, q_0^{\mathcal{A}})\big(\text{val}_t(\varepsilon)(x_1, \ldots, x_i)\big) \to g((p_{j_1}, P_1)(x_{j_1}), \ldots, (p_{j_j}, P_j)(x_{j_j})) \in \Delta$ or a transition $(q_0^{\mathcal{B}}, q_0^{\mathcal{A}})\big(\text{val}_t(\varepsilon)\big) \to g(t'_1, \ldots, t'_j) \in \Delta$, $t'_1, \ldots, t'_j \in T_\Gamma$, and consequently there exists a (unique) successor configuration (t, t_1, φ_1) such that $(t, t_0, \varphi_0) \to_{\mathcal{T}} (t, t_1, \varphi_1)$ with $\text{val}_{t_1}(\varepsilon) = g$. From the fact that the g-edge exists in the game graph we know that there exists a transition of the form $\big(q_0^{\mathcal{A}}, (\text{val}_t(\varepsilon), \text{val}_{t_1}(\varepsilon)), q_1, \ldots, q_n\big) \in \Delta_{\mathcal{A}}$, $q_1, \ldots, q_n \in Q_{\mathcal{A}}$. It remains to be shown, that the parts of the output which are final, i.e., the subtrees of $t \otimes t_1$ that do not contain states as leaves, are accepted by $\mathcal{A}$. In order to show that, we distinguish two cases. First, let $f \in \Sigma_0$, then $n = j$ and $(t \otimes t_1)|_k = \bot \otimes t'_k$ for k, $1 \leq k \leq j$. We obtain $\bot \otimes t'_k \in T(\mathcal{A}_{q_k})$ for k, $1 \leq k \leq j$, because the existence of the g-edge implies that these constraints are satisfied. Secondly, let $i > j$, then $n = i$ and $(t \otimes t_1)|_k = s_k \otimes \bot$ for some $s_k \in T(\mathcal{B}_{p_k})$ for k, $j+1 \leq k \leq j$. Again, the existence of the g-edge implies that $s_k \otimes \bot \in T(\mathcal{A}_{q_k})$ for k, $j+1 \leq k \leq j$. Hence, the claim holds.

For the induction step, let $c_n = (t, t_n, \varphi_n)$, $c_{n+1} = (t, t_{n+1}, \varphi_{n+1})$ be configurations such that $(t, t_0, \varphi_0) \to_{\mathcal{T}}^n (t, t_n, \varphi_n) \to_{\mathcal{T}} (t, t_{n+1}, \varphi_{n+1})$ with $D_{t_{n+1}} \neq \emptyset$. Assume the claim holds for n. Consider any $u \in D_{t_{n+1}}$. In the case that $u \in D_{t_n}$, it follows directly from the induction hypothesis that the claim holds. So, assume $u \notin D_{t_n}$. In this case the configuration c_{n+1} was reached by producing output at the predecessor u' of u. That is, there exists $u' \in D_{t_n}$, $u' \notin D_{t_{n+1}}$ with $\text{val}_{t_n}(u') = (p, P)$, $\text{val}_t(\varphi_n(u')) = f \in \Sigma_i$, $i > 0$ and $(p, P)(f(x_1, \ldots, x_i)) \to g((p_{j_1}, P_1)(x_{j_1}), \ldots, (p_{j_j}, P_j)(x_{j_j})) \in \Delta$ and c_{n+1} was reached by applying this transition producing output at u'. Furthermore, we know that in a play according to σ the vertex $\big((p, P), f\big)$ is reachable and the next move leads to $\big((p_{j_1}, P_1), \ldots, (p_{j_j}, P_j)\big)$ via an g-edge. Let u be the kth successor of u', then $\text{val}_{t_{n+1}}(u) = (p_{j_k}, P_k)$ and $\varphi_{n+1}(u) = v$ such that $v = v' j_k$ for $\varphi_{t_n}(u') = v'$. To prove part (a) of the claim, we distinguish two cases:

(i) $P = q \in Q_{\mathcal{A}}$:

By induction hypothesis we know that $\rho_{\mathcal{B}}(u') = p$ and $\rho_{\mathcal{A}}(u') = q$. Since σ defines the next move of Out from $\big((p, q), f\big)$ with output g, there exist transitions $(p, f, p_1, \ldots, p_i) \in \Delta_{\mathcal{B}}$ and $(q, (f, g), q_1, \ldots, q_n) \in \Delta_{\mathcal{A}}$. Hence, $\rho_{\mathcal{A}}(u) = q_k$. By construction of P_k it holds that $P_k = q_k$ if $j_k = k$, or if $j_k \neq k, j_k \leq rk(g)$ it holds $P_k = \{(p_k, q_k)\}$. In both cases, $\rho_{\mathcal{B}}(u) = p_k$ and $\rho_{\mathcal{A}}(u) = q_k$. If $j_k > rk(g)$, it holds $P_k = \{(p_\bot, q_k)\}$, $\rho_{\mathcal{A}}(u) = q_k$ and $\rho_{\mathcal{B}}(u)$ is not defined.

(ii) $P \in 2^{Q_{\mathcal{C}}}$:

By induction hypothesis we know that $\rho_{\mathcal{B}}(u') = p'$ and $\rho_{\mathcal{A}}(u') = q$ for some

$(p', q) \in P$, and also $\big(q, (\mathrm{val}_t(u'), \mathrm{val}_{t_{n+1}}(u')), q_1, \ldots, q_n\big) \in \Delta_{\mathcal{A}}$. Therefore, the run of $\mathcal{A}$ on $t \otimes \mathcal{T}(t)$ results in $\rho_{\mathcal{A}}(u) = q_k$. If $u' \notin \mathrm{dom}_t$, then $u \notin \mathrm{dom}_t$ and $p' = p_\perp$, otherwise if $u' \in \mathrm{dom}_t$, then $p' \in Q_{\mathcal{B}}$ and there exists $(p', \mathrm{val}_t(u'), p_1, \ldots, p_{rk(\mathrm{val}_t(u'))}) \in \Delta_{\mathcal{B}}$, because $t \in dom(R)$. Hence, if $u \in \mathrm{dom}_t$ the run of $\mathcal{B}$ on t results in $\rho_{\mathcal{B}}(u) = p_k$. Since there exists an g-edge from $\big((p, P), f\big)$ to $\big((p_{j_1}, P_1), \ldots, (p_{j_j}, P_j)\big)$, we can conclude that P_k contains $(p_\perp, q_k)$ respectively (p_k, q_k).

Concerning part (b) of the claim, from $\big((p_{j_1}, P_1), \ldots, (p_{j_j}, P_j)\big)$ In can move to $\big((p_{j_k}, P_k), \mathrm{val}_t(v)\big)$. Since σ is a winning strategy for Out, she moves to the next vertex via some g-edge. Consequently, there exists a corresponding transition in $\mathcal{T}$, and thus also a successor configuration $c_{n+2} = (t, t_{n+2}, \varphi_{n+2})$ of c_{n+1} such that $\mathrm{val}_{t_{n+2}}(u) = g$. Again, we distinguish two cases.

(i) $P_k = q_k \in Q_{\mathcal{A}}$:

In this case we have $u = v$, then the existence of a g-edge implies that there exists $\big(q_k, (\mathrm{val}_t(u), \mathrm{val}_{t_{n+2}}(u)), q_{k1}, \ldots, q_{kn}\big) \in \Delta_{\mathcal{A}}$, $(p_k, \mathrm{val}_t(u), p_{k1}, \ldots, p_{ki}) \in \Delta_{\mathcal{B}}$ and $\rho_{\mathcal{A}}(ul) = q_{kl}$ for l, $1 \leq l \leq n$. It holds either $\mathrm{val}_t(v) \in \Sigma_0$ or $\mathrm{val}_t(v) \in \Sigma_i$, $i > 0$. In the first case, $(t \otimes t_{n+2})|_{ul} = \perp \otimes t'_l$ for l, $1 \leq l \leq n$ and $t'_1, \ldots, t'_n$ satisfy the constraints of the g-edge. It follows directly that $(t \otimes t_{n+2})|_{ul} \in T(\mathcal{A}_{q_{kl}})$ for l, $1 \leq l \leq n$. In the latter case, assume $g \in \Gamma_j$ with $j < i$, then $(t \otimes t_{n+2})|_{ul} = s_l \otimes \perp$ for some $s_l \in T(\mathcal{B}_{p_{kl}})$ for l, $1 \leq l \leq j$. Again, the possibility of the move yields that $(t \otimes t_{n+2})|_{ul} \in T(\mathcal{A}_{q_{kl}})$ for l, $1 \leq l \leq j$.

(ii) $P_k \in 2^{Q_C}$:

From above, we have $\rho_{\mathcal{A}}(u) = q_k$, $\rho_{\mathcal{B}}(u) = p_k$, and $(p_k, q_k) \in P_k$. Since the g-edge exists, by construction of the game graph, it holds that for each p, each $f \in \Sigma$ if there exists a transition with left-hand side $(p, f) \in \Delta_{\mathcal{B}}$ or for $p = p_\perp$ and $\perp$ such that $(p, q) \in P_k$, there exists a transition with left-hand side $(q, (f, g))$ respectively $(q, (\perp, g) \in \Delta_{\mathcal{A}}$. Hence, a successor configuration c_2 is reachable and there exists a transition of the form $\big(q_k, (\mathrm{val}_t(u), \mathrm{val}_{t_{n+2}}(u)), q_{k1}, \ldots, q_{kn}\big) \in \Delta_{\mathcal{A}}$. As in the above case, it follows with the same argumentation that each $(t \otimes t_{n+2})|_{ul}$ with $1 \leq l \leq n$ such that $(t \otimes t_{n+2})|_{ul} \in T_{\Gamma_\perp}$ is accepted by $\mathcal{A}_{q_{kl}}$, because the constraints hold for each $(q, p) \in P_k$.

In both cases the claim holds.

Altogether, we obtain for $t \in dom(R)$ that $t \to^*_{\mathcal{T}} t' \in T_\Gamma$ and $(t, t') \in R$. Obviously, $\mathcal{T}$ is deterministic, thus $\mathcal{T}$ defines a uniformization of R.

□

We now show that the other direction holds as well.

Lemma 4.26 *If there exists a TDT $\mathcal{T}$ with synchronous input and output that uniformizes R, then* Out *has a winning strategy in the safety game $\mathcal{G}_{\mathcal{A},\mathcal{B}}$.*

Proof. Assume R is uniformized by a TDT $\mathcal{T}$ with synchronous input and output. A winning strategy for Out can mimic $\mathcal{T}$ by choosing her moves corresponding to the applied transitions of $\mathcal{T}$.

After n moves of In, let $f_1 \dots f_n \in \Sigma^*$ be the induced sequence of input labels and $i_1 \dots i_{n-1} \in \mathrm{dir}_\Sigma$ be the induced path. Let $g_1 \dots g_n \in \Gamma^*$ be the sequence of output labels on the path $u_1 \dots u_{n-1} \in \mathrm{dir}_\Gamma$ that was produced by $\mathcal{T}$ reading the labeled path $f_1 i_1 \dots f_n$ in some valid input tree $t \in T_\Sigma^{f_1 i_1 \dots f_n}$.

We show by induction on the number of played moves by In that after n moves of In for a reached vertex $((p, P), f_n)$ of Out holds that there is a configuration $c_n = (t, t', \varphi_n)$ of $\mathcal{T}$ on some $t \in T_\Sigma^{f_1 i_1 \dots f_n} \cap dom(R)$ such that there exists $u_1 \dots u_{n-1} \in \mathrm{dom}_{t'}$ with $\varphi_n(u_1 \dots u_{n-1}) = i_1 \dots i_{n-1}$ and $s(f_n(x_1, \dots, x_i)) \to g_n(s_1(x_{j_1}), \dots, s_k(x_{j_k})) \in \Delta_\mathcal{T}$ for $\mathrm{val}_{t'}(u_1 \dots u_{n-1}) = s$ such that

(i) if $P = q \in Q_\mathcal{A}$, then $u_1 \dots u_{n-1} = i_1 \dots i_{n-1}$ with

$$\mathcal{C} : q_0^\mathcal{C} \xrightarrow{f_1 \dots f_{n-1} \otimes g_1 \dots g_{n-1}}_{i_{n-1}} (p, q)$$

and $((p, q), (f_n, g_n), (p_1, q_1), \dots, (p_m, q_m)) \in \Delta_\mathcal{C}$.

(ii) if $P \in 2^{Q_\mathcal{C}}$, then for each $(p', q') \in P$ and each $(p', f_n', p_1', \dots, p_i') \in \Delta_\mathcal{B}^\perp$ the tree t can be chosen with $\mathrm{val}_t^\perp(\varepsilon) = f_1'$, $\mathrm{val}_t^\perp(u_i) = f_{i+1}'$ for $1 \leq i < n$ such that

$$\mathcal{C} : q_0^\mathcal{C} \xrightarrow{f_1' \dots f_{n-1}' \otimes g_1 \dots g_{n-1}}_{u_{n-1}} (p', q')$$

and $\big((p', q'), (f_n', g_n), (p_1', q_1'), \dots, (p_m', q_m')\big) \in \Delta_\mathcal{C}$.

Note that since $\mathcal{T}$ uniformizes R, if the claim is true, then this implies that from this vertex an outgoing edge exists. Then Out can make the move corresponding to the transition applied in the next computation step of $\mathcal{T}$ and wins, i.e., Out can take the g_n-labeled edge from $\big((p, P), f\big)$ to $\big((p_{j_1}, P_1), \dots, (p_{j_k}, P_k)\big)$.

For the induction base, consider the first reached vertex of Out of the form $\big((q_0^\mathcal{B}, q_0^\mathcal{A}), f_1\big)$. In the initial configuration $c_1 = (t, t', \varphi_1)$ of $\mathcal{T}$ on each tree $t \in T_\Sigma^f$ it holds that $\varphi_1(\varepsilon) = \varepsilon$. Since $\mathcal{T}$ uniformizes R, there exists $s(f_1(x_1, \dots, x_i)) \to g_1(s_1(x_{j_1}), \dots, s_k(x_{j_k})) \in \Delta_\mathcal{T}$ such that s is the initial state of $\mathcal{T}$. It follows directly that the claim holds.

For the induction step, assume that the claim holds for n. Let $((p, P), f_n)$ be the nth reached vertex of Out in a play, $\big((p_{j_1}, P_1), \dots, (p_{j_k}, P_k)\big)$ be the subsequent reached vertex of In, and $((p', P'), f_{n+1})$ the $(n+1)$th reached vertex of Out. We distinguish two cases.

(i) $P' = q' \in Q_\mathcal{A}$, then $((p', q'), f_{n+1})$ was reached from $((p, q), f_n)$ with $P = q$ for some $q \in Q_\mathcal{A}$.

(ii) $P' \in 2^{Q_\mathcal{C}}$, then $((p', P'), f_{n+1})$ was reached from

  (a) $((p, P), f_n)$ with $P \in 2^{Q_\mathcal{C}}$, or

  (b) $((p, q), f_n)$ with $P = q$ for some $q \in Q_\mathcal{A}$.

In both cases, it follows from the induction hypothesis that there exists a configuration $c_n = (t, t', \varphi_n)$ of $\mathcal{T}$ with $t \in T_\Sigma^{f_1 i_1 \dots f_n} \cap dom(R)$ such that there exists $u_1 \dots u_{n-1} \in \mathrm{dom}_{t'}$ with $\varphi_n(i_1 \dots i_{n-1}) = u_1 \dots u_{n-1}$, $\mathrm{val}_{t'}(u_1 \dots u_{n-1}) = s$, $\mathrm{val}_t(i_1 \dots i_{n-1}) = f_n$ and $s(f_n(x_1, \dots, x_i)) \to g_n(s_1(x_{j_1}), \dots, s_k(x_{j_k})) \in \Delta_{\mathcal{T}}$. Out takes the edge corresponding to this transition.

In case (i), the induction hypothesis yields that $\mathcal{C} : q_0^{\mathcal{C}} \xrightarrow{f_1 \dots f_{n-1} \otimes g_1 \dots g_{n-1}}_{i_{n-1}} (p, q)$ and $\big((p, q), (f_n, g_n), (p_1, q_1), \dots, (p_m, q_m)\big) \in \Delta_{\mathcal{C}}$. We obtain $p' = p_{i_n}$ and $q' = q_{i_n}$. Consequently, $\mathcal{C} : (p, q) \xrightarrow{f_n \otimes g_n}_{i_n} (p', q')$. It is easily seen that if additionally $\mathrm{val}_t(i_1 \dots i_n) = f_{n+1}$, then a configuration c_{n+1} with $c_n \to_{\mathcal{T}} c_{n+1}$ is reachable that satisfies the claim.

For part (a) of case (ii), let In move from $\big((p_{j_1}, P_1), \dots, (p_{j_k}, P_k)\big)$ to the jth element of this list together with input f_{n+1}. Then, $i_n = j_j$, $u_n = j$, and consequently $p' = p_{j_j}$ and $P' = P_j$. Consider a pair $(p'_j, q'_j) \in P_j$, by construction $(p'_j, q'_j) \in P_j$ if there is some $(p', q') \in P$ and $\big((p', q'), (f'_n, g_n), (p'_1, q'_1), \dots, (p'_m, q'_m)\big) \in \Delta_C$ for some $f'_n \in \Sigma_\perp$. By induction hypothesis and the existence of the transition, we can assume that t is chosen such that $\mathcal{C} : q_0^{\mathcal{C}} \xrightarrow{f'_1 \dots f'_{n-1} \otimes g_1 \dots g_{n-1}}_{u_{n-1}} (p', q') \xrightarrow{f'_n \otimes g_n}_{u_n} (p'_j, q'_j)$. Let $c_{n+1} = (t, t'', \varphi_{n+1})$ be a successor configuration of c_n that results from applying $s(f_n(x_1, \dots, x_i)) \to g_n(s_1(x_{j_1}), \dots, s_k(x_{j_k}))$ thereby producing output g_n at $u_1 \dots u_{n-1}$. We obtain $\varphi_{n+1}(u_1 \dots u_n) = i_1 \dots i_n$ with $\mathrm{val}_t(i_1 \dots i_n) = f_{n+1}$ and $\mathrm{val}_{t''}(u_1 \dots u_n) = s_j$. Since $\mathcal{T}$ uniformizes R, there has to exist a transition $s_j(f_{n+1}(x_1, \dots, x_i)) \to g_{n+1}(s'_1(x_{j_1}), \dots, s'_k(x_{j_k})) \in \Delta_{\mathcal{T}}$. For each $(p'_j, f'_{n+1}, p'^1_j, \dots, p'^i_j) \in \Delta_{\mathcal{B}}^\perp$, assume t is chosen such that $\mathrm{val}_t^\perp(u_1 \dots u_n) = f'_{n+1}$. In the next computation step of $\mathcal{T}$, the output g_{n+1} at $u_1 \dots u_n$ is produced independently of $\mathrm{val}_t^\perp(u_1 \dots u_n)$, thus there exists $\big((p'_j, q'_j), (f'_{n+1}, g_{n+1}), (p'^1_j, q'^1_j), \dots, (p'^m_j, q'^m_j)\big) \in \Delta_{\mathcal{C}}$ for each $(p'_j, f'_{n+1}, p'^1_j, \dots, p'^i_j) \in \Delta_{\mathcal{B}}^\perp$. Hence, the claim holds for $n + 1$.

We skip part (b) of case (ii), because $((p, q), f_n)$ corresponds to $(p, \{(p, q)\}, f_n)$ such that only $(p, f_n, p_1, \dots, p_i) \in \Delta_{\mathcal{B}}$ has to be considered.

□

From Lemmata 4.25 and 4.26 together with the fact that a winning strategy for Out can effectively be computed in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$ we obtain the following result.

Theorem 6 *It is decidable whether a D↓TA-recognizable relation with D↓TA-recognizable domain has a uniformization by a top-down tree transducer with synchronous input and output.*

We conclude the chapter by providing an example of a relation where it is necessary for a transducer which uniformizes the relation, that input and output are asynchronous.

Example 4.27 Let Σ be a ranked alphabet given by $\Sigma_2 = \{f\}$ and $\Sigma_0 = \{a, b\}$ and Γ be a ranked alphabet given by $\Gamma_2 = \{f\}$, $\Gamma_1 = \{g\}$ and $\Gamma_0 = \{a, b\}$. Consider the following relation $R_2 \subseteq T_\Sigma \times T_\Gamma$ given by $\{\big(f(b, t), g(f(t', b))\big) \mid \neg\exists u \in \mathrm{dom}_t : \mathrm{val}_t(u) = b\}$.

The relation R_2 is very similar to the relation R_1 from Example 4.23. Thus, it is easy to see that R_2 and $dom(R_2)$ are D↓TA-recognizable. Also, a TDT that uniformizes R_2 can be obtained from the TDT $\mathcal{T}$ from Example 4.23 by replacing the transition $q_0(f(x_1, x_2)) \to f(q_1(x_2), q_2(x_1))$ with $q_0(f(x_1, x_2)) \to g(f(q_1(x_2), q_2(x_1)))$.

However, postponing the output would result in a transition $q_0(f(x_1, x_2)) \to g(q'(x_1))$ or $q_0(f(x_1, x_2)) \to g(q'(x_2))$. For the same reasons as presented in Example 4.23, it is not possible to pursue only one path, because then the input tree can not completely be verified and the transducer would transform invalid input trees.

We have seen in the above example that we cannot alter a transducer that defines a uniformization such that the output is postponed as described in Remark 4.12 if the input tree has to be verified. Let us recall the definition of $\mathcal{G}_{\mathcal{A},\mathcal{B}}$, we see that the underlying structure of the game graph does not allow for more than one produced output symbol at the same time.

Chapter 5

Uniformization by Bottom-Up Tree Transducers

While in the previous chapter we considered uniformization in the class of top-town tree transformations, we now consider the question whether a tree-automatic relation has a uniformization by a bottom-up transducer. We only explore this topic briefly by focusing on a very restricted setting similar to the case presented at the beginning of the previous chapter.

5.1 Bottom-Up Tree Transducer

Bottom-up tree transducers transform an input tree from the leafs to the root. Unlike top-down tree transducers, in the bottom-up case it is possible for a transducer to discard already produced output. For an introduction to bottom-up tree transducers we mention [CDG+07].

Definition 5.1 (BTT). A *bottom-up tree transducer* is of the form $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, F)$ consisting of a finite set of states Q, a finite input alphabet Σ, a finite output alphabet Γ, a set F of final states, and Δ is a finite set of transition rules of the form

$$f(q_1(x_1), \ldots, q_i(x_i)) \to q(u),$$

where $f \in \Sigma_i$, $u \in T_\Gamma(X_i)$, and $q, q_1, \ldots, q_n \in Q$, or

$$q(x_1) \to q'(u) \quad (\varepsilon\text{-transition}),$$

where $u \in T_\Gamma(X_1)$ and $q, q' \in Q$.

A *configuration* of a bottom-up tree transducer is a tree over the ranked alphabet $\Sigma \cup \Gamma \cup Q$, where Q is used as a set of unary symbols.

Let $t, t' \in T_{\Sigma \cup \Gamma \cup Q}$ be configurations of a bottom-up tree transducer. We define a successor relation $\to_{\mathcal{T}}$ on configurations by:

$$t \to_{\mathcal{T}} t' :\Leftrightarrow \begin{cases} \exists f(q_1(x_1), \ldots, q_n(x_n))) \to q(u) \in \Delta \\ \exists s \in S_{\Sigma \cup \Gamma \cup Q} \\ \exists t_1, \ldots, t_i \in T_\Gamma \\ t = f(q_1(t_1), \ldots, q_i(t_i)) \cdot s \\ t' = q(u[x_1 \to t_1, \ldots, x_i \to t_i]) \cdot s \end{cases}$$

Furthermore, let $\to_{\mathcal{T}}^*$ be the reflexive and transitive closure of $\to_{\mathcal{T}}$ and $\to_{\mathcal{T}}^n$ the reachability relation for $\to_{\mathcal{T}}$ in n steps.

A bottom-up tree transducer is *deterministic* (a DBTT) if it contains no ε-transition and there are no two rules with the same left-hand side.

Definition 5.2 (Semantics of BTTs). The relation $R(\mathcal{T}) \subseteq T_\Sigma \times T_\Gamma$ induced by a bottom-up tree transducer $\mathcal{T}$ is

$$R(\mathcal{T}) = \{(t, t') \mid t \to_{\mathcal{T}}^* q(t') \text{ with } t \in T_\Sigma, t' \in T_\Gamma \text{ and } q \in F\}.$$

For a tree $t \in T_\Sigma$ let $\mathcal{T}(t) := \{t' \in T_\Gamma \mid (t, t') \in R(\mathcal{T})\}$.

The class of relations definable by BTTs is called the class of *bottom-up tree transformations.*

Example 5.3 Let Σ be a ranked alphabet given by $\Sigma_2 = \{f\}$, $\Sigma_1 = \{g, h\}$, and $\Sigma_0 = \{a\}$. Consider the BTT $\mathcal{T}$ given by $(\{q\}, \Sigma, \Sigma, \Delta, \{q\})$ with $\Delta =$

$$\begin{array}{lrcll} \{ & a & \to & q(a), & \\ & a & \to & q(f(a,a)), & \\ & g(q(x_1)) & \to & q(g(x_1)), & \\ & h(q(x_1)) & \to & q(h(x_1)), & \\ & f(q(x_1), q(x_2)) & \to & q(f(x_1, x_2)) & \}. \end{array}$$

For each $t \in T_\Sigma$ the transducer can non-deterministically replace all occurrences of a in t by $f(a, a)$.

5.2 A Restricted Uniformization Case

In this section we only regard tree-automatic relations such that for each pair (t, t') that is part of the relation holds $\text{dom}_t = \text{dom}_{t'}$. A BTT that implements a uniformization of such a relation should do so by relabeling the nodes in an input tree. Formally, the corresponding uniformization problem is defined as follows.

Definition 5.4 Let R be a tree-automatic relation such that for each $(t, t') \in R$ holds that $\text{dom}_t = \text{dom}_{t'}$. The *restricted uniformization problem* is the decision problem whether there exists a uniformization of R whose graph is recognizable by a deterministic BTT $\mathcal{T}$ such that only transitions of the following form are used:

$$f(q_1(x_1), \ldots, q_i(x_i)) \to q(g(x_1, \ldots, x_i)),$$

where $f \in \Sigma_i$, $g \in \Gamma_i$, and $q, q_1, \ldots, q_i \in Q$.

For the rest of this chapter we fix a tree-automatic relation $R \subseteq T_\Sigma \times T_\Gamma$ with $\Sigma = \bigcup_{i=0}^{m} \Sigma_i$ such that $\mathrm{dom}_t = \mathrm{dom}_{t'}$ for each $(t, t') \in R$. Let a DTA $\mathcal{A} = (Q_\mathcal{A}, \Sigma \times \Gamma, \Delta_\mathcal{A}, F_\mathcal{A})$ recognize R and let a DTA $\mathcal{B} = (Q_\mathcal{B}, \Sigma, \Delta_\mathcal{B}, F_\mathcal{B})$ recognize $dom(R)$.

We provide a decision procedure for the problem mentioned above that reduces the question whether there exists such a uniformization to the question of the existence of winning strategies in a safety game between In and Out.

Similar to the previously considered games, the vertices of the game graph keep track of the state of $\mathcal{B}$ on the input and the state of $\mathcal{A}$ on the combination of input and output. However, since the input tree is read from bottom to top, the vertices also keep track of the previously reached combinations of states of $\mathcal{A}$ and $\mathcal{B}$. Player In can combine an input symbol with states of $\mathcal{B}$ from the set of reached states such that a valid left-hand side of a transition in $\Delta_\mathcal{B}$ is formed. If possible, Out can play an output symbol such that the (to the states of $\mathcal{B}$) associated states of $\mathcal{A}$ and the pair of input symbol and chosen output symbol form a valid left-hand side of a transition in $\Delta_\mathcal{A}$.

With these properties in mind, the game graph $G_{\mathcal{A},\mathcal{B}}$ is built as follows:

- $V_{\mathsf{In}} \subseteq 2^{Q_\mathcal{A} \times Q_\mathcal{B}}$ is the set of vertices of player In,
- $V_{\mathsf{Out}} \subseteq (\bigcup_{i=0}^{m} (Q_\mathcal{A} \times Q_\mathcal{B})^i \times \Sigma_i) \times V_{\mathsf{In}}$ is the set of vertices of player Out.
- From a vertex of In the following moves are possible:
  - $P \to \big[\big((q_1, p_1), \ldots, (q_i, p_i), f\big), P\big]$ if $f \in \Sigma_i$ and $(q_1, p_1), \ldots, (q_i, p_i) \in P$ such that there exists $(p_1, \ldots, p_i, f, p) \in \Delta_\mathcal{B}$
- From a vertex of Out the following moves are possible:
  - $\big[\big((q_1, p_1), \ldots, (q_i, p_i), f\big), P\big] \xrightarrow{r} \{(q, p)\} \cup P$ if there exists a transition $r = (q_1, \ldots, q_i, (f, g), q) \in \Delta_\mathcal{A}$ and a transition $(p_1, \ldots, p_i, f, p) \in \Delta_\mathcal{B}$
- The initial vertex is $\emptyset$.

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 all $P \in V_{\mathsf{In}}$ such that there is $(q, p) \in P$ with $p \in F_\mathcal{B}$, but $q \notin F_\mathcal{A}$ and by all vertices in V_{Out} that have no outgoing edges. Let B denote this set. If one of these bad vertices is reached during a play, Out must lose the game. Therefore, we define $\mathcal{G}_{\mathcal{A},\mathcal{B}} = (G_{\mathcal{A},\mathcal{B}}, V \setminus B)$ as safety game for Out.

Intuitively, since the set of reachable states only grows in a play, it is not necessary in a winning strategy for Out that Out choses a different output symbol in case that a left-hand side of a transition is visited again (together with a grown set of reachable states). Before we turn to the proof that from a winning strategy in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$ a BTT that uniformizes R can be constructed, we formalize the above intuition.

Remark 5.5 *Given a positional winning strategy σ for* Out *in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$. Consider two vertices in V_{Out} such that*

$$\sigma : \big[\big((q_1, p_1), \dots, (q_i, p_i), f\big), P\big] \overset{r}{\mapsto} \{(q, p)\} \cup P$$

and

$$\sigma : \big[\big((q_1, p_1), \dots, (q_i, p_i), f\big), P'\big] \overset{r'}{\mapsto} \{(q', p')\} \cup P' \text{ with } P \subseteq P'.$$

Then redefining σ such that $\sigma : \big[\big((q_1, p_1), \dots, (q_i, p_i), f\big), P'\big] \overset{r}{\mapsto} \{(q, p)\} \cup P'$ also yields a winning strategy for Out*.*

Proof. In each play on $G_{\mathcal{A},\mathcal{B}}$, when a vertex $P \in V_{\mathsf{In}}$ is reached, each vertex that is visited afterwards in the play is either of the form $P' \in V_{\mathsf{In}}$ or $[(\dots), P'] \in V_{\mathsf{Out}}$ with $P \subseteq P'$ by construction of the game graph. That means $x := \big[\big((q_1, p_1), \dots, (q_i, p_i), f\big), P\big]$ is visited before $y := \big[\big((q_1, p_1), \dots, (q_i, p_i), f\big), P'\big]$ in a play played according to σ. When the play is in x, the next moves leads to $\{(q, p)\} \cup P$. If eventually y is reached, it is also possible for Out to take the r-edge instead of the r'-edge specified by σ. The r-edge has to exist, because the constraints to add a r-edge outgoing from y in the game graph are also fulfilled. Choosing the r-edge from this vertex leads to $\{(q, p)\} \cup P'$. It already holds that $\{(q, p)\} \in P'$, thus the move leads again to P'. Since σ is a winning strategy and P' was visited before, this also defines a winning strategy for Out.

□

The following lemma shows us that we can construct a bottom-up tree transducer that uniformizes R from a winning strategy of Out in the game and vice versa.

Lemma 5.6 *The relation R can be uniformized by a restricted BTT in the sense as defined in Definition 5.4 if, and only if,* Out *has a winning strategy in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$.*

Proof. Assume that Out has a winning strategy in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$. Then there also exists a positional one. We can represent a positional winning strategy by a function $\sigma : V_{\mathsf{Out}} \to \Delta_{\mathcal{A}}$. More conveniently, we usually also indicate the reached vertex of In.

With Remark 5.5 in mind, we construct a deterministic BTT $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, F)$ from σ as follows:

- $Q := Q_{\mathcal{A}} \times Q_{\mathcal{B}}$ is the set of states, and
- $F := \{(q, p) \in Q \mid q \in F_{\mathcal{A}}, p \in F_{\mathcal{B}}\}$ is the set of final states, and
- Δ is build up in the following way:
  1. Start at $x := \emptyset \in V_{\mathsf{In}}$
  2. From $x \in V_{\mathsf{In}}$ move to $y := \big[\big((q_1, p_1), \dots, (q_i, p_i), f\big), P\big] \in V_{\mathsf{Out}}$ such that no $y' := \big[\big((q_1, p_1), \dots, (q_i, p_i), f\big), P'\big]$ with $P \subseteq P'$ was visited before:

(a) if $\sigma(y) = \{(q,p)\} \cup P$ via $r = (q_1, \dots, q_i, (f,g), q) \in \Delta_\mathcal{A}$ add $f\big((q_1,p_1)(x_1), \dots, (q_i,p_i)(x_i)\big) \to (q,p)\,(g(x_1,\dots,x_i))$ to Δ

(b) $x := \{(q,p)\} \cup P$

3. Continue with step 2, if no new vertex can be reached, stop.

The resulting BTT $\mathcal{T}$ is deterministic and it is easy to see that $R(\mathcal{T}) \subseteq R$.

We verify that $\mathcal{T}$ defines a uniformization of R. Let $t \in T_\Sigma$ and ρ is the run of $\mathcal{B}$ on t. We show by induction on the height of t that there exists a tree $t' \in T_\Gamma$ such that $t \to^*_\mathcal{T} (q,p)(t')$ if $\rho(\varepsilon) = p$.

For the induction base, we have $t = a \in \Sigma_0$. Let $\rho(\varepsilon) = p$. Each play starts in $\emptyset$, thus In can reach the vertex $[a, \emptyset]$. The winning strategy σ defines the next move to $\{(q,p)\}$ via some r-edge with $r = ((a,b),q)$. Hence, the transition $a \to (p,q)(b)$ is added to Δ and we obtain $a \to^*_\mathcal{T} (q,p)(b)$.

For the induction step, consider $t = f(t_1, \dots, t_i)$ and let $\rho(\varepsilon) = p$ and $\rho(j) = p_j$ for $1 \leq j \leq i$. By induction hypothesis, we have $t_j \to^*_\mathcal{T} (q_j,p_j)(t'_j)$ for $1 \leq j \leq i$. Therefore, in a play played according to σ, a vertex $P \in V_{\mathsf{In}}$ can be reached such that $\bigcup_{j=1}^{i}(q_j,p_j) \subseteq P$. From this vertex In can move to $\big[\big((q_1,p_1), \dots, (q_i,p_i), f\big), P\big]$. Next, Out moves to some $\{(q,p)\} \cup P$ via an r-edge with $r = (q_1, \dots, q_i, (f,g), q) \in \Delta_\mathcal{A}$. It follows that there exists a corresponding transition in Δ and we can conclude $f(t_1,\dots,t_i) \to^*_\mathcal{T} (q,p)\,(g(t'_1,\dots,t'_i))$. Thus, the claim holds.

Note that only states (q,p) with $q \in F_\mathcal{A} \Leftrightarrow p \in F_\mathcal{B}$ can occur in configurations of $\mathcal{T}$, because σ is a winning strategy. Now, consider $t \in dom(R)$, then $\rho(\varepsilon) = p \in F_\mathcal{B}$. From the above induction we derive that there exists a tree $t' \in T_\Gamma$ such that $t \to^*_\mathcal{T} (q,p)(t')$ with $(q,p) \in F$, i.e., $(t,t') \in R$.

For the other direction, assume that R is uniformized by a restricted deterministic BTT $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, F)$ in the sense as defined in Definition 5.4. Our goal is to translate the outputs generated by $\mathcal{T}$ on the inputs played by In into moves of Out. We define the strategy σ of Out inductively. We show by induction on the number of moves by In that for each reached vertex $P \in V_{\mathsf{In}}$ in a play for each $(q,p) \in P$ holds there exists $t \in T_\Sigma$ such that the run $\rho_\mathcal{B}$ of $\mathcal{B}$ on t yields $\rho_\mathcal{B}(\varepsilon) = p$, and letting $t' = \mathcal{T}(t)$ the run $\rho_\mathcal{A}$ of $\mathcal{A}$ on $t \otimes t'$ yields $\rho_\mathcal{A}(\varepsilon) = q$.

For the induction base, we consider the beginning of each play which starts in $\emptyset \in V_{\mathsf{In}}$. The first move leads to $\big[(a), \emptyset\big] \in V_{\mathsf{Out}}$ for some $(a,p) \in \Delta_\mathcal{B}$. Letting $b = \mathcal{T}(a)$, we define $\sigma(\big[(a),\emptyset\big]) = \{(q,p)\}$ via r with $r = \big((a,b),q\big) \in \Delta_\mathcal{A}$, where for (q,p) the claim holds.

For the induction step, we consider a vertex $P \in V_{\mathsf{In}}$ reached after the nth move of In. Let the induction hypothesis be true for each $(q_j,p_j) \in P$ for $1 \leq j \leq n$. Thus, there exists $t_j \in T_\Sigma$ with $t_j \to^*_\mathcal{T} s_j(t'_j)$ such that $\rho_\mathcal{B}$ on t_j ends in p_j and $\rho_\mathcal{A}$ on $t_j \otimes t'_j$ ends in q_j for each (q_j,p_j) for $1 \leq i \leq n$. Subsequently, In moves to some $\big[\big((q_1,p_1), \dots, (q_i,p_i), f\big), P\big] \in V_{\mathsf{Out}}$ with $(p_1, \dots, p_i, f, p) \in \Delta_\mathcal{B}$. Since $\mathcal{T}$ uniformizes R, we know that there exists a transition $f(s_1(x_1), \dots, s_i(x_i)) \to s(g(x_1,\dots,x_i)) \in \Delta$. Since $\mathcal{A}$ recognizes R, there also has to exist a corresponding transition $(q_1, \dots, q_i, (f,g), q) \in \Delta_\mathcal{A}$. Hence, Out chooses this transitions for her next move which leads to $\{(q,p)\} \cup$

$P \in V_{\mathsf{In}}$. Thus, the $(n+1)$th move of In leads to $\{(q,p)\} \cup P$ and for (q,p) holds that the run $\rho_{\mathcal{B}}$ of $\mathcal{B}$ on t with $t = f(t_1, \dots, t_i)$ yields $\rho_{\mathcal{B}}(\varepsilon) = p$, and the run $\rho_{\mathcal{A}}$ of $\mathcal{A}$ on $t \otimes t'$ with $t' = g(t'_1, \dots, t'_i)$ yields $\rho_{\mathcal{A}}(\varepsilon) = q$. Hence the claim holds.

It remains to show that σ defines a winning strategy for Out. That is, no bad vertex can be reached during a play. In the above induction we have seen, that each reached vertex of Out has at least one outgoing edge. Furthermore, for each reached vertex of In for each $(q,p) \in P$ the run of $\mathcal{A}$ on $t \otimes \mathcal{T}(t)$ ends in q. Since $\mathcal{T}$ uniformizes R it follows that $q \in F_{\mathcal{A}} \Leftrightarrow p \in F_{\mathcal{B}}$. This shows that σ is indeed a winning strategy for Out.

□

Combining Lemma 5.6 and the fact that a winning strategy for Out can be effectively computed in $\mathcal{G}_{\mathcal{A},\mathcal{B}}$ we derive the following result.

Theorem 7 *The restricted uniformization problem in the class of bottom-up tree transformations is decidable.*

Chapter 6

Conclusion

In this thesis we focused on uniformization of tree-automatic relations in the class of tree transformations defined by top-down and bottom-up tree transducers.

In Chapter 4 we were concerned whether the question, if a given deterministic top-down tree automaton-definable relation has a uniformization by a top-down tree transducer, is decidable.

We have shown that this question is decidable under the restriction that a top-down tree transducer is not required to validate the input, meaning that a transducer implementing a uniformization can behave arbitrarily on invalid inputs. We were able to provide a decision procedure corresponding to the uniformization question in this setting.

We have seen that the presented decision procedure concerning uniformization without input validation can not 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. Although the procedure can not be transferred one-to-one, we were able to adept it to the case where a top-down tree transducer outputs one symbol for each read input symbol. Thus, we have shown that it is decidable whether a (deterministic top-down tree-automaton definable) relation has a uniformization by a top-down transducer with synchronous input and output.

In Chapter 5, concerning the uniformization of tree-automatic relations by bottom-up transducers, we have seen that it is decidable whether a relation has a uniformization by a bottom-up tree transducer which exchanges the labels of each node.

Future Research

Regarding the uniformization by top-down tree transducers we mention two directions. One direction for future research (in the setting where it is not required to verify the input) is to also allow the relation to be specified by non-deterministic top-down tree automata. The other direction is to further explore the setting where the input has to be verified.

We have only briefly explored uniformization by bottom-up tree transducers, which could be studied further to obtain more general results. In comparison to top-down tree transducers, this model is less restricted in the sense that bottom-up tree transducers can discard transformed output, thereby it is possible to check input without keeping the output produced for doing so.

Generally, one could also study uniformization questions for other models of transducers besides the models considered in this thesis. For example, top-down transducers with regular lookahead, see [Eng76], or, for further models see the bibliographic section on tree transducers in [CDG+07].

Bibliography

[BL69] J Richard Büchi and Lawrence H Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 1969. 1

[CDG+07] Hubert Comon, Max Dauchet, Rémi Gilleron, Christof Löding, Florent Jacquemard, Denis Lugiez, Sophie Tison, and Marc Tommasi. Tree Automata Techniques and Applications. Available on: http://www.grappa.univ-lille3.fr/tata, 2007. Release October, 12th 2007. 5, 8, 17, 61, 68

[CG99] Christian Choffrut and Serge Grigorieff. Uniformization of rational relations. In *Jewels are Forever*, pages 59–71. Springer, 1999. 1

[Chu62] Alonzo Church. Logic, arithmetic and automata. In *Proceedings of the international congress of mathematicians*, pages 23–35, 1962. 1

[CL07] Thomas Colcombet and Christof Löding. Transforming structures by set interpretations. *Logical Methods in Computer Science*, 3(2):1–36, 2007. 1, 13

[CL12] Arnaud Carayol and Christof Löding. Uniformization in Automata Theory. Unpublished Manuscript, 2012. 1, 13, 32, 38

[Die00] Reinhard Diestel. *Graph Theory, 2nd Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2000. 40

[Eng76] Joost Engelfriet. Top-down tree transducers with regular look-ahead. *Mathematical Systems Theory*, 10(1):289–303, 1976. 68

[Eng78] Joost Engelfriet. On tree transducers for partial functions. *Information Processing Letters*, 7(4):170–172, 1978. 1, 2, 13

[GS83] Yuri Gurevich and Saharon Shelah. Rabin's uniformization problem. *Journal of Symbolic Logic*, pages 1105–1119, 1983.

[GS84] Ferenc Gèsceg and Magnus Steinby. Tree automata. *Akademiai Kiado*, 1984. 5, 7

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games*. Number 2500 in Lecture Notes in Compter Science. Springer, 2002. 11

[HMU01] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation, 2nd Edition*. Addison-Wesley, 2001. 5

[KW11] Dietrich Kuske and Thomas Weidner. Size and computation of injective tree automatic presentations. In *Mathematical Foundations of Computer Science 2011*, pages 424–435. Springer, 2011. 1, 13

[Ram30] F.P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, 2(1):264, 1930. 40

[Sie75] Dirk Siefkes. The recursive sets in certain monadic second order fragments of arithmetic. *Archive for Mathematical Logic*, 17(1):71–80, 1975. 1

[Vir80] J. Virágh. Deterministic ascending tree automata I. *Acta Cybernetica*, 5:33–44, 1980. 7