


Regular matching problems for infinite trees

Carlos Camino and Volker Diekert 


FMI, Universität Stuttgart, Germany: lastname@fmi.uni-stuttgart.de

Besik Dundua 

Tbilisi State University and International Black Sea University, Georgia: bdundua@gmail.com

Mircea Marin 

FMI, West University of Timișoara, Romania: mircea.marin@e-uvr.ro

Géraud Sénizergues 

LaBRI, Université de Bordeaux, France: geraud.senizergues@u-bordeaux.fr

Abstract

We investigate regular matching problems with respect to substitutions. The classical reference is Conway’s 1971 textbook *Regular algebra and finite machines*. In Chapt. 6, Conway developed a factorization calculus for regular languages. Some of his results can be stated as follows. Let $L \subseteq (\Sigma \cup \mathcal{X})^*$ and $R \subseteq \Sigma^*$ be two regular languages where Σ is a set of constants and \mathcal{X} is a set of variables. Substituting every variable x by a (regular) subset $\sigma(x) \subseteq \Sigma^*$ yields a (regular) subset $\sigma(L) \subseteq \Sigma^*$. A substitution σ solves the decision problem “ $L \subseteq R$?” if $\sigma(L) \subseteq R$. Conway showed that there are only finitely many maximal solutions and that every solution is included in a maximal one. Moreover, the maximal solutions σ are effectively computable and $\sigma(x)$ is regular for all $x \in \mathcal{X}$; and “ $\exists \sigma : \sigma(L) = R$?” becomes decidable, too. We generalize this type of results to infinite trees. We define a notion of *choice function* γ which for any tree s over $\Sigma \cup \mathcal{X}$ and for each position u labeled by variable $x \in \mathcal{X}$ selects at most one tree $\gamma(u)$ in $\sigma(x)$. In this way we can define $\gamma_\infty(s)$ as the limit of a Cauchy sequence; and the union over all these $\gamma_\infty(s)$ leads to a natural notion of $\sigma(s)$. Our definition coincides with the classical IO substitutions. Hence, we write $\sigma_{\text{io}}(L)$ instead of $\sigma(L)$. Our main result is that by using choice functions we obtain a generalization of Conway’s results for infinite trees. This includes the decidability of the question “ $\exists \sigma : \sigma_{\text{io}}(L) = R$?”, although (in contrast to word languages) it is not true in general that $\sigma_{\text{io}}(L)$ is regular as soon as $\sigma(x)$ is regular for all $x \in \mathcal{X}$. As a special case of our results, for L context-free and R regular “ $\exists \sigma : \sigma_{\text{io}}(L) \subseteq R$?” is decidable, but “ $\exists \sigma : \sigma_{\text{io}}(L) = R$?” is not. Hence, decidability of “ $\exists \sigma : \sigma_{\text{io}}(L) = R$?” for regular L and R can be viewed as optimal.

2012 ACM Subject Classification Theory of computation \rightarrow Formal languages and automata theory

Keywords and phrases Regular tree languages, infinite trees, Factorization theory, IO and OI

1 Introduction

Regular matching problems using generalized sequential machines were studied first by Ginsburg and Hibbard, [12]. They showed that, on input regular languages L and R , it is decidable whether there is a generalized sequential machine which maps L onto R . They also treat several variants of this problem; and it is noticed that the results cannot be extended to context-free languages. Here we are interested in regular matching problems from the perspective of solving language equations. For a recent survey on language equations, see [19]. A particular case is as follows. Let Σ be a set of constants, \mathcal{X} be a set of variables, $L \subseteq (\Sigma \cup \mathcal{X})^*$, and $R \subseteq \Sigma^*$. A substitution $\sigma : \mathcal{X} \rightarrow 2^{\Sigma^*}$ is called *solution* of the problem “ $L \subseteq R$?” if $\sigma(L) \subseteq R$. The following facts can be derived from [6, Chapt. 6]: **1.** It is decidable whether there is a substitution $\sigma : \mathcal{X} \rightarrow 2^{\Sigma^*}$ such that $\sigma(L) \subseteq R$ (and, say, $\emptyset \neq \sigma(x)$ for some $x \in \mathcal{X}$ to make it nontrivial). **2.** Define $\sigma \leq \sigma'$ by $\sigma(x) \subseteq \sigma'(x)$ for all $x \in \mathcal{X}$. Then every solution is upper bounded by a maximal solution; and the number of maximal solutions is finite. **3.** If σ is maximal, then $\sigma(x)$ is regular for all $x \in \mathcal{X}$; and all maximal solutions are effectively computable.

Modern proofs of these facts are easy exercises using the concept of *recognizing morphisms*. Moreover, the question “ $\exists \sigma : \sigma(L) \subseteq R$?” turns out to be PSPACE-complete if L and R are given by DFAs by [17] (for $\mathcal{X} = \emptyset$). The apparently similar question “ $\exists \sigma : \sigma(L) = R$?” is EXPSPACE-complete [1]. Another significant result in this area is due to Kunc: there is a finite L such that the (unique) maximal solution of $LX = XL$ is co-r.e.-complete [18].

Since regular languages over infinite words are recognized by a congruence of finite index [4], Conway’s results generalize smoothly to infinite words. The generalization to regular tree languages is the main theme of the current paper. We begin with a finite ranked alphabet Σ of constants, a nonempty set of *holes* H of rank 0, and a set \mathcal{X} of variables. Thus, each symbol x has a rank $\text{rk}(x) \in \mathbb{N}$. Finite trees (or *terms*) are defined as usual inductively. A standard definition for infinite trees is given below. Trees can be written as $x(s_1, \dots, s_r)$ where $r = \text{rk}(x) \geq 0$ and s_i are trees. In particular, all symbols of rank 0 are terms. Unlike in the case of term rewriting systems, substitutions are applied at inner positions of a tree. That is why we need trees with holes. The set of holes H is a nonempty subset of natural numbers: $H = \{1, \dots, |H|\}$. Every hole is viewed as a symbol of rank 0. Therefore we work with a finite ranked alphabet Δ which is a disjoint union $\Delta = \Sigma \cup \mathcal{X} \cup H$. The set of trees over Δ is denoted by $T(\Delta)$ and $T_{\text{fin}}(\Delta)$ is the subset of finite trees. A *substitution* (resp. *homomorphism*) means a mapping $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H) \setminus H}$ (resp. $h : \mathcal{X} \rightarrow T(\Sigma \cup H) \setminus H$) such that for each $x \in \mathcal{X}$ with $\text{rk}(x) = r$ we have $\sigma(x) \subseteq T(\Sigma \cup [r])$ (resp. $h(x) \in T(\Sigma \cup [r])$). Here, $[r] = \{1, \dots, r\}$. Trees in $\sigma(x)$ may have holes but no variables. Moreover, we impose $\sigma(x) \cap H = \emptyset$ (resp. $h(x) \notin H$); and we extend σ to a mapping from $\Sigma \cup \mathcal{X}$ to $2^{T_{\text{fin}}(\Sigma \cup H)}$ by $\sigma(f) = \{f(1, \dots, \text{rk}(f))\}$ for $f \in \Sigma$. A substitution σ is called *regular* if $\sigma(x)$ is regular¹ for all $x \in \mathcal{X}$. If there is some $t \in \sigma(x)$ where a hole $i \in H$ appears twice, then we are in *nonlinear* or *duplication mode*. Duplications cannot appear in the special case of words, but for trees duplication makes the problem “ $L \subseteq R$?” harder, even in the restricted case of homomorphisms: Indeed, consider a term $s = x(s_1, \dots, s_r) \in T_{\text{fin}}(\Sigma \cup \mathcal{X})$ with $x \in \mathcal{X} \cup \Sigma$, and a homomorphism $h : \mathcal{X} \rightarrow T_{\text{fin}}(\Sigma \cup H)$. The definition of $h(s)$ is given by induction: $h(x(s_1, \dots, s_r)) = h(x)[i_j \leftarrow h(s_i)]$. Here, the notation $h(x)[i_j \leftarrow h(s_i)]$ means that each leaf labeled by some hole i is replaced by the tree $h(s_i)$. In duplication mode, $h(L)$ need not be regular.² This led to the *HOM-problem*. The input is a homomorphism h and the question is whether $h(L)$ is regular. The problem is decidable by [13] and DEXPTIME-complete by [8].

Classically, there are two extensions from $\sigma : \mathcal{X} \rightarrow 2^{T_{\text{fin}}(\Sigma \cup H)}$ to a substitution from $T_{\text{fin}}(\Sigma \cup \mathcal{X})$ to $2^{T_{\text{fin}}(\Sigma \cup H)}$ [10, 11]: *outside-in* (OI for short) and *inside-out* (IO for short). The corresponding notation is σ_{oi} and σ_{io} respectively. Let $s = x(s_1, \dots, s_r) \in T_{\text{fin}}(\Sigma \cup \mathcal{X})$ with $x \in \mathcal{X} \cup \Sigma$. If $r = 0$, then we let $\sigma_{\text{oi}}(s) = \sigma_{\text{io}}(s) = \sigma(s)$. Recall that $\sigma(x)$ was extended above to be defined for all $x \in \Sigma \cup \mathcal{X}$. For $r \geq 1$ each s_i has less vertices than s . Hence, $\sigma_{\text{io}}(s_i)$ and $\sigma_{\text{oi}}(s_i)$ are defined for all $1 \leq i \leq r$ by induction on the size of s . We let

$$\sigma_{\text{oi}}(s) = \{t_x[i_j \leftarrow t_{i_j}] \mid t_x \in \sigma(x) \wedge t_{i_j} \in \sigma_{\text{oi}}(s_i)\} \quad (1) \quad \text{eq:lea}$$

$$\sigma_{\text{io}}(s) = \{t_x[i_j \leftarrow t_i] \mid t_x \in \sigma(x) \wedge t_i \in \sigma_{\text{io}}(s_i)\} \quad (2) \quad \text{eq:frida}$$

The convention is as above: the i_j ’s run over all leaves labeled by the hole $i \in H$. Hence,

1. For every term $s \in T_{\text{fin}}(\Sigma \cup \mathcal{X})$ we have $\sigma_{\text{io}}(s) \subseteq \sigma_{\text{oi}}(s)$. Moreover, since $\sigma(x) \cap H = \emptyset$, we see by induction: $\sigma_{\text{io}}(s) \subseteq T_{\text{fin}}(\Sigma)$. (No variables and no holes appear in $\sigma_{\text{io}}(s)$ or $\sigma_{\text{oi}}(s)$.)
2. If $h = \sigma$ is a homomorphism, then $h(s) = h_{\text{io}}(s) = h_{\text{oi}}(s)$ for all $s \in T(\Sigma \cup \mathcal{X})$.

¹ We use any of the several equivalent definitions for regular tree languages, e.g. see [5, 22, 20, 25].

² The classical example for nonregularity is: $L = \{x^n(y) \mid n \in \mathbb{N}\}$ where $x, y \in \mathcal{X}$ with $\text{rk}(x) = 1$, $\text{rk}(y) = 0$, $h(x) = a(1, 1)$, and $h(y) = b$.

Our positive results cover the IO interpretation σ_{io} , only. In our setting, it is convenient to deal with finite and infinite trees simultaneously. The first step is to define an adequate notion of *syntactic congruence* for a regular tree language. This can be done directly using non-deterministic tree automata (NTAs for short). Another possibility is to use a general notion of *monad*. For languages over finite trees, see [3].³ For infinite trees a notion of *syntactic algebra* for regular languages was proposed only very recently by Blumensath: [2] characterizes regular languages of infinite trees by finite syntactic algebras.

Our approach circumvents the theory of monads. It is more direct. Still, our results for infinite trees are far from trivial. The first challenge is to define $\sigma_{io}(s)$ for infinite trees such that Equation (2) still holds. The idea is as follows. Let σ be a substitution mapping variables to (possibly empty) sets of finite or infinite trees with holes. Then for a tree s we define a partially defined choice function γ . If u is labeled by a variable x and $\sigma(x) \neq \emptyset$, then $\gamma(u) \in \sigma(x)$. If $\sigma(x) = \emptyset$, then $\gamma(u)$ is not defined, which we denote by $\gamma(u) = \perp$. For each choice function we can associate a Cauchy sequence $\gamma_n(s)$ in the complete metric space $T(\Sigma \cup \mathcal{X} \cup H) \cup \{\perp\}$; and we let $\gamma_\infty(s)$ be its limit. Finally we define

$$\sigma_{io}(s) = \{\gamma_\infty(s) \mid \gamma \text{ is a choice function for } s \text{ and } \gamma_\infty(s) \neq \perp\}. \quad (3)$$

eq:firstchoice

It turns out that this definition satisfies Equation (2). In particular, it coincides with the former definition for finite trees. It also shows that the use of choice functions leads us to the IO definition of $\sigma_{io}(s)$ rather than OI. A technical lemma will show that for a regular set of finite or infinite trees R over Σ , the “inverse image” $\sigma_{io}^{-1}(R) = \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{io}(s) \subseteq R\}$ is a regular set of trees. This fact is the basis for our main result. It can be stated as follows.

thm:mainintro

► **Theorem 1.** *Let $L \subseteq T(\Sigma \cup \mathcal{X})$ and $R \subseteq T(\Sigma)$ be tree languages over $\Sigma \cup \mathcal{X}$ and Σ respectively, where Σ is a finite ranked alphabet of constants, \mathcal{X} is a finite ranked alphabet of variables, and R is regular. Let $\# \in \{\subseteq, =\}$. Then the following hold:*

1. *Every solution of “ $L \# R$?” is less than or equal to a maximal solution.*
2. *The number of maximal solutions is finite.*
3. *If σ is maximal, then $\sigma(x)$ is a regular tree language for all $x \in \mathcal{X}$ (that is: σ is regular); and the set of all maximal solutions is effectively computable.*
4. *Let \mathcal{C} be a class of tree languages over $\Sigma \cup \mathcal{X}$ such that on input $L \in \mathcal{C}$ and a regular tree language K the problem “ $L \# K$?” is decidable. Then, on input $L \in \mathcal{C}$, a regular language $R \subseteq T(\Omega)$, and regular substitutions $\sigma_1, \sigma_2 : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$, the following problem is decidable, too: “ $\exists \sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ such that $\sigma_1 \leq \sigma \leq \sigma_2$ and $\sigma_{io}(L) \# R$?”.*

The generalization of Conway’s result to infinite trees is a special case of Theorem 1 where L is regular, too. However, for $\# = \subseteq$, Item 4 applies also e.g. to context-free tree languages. There are different notions of context-freeness of tree languages over finite trees, see [7, 10, 15, 23]. Moreover, whenever deterministic context-free languages have been defined, then (to the best of our knowledge) on input a deterministic context-free tree language and a regular tree language K , the problem “ $L = K$?” is decidable.

Our proof of Theorem 1 uses the concept of alternating tree automata and parity games. The role of “syntactic algebra” is replaced by an equivalence relation that trees have the same *profile*. By coincidence, a similar (although different) notion of profile is the starting point in [2], too. The coincidence is not surprising. The notion pops up very naturally.⁴

³ Conway’s result for finite trees follows from that: personal communication Mikołaj Bojańczyk, 2019.

⁴ The technical part of [2] does not deal with substitutions. In particular, there is no difference between σ_{oi} and σ_{io} . Since [2] does not consider duplications, the concept of “profile” is simpler than ours.

We show that, without loss of generality, each class defined by a profile contains a finite tree with the same profile. This implies that it is enough to show Theorem 1 when $\sigma(x)$ is a finite set of finite trees. Concerning the complexity issues, we know that the universality problem for tree automata is DEXPTIME-complete, [24], [5, Thm. 1.7.7]. So we cannot be better than that. For the upper bound we do not have any new insight other than applying known worst-case estimation for standard automata constructions. So, we cannot provide any new contribution in that direction. Therefore, we concentrate on decidability questions.

The proof of Theorem 1 is a formal consequence of Corollary 18 for the first three items. Item 4 of Theorem 1 is shown in Corollary 26. Finally, it is worth noting that every solution is less than or equal to a maximal solution holds for all subsets $L \subseteq T_{\text{fin}}(\Sigma \cup \mathcal{X})$ and $R \subseteq T(\Sigma)$ by Zorn's Lemma. In general, this result fails if L is a subset of infinite words. Take $\Sigma = \{a, b\}$, $\mathcal{X} = \{x\}$, $L = (ax)^\omega$, and $R = \{u \in \Sigma^\omega \mid \exists k \geq 1, b^k \text{ is not factor of } u\}$. Then there is no maximal solution. Indeed, given any solution $\sigma(x) \subseteq \{a, b\}^*$, we can define $n = \max\{k \geq 1 \mid b^k \text{ is a factor of some word in } \sigma(x)\}$. Then $\sigma'(x) = \sigma(x) \cup \{b^{n+1}\}$ is another solution; and it is strictly larger than σ . So, we cannot drop the assumption that R is regular, in general. Example 3 shows a similar situation for outside-in substitutions and trees.

2 Notation

sec:result

For $r \in \mathbb{N} = \{0, 1, \dots\}$ we let $[r] = \{1, \dots, r\}$. The power set of a set S is 2^S and S_\perp denotes the disjoint union of S and $\{\perp\}$, where \perp is a special symbol playing the role of “undefined”. In a formal reading, if it happens that $\perp \in S$, then S_\perp uses a different symbol for \perp by definition of a disjoint union. We identify a totally defined mapping $f : S' \rightarrow S_\perp$ with a partially defined function $f_\perp : S' \rightarrow S$ such that the domain of f_\perp is the set $S' \setminus f^{-1}(\perp)$. An element $x \in S$ is identified with the singleton $\{x\} \in 2^S$.

sec:tt

2.1 Trees and metrics

A *tree* s is given with a representation of its vertices as a subset of \mathbb{N}^* (= sequences over \mathbb{N}) as follows: the edge from a vertex u to its i -th child (from the left) is labeled with i ; and u is represented by the label of the unique path from the root to u . If s is the full infinite binary tree, then the set of vertices is $\{1, 2\}^*$. Therefore, s is a partially defined function from the free monoid \mathbb{N}^* over the natural numbers \mathbb{N} to a set of labels Ω . The domain of s is called the *set of positions* and denoted by $\text{Pos}(s)$. It is the set of vertices of the tree and if $u \in \text{Pos}(s)$, then $s(u) \in \Omega$ is the label of the vertex u . The set $\text{Pos}(s)$ is subject to the following conditions. If $(n_1, \dots, n_k) \in \text{Pos}(s)$, then $k \geq 0$. For $k = 0$ this is the empty sequence ε . It is the position of the root. For $k \geq 1$ we have $(n_1, \dots, n_{k-1}) \in \text{Pos}(s)$ and $(n_1, \dots, n_{k-1}, j) \in \text{Pos}(s)$ for all $1 \leq j \leq n_k$. Moreover, if the vertex $u = (n_1, \dots, n_k) \in \text{Pos}(s)$ has d children, then for $1 \leq i \leq d$ the position of the i -th child $u.i$ is the sequence (n_1, \dots, n_k, i) . If u has rank r , then u has exactly r children. Vertices without children are called *leaves*. For $x \in \Omega$ we denote by $\text{leaf}_x(s)$ the set of leaves labeled by x . In particular, $\text{leaf}_x(s) \subseteq \text{Pos}(s)$. The subtree of s rooted at position u is denoted by $s|_u$. Clearly, $s = s|_\varepsilon$. If $s(u) = x$, then x is the label of the root of $s|_u$. If u has r children, then we also use a “term” notation $s|_u = x(s_1, \dots, s_r)$ where $s_i = s|_{u.i}$. Finally, the length $|u|$ is the distance to the root of s .

By $T(\Omega)$ we denote the set of finite and infinite trees $s : \text{Pos}(s) \rightarrow \Omega$ with vertex labels in Ω . There is a standard way to endow $T(\Omega)$ with a metric. We identify s with the totally defined function from $\mathbb{N}^* \rightarrow \Omega$ and, using $2^{-\infty} = 0$, we let $d'(s, s') = 2^{-\inf\{|u| \in \mathbb{N} \mid s(u) \neq s'(u)\}}$. By classical results $T(\Omega)$ becomes a compact space. For our purposes another metric is more

convenient. First, the symbol \perp for “undefined” is now viewed as a constant of rank 0 (not in Ω). Formally, extending $s : \text{Pos}(s) \rightarrow \Omega \cup \{\perp\}$ to a total function on \mathbb{N}^* has to use a different \perp -symbol, but this is not essential. We can use the same symbol, if we accept that a position $u \in \text{Pos}(s)$ can be labeled by \perp . Clearly, if $s(u) = \perp$ for $u \in \text{Pos}(s)$, then u is a leaf.

We define a metric d on $T(\Omega \cup \{\perp\})$ by a case distinction:

$$d(s, s') = \begin{cases} 1 & \text{if either } s \text{ or } s' \text{ uses the symbol } \perp \text{ but not both,} \\ 2^{-\inf\{|u| \in \mathbb{N} \mid u \in \mathbb{N}^* : s(u) \neq s'(u)\}} & \text{otherwise.} \end{cases}$$

Observe that $d'(s, \perp) = d(s, \perp) = 1$ for all $s \in T(\Omega)$. Let \sim denote the equivalence relation \sim on $T(\Omega \cup \{\perp\})$ which identifies all trees $T(\Omega \cup \{\perp\}) \setminus T(\Omega)$ into a single class represented by the term \perp . Since every $s \in T(\Omega \cup \{\perp\})$ where some position is labeled by \perp has the same distance 1 to every tree in $T(\Omega)$, there is a canonical quotient metric d_\sim on the quotient space such that the canonical mapping from the subspace $(T(\Omega \cup \{\perp\}), d') = (T(\Omega \cup \{\perp\}), d)$ to $(T(\Omega \cup \{\perp\})/\sim, d_\sim)$ becomes an isometry.⁵

Hence, the metric spaces $(T(\Omega), d') = (T(\Omega), d)$, $(T_\perp(\Omega), d)$, and $(T(\Omega \cup \{\perp\}), d) = (T(\Omega \cup \{\perp\})/\sim, d_\sim)$ are complete and compact. In particular, Cauchy sequences have unique limits.

The subset of finite terms in $T(\Omega)$ is denoted by $T_{\text{fin}}(\Omega)$. If there is a symbol of rank 0, then $T_{\text{fin}}(\Omega)$ is a discrete, open, and dense subset of $T(\Omega)$. Finally, if $\Delta \subseteq \Omega$, then we view $T(\Delta)$ as a (closed) subset of $T(\Omega)$. Moreover, $\Delta = \Sigma \cup \mathcal{X} \cup H$ is the disjoint union of finite ranked alphabets: Σ = constants, \mathcal{X} = variables, and H is a nonempty set $\{1, \dots, |H|\} \subseteq [\text{rk}_\Omega]$ of *holes*.⁶ Every hole has rank 0. In particular, $T_{\text{fin}}(\Delta) \neq \emptyset$.

2.2 Substitutions

sec:subst

Recall that a (*regular*) *substitution* is defined by a mapping $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ such that $\sigma(x) \subseteq T(\Sigma \cup [\text{rk}(x)]) \setminus H$ (is regular). It is extended by $\sigma(f) = \{f(1, \dots, \text{rk}(f))\}$ for $f \in \Sigma$. Defining $\sigma \leq \sigma'$ if $\sigma(x) \subseteq \sigma'(x)$ for all $x \in \mathcal{X}$ yields a natural partial order on the set of all substitutions. Let \star be the symbol for “outside-in” resp. “inside-out”: $\star \in \{\text{oi}, \text{io}\}$. If s is a finite tree, then we can define $\sigma_\star(s)$ by induction on the size of s . Let $s = x(s_1, \dots, s_r) \in T_{\text{fin}}(\Sigma \cup \mathcal{X})$. If the size of s is 1, then $r = 0$; and for $r = 0$ we let $\sigma_\star(s) = s$. For $r \geq 1$ the sets $\sigma_\star(s_i)$ with $\star \in \{\text{oi}, \text{io}\}$ are defined by induction because each s_i is smaller than s . We define

$$\sigma_{\text{oi}}(s) = \{t_x[i_j \leftarrow t_{i_j}] \mid t_x \in \sigma(x) \wedge t_{i_j} \in \sigma_{\text{oi}}(s_i)\} \quad (4)$$

$$\sigma_{\text{io}}(s) = \{t_x[i_j \leftarrow t_i] \mid t_x \in \sigma(x) \wedge t_i \in \sigma_{\text{io}}(s_i)\} \quad (5)$$

eq:oi

eq:io

The difference between (4) and (5) is that for leaves $i_j \neq i_k$ we may choose different terms t_{i_j} and t_{i_k} in OI, whereas $t_{i_j} = t_{i_k}$ for IO. If $|\sigma(x)| \leq 1$ for all $x \in \mathcal{X}$, then $\sigma_{\text{io}}(s) = \sigma_{\text{oi}}(s)$. However, possibly $\sigma_{\text{io}}(s) \neq \sigma_{\text{oi}}(s)$ if $|\text{leaf}_i(t)| \geq 2$ for some $i \in H$ and $t \in \sigma(x)$ as in Figure 2.

Trees are represented graphically, too. For example, $s = g(a, f(g(a, a)))$ and $t = g(1, f(g(a, 1)))$ are represented in Figure 1. Note that the ranks of g, f, a are equal to 2, 1, 0 respectively.

Theorem 1 states for a regular tree language R that every solution σ of $\sigma_{\text{io}}(L) \# R$ is less than or equal to a maximal solution. We have seen in Section 1 that the hypothesis of

⁵ In general topology there is a general notion of (*pseudo*)*quotient metric*. For the curious reader the appendix explains that d_\sim is indeed the quotient metric.

⁶ N.B.: In other papers holes are sometimes called “variables” if there are no variables at inner positions.



fig:tree1

■ **Figure 1** The right tree has two holes, both are labeled by 1.

$$s = \begin{array}{c} x \\ | \\ y \end{array}, \quad \sigma(x) = \begin{array}{c} f \\ / \quad \backslash \\ 1 \quad 1 \end{array}, \quad \sigma(y) = \{a, b\}, \quad \begin{array}{c} f \\ / \quad \backslash \\ a \quad b \end{array} \in \sigma_{oi}(x) \setminus \sigma_{io}(x)$$

fig:iooi

■ **Figure 2** A “nonlinear” substitution means duplication mode and possibly $\sigma_{io}(s) \not\subseteq \sigma_{oi}(s)$.

regularity on R is necessary, in general. Section 1 also claims that for the first item of the theorem no assumption on L or R is necessary if we consider only IO-substitutions on finite trees. More general, let $T_{fin, \mathcal{X}}(\Sigma \cup \mathcal{X})$ be the subset of trees in $T(\Sigma \cup \mathcal{X})$ where the number of positions labeled by a variable is finite. The following proposition implies the claim.

prop:Zorn

► **Proposition 2.** Let $\# \in \{\subseteq, =\}$ and $\sigma_1, \sigma_2 : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H) \setminus H}$ be two substitutions. Let $L \subseteq T_{fin, \mathcal{X}}(\Sigma \cup \mathcal{X})$ and $R \subseteq T(\Sigma)$ be arbitrary subsets. Then for every substitution $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H) \setminus H}$ such that $\sigma_1 \leq \sigma \leq \sigma_2$ and $\sigma_{io}(L) \# R$ there is a maximal substitution having the same property.

Proof. Let us show that \leq is an *inductive ordering* on the set of solutions to the problem

$$“\exists \sigma : \sigma_1 \leq \sigma^{(i)} \leq \sigma_2 \wedge \sigma_{io}(L) \# R?” \quad (6)$$

eq:Zorn

To see this, let I be a totally ordered index set and $\{\sigma^{(i)} : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H) \setminus H} \mid i \in I\}$ be a set of substitutions such that $\sigma_1 \leq \sigma^{(i)} \leq \sigma_2$, $\sigma_{io}^{(i)}(L) \# R$, and $\sigma^{(i)} \leq \sigma^{(j)}$ for all $i \leq j$. We will show that I has an upper bound σ such that $\sigma_1 \leq \sigma \leq \sigma_2$ and $\sigma_{io}(L) \# R$. We may assume that I is nonempty because every substitution $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H) \setminus H}$ such that $\sigma_1 \leq \sigma \leq \sigma_2$ and $\sigma_{io}(L) \# R$ serves as an upper bound for $I = \emptyset$. (If there is no such substitution σ , then the proposition holds trivially.) Define $\tilde{\sigma}$ by $\tilde{\sigma}(x) = \bigcup \{\sigma^{(i)}(x) \mid i \in I\}$, then $\sigma_1 \leq \sigma^{(i)} \leq \tilde{\sigma} \leq \sigma_2$ is trivial. It remains to show $\tilde{\sigma}_{io}(L) \subseteq R$. To see this, let $t \in \tilde{\sigma}_{io}(L)$. We can write $t = t_x[i_j \leftarrow t_i]$ for some $s = x(s_1, \dots, s_r) \in L$ such that $t_x \in \tilde{\sigma}(x)$ and $t_i \in \tilde{\sigma}_{io}(s_i)$ for $1 \leq i \leq r$. We show that there is an index $k_t \in I$ such that $t \in \sigma_{io}^{(k_t)}(L)$. We use a two-parameter induction. The first and more important parameter is the number of positions in s labeled by a variable. The second parameter is the minimal distance from the root of s to a position labeled by some variable. If s is without any variable, then $s = t \in \sigma_{io}^{(k_t)}(L)$ for every index k_t . Hence, we may assume that some variable occurs in s . By induction there are indices k_i such that $t_i \in \sigma_{io}^{(k_i)}(s_i)$. Moreover, since $t_x \in \tilde{\sigma}(x)$, there is an index k_x with $t_x \in \sigma^{(k_x)}(x)$. Hence, we can define k_t by choosing the maximum of k_x and the k_i 's. Hence, \leq is indeed an *inductive ordering* on the set of solutions to the problem stated in (6). By Zorn's Lemma: every solution is upper-bounded by some *maximal* solution. ◀

Proposition 2 is another indication that the OI-substitutions are more complicated: it fails in the OI-setting. We have the following example.

ex:OInotZORN

► **Example 3.** Let $\Sigma = \{f, a, b\}$ and $\mathcal{X} = \{x, y\}$ with ranks $\text{rk}(f) = 2$, $\text{rk}(x) = \text{rk}(a) = 1$, and $\text{rk}(y) = \text{rk}(b) = 0$. We let $t \in T(\{f, 1\})$ any infinite term with infinitely many leaves

labeled by the hole 1. Let $R \subseteq T(\Sigma)$ be the set of infinite trees where there is some $k \geq 1$ such that the number of a 's on each branch is bounded by k . Define $\sigma^{(n)}(x) = t$, $\sigma^{(n)}(y) = \{a^m b \mid m \leq n\}$. Then we have $\sigma_{\text{io}}^{(n)}(x(y)) \subseteq \sigma_{\text{oi}}^{(n)}(x(y)) \subseteq \sigma_{\text{oi}}^{(n+1)}(x(y)) \subseteq R$. We also have $\sigma_{\text{io}}(x(y)) \subseteq R$ for $\sigma(y) = a^*b$, but there is no maximal solution to the problem “ $\exists \sigma : \sigma_{\text{oi}}(x(y)) \subseteq R$?” as $\sigma(y) = a^*b$ leads $\sigma_{\text{oi}}(x(y))$ out of R .

2.3 Deletions, duplications and the HOM-problem

The aim is to give a visual explanation to the following facts. Due to duplication there is a homomorphism h for the regular set of trees $K = \{x^n(a) \mid n \in \mathbb{N}\}$ such that $h(K)$ is not regular (this is the classical example mentioned in Section 1) and there is a partial homomorphism σ and a regular set $L = \{t_n \mid n \in \mathbb{N}\}$ with leaf set $\{a, z\}$ where $\sigma(z) = \emptyset$, nevertheless $\sigma(K) = h(L)$. For that we let $x, y, z \in \mathcal{X}$ with $\text{rk}(x) = 1$, $\text{rk}(y) = 2$ and $a, b \in \Sigma$ with $\text{rk}(z) = \text{rk}(a) = \text{rk}(b) = 0$, $h(x) = \sigma(x) = \sigma(y) = f(1, 1)$, $h(z) = b$, and $\sigma(z) = \emptyset$. The corresponding trees of height 3 is depicted in Figure 3.

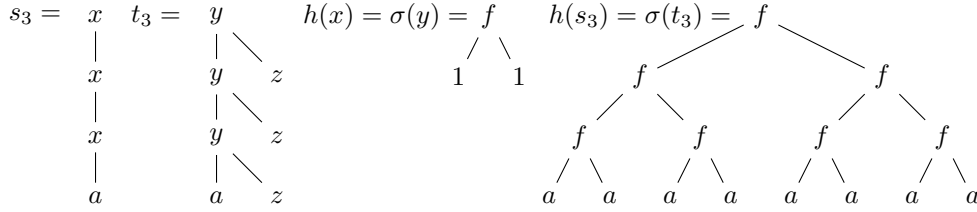


fig:gunnar

■ **Figure 3** $K = x^*(a)$ and $L = \{t_n \mid n \in \mathbb{N}\}$ are regular, but $h(K) = \sigma_{\text{io}}(L) = \sigma_{\text{oi}}(L)$ are not.

3 Choice functions

sec:fta

If s is a finite tree, then we defined $\sigma_{\text{io}}(s)$ by Equation (5). The next step is to define $\sigma_{\text{io}}(s)$ for finite and infinite terms $s \in T(\Sigma \cup \mathcal{X})$ simultaneously without altering Equation (5) for finite trees.

A *choice function* for a tree $s : \text{Pos}(s) \rightarrow \Sigma \cup \mathcal{X}$ is a function $\gamma : \text{Pos}(s) \rightarrow T_{\perp}(\Sigma \cup H)$. Thus, a choice function selects a tree on $\text{Pos}(s) \setminus \gamma^{-1}(\perp)$. By $\Gamma_{\perp}(s)$ we denote the set of choice functions for s . If $u \in \text{Pos}(s)$ is any vertex in the tree s and $s' = s|_u$, then we can write $\text{Pos}(s') = \{v \mid uv \in \text{Pos}(s)\}$. That is, $u \text{Pos}(s') = \text{Pos}(s)$. Thus, a choice function γ for s defines by $\gamma|_u(v) = \gamma(uv)$ a unique choice function $\gamma|_u : \text{Pos}(s') \rightarrow T_{\perp}(\Sigma \cup H)$ for each subtree $s|_u$. Having this, we define $\gamma_n(s) \in T_{\perp}(\Sigma \cup H)$ for all $n \in \mathbb{N}$ by induction. We let $\gamma_0(s) = \gamma(\text{root}(s))$. For $n \geq 1$ we let $\gamma_n(s) = \gamma(\text{root}(s))[i_j \leftarrow (\gamma|_i)_{n-1}(s_i)]$.

lem:cauchy

► **Lemma 4.** *The sequence $n \mapsto \gamma_n(s)$ is a Cauchy sequence in the complete metric space $((T_{\perp}(\Sigma \cup H), d) = (T_{\perp}(\Sigma \cup H)/\sim, d_{\sim})$ which therefore has unique limit $\lim_{n \rightarrow \infty} \gamma_n(s)$. Moreover, $\lim_{n \rightarrow \infty} \gamma_n(s) \in T_{\perp}(\Sigma)$ is either \perp or a tree without holes and without variables.*

Proof. Assume first that $\gamma_m(s) = \perp$ for some $m \in \mathbb{N}$. Then, by induction, we have $\gamma_n(s) = \perp$ for all $n \geq m$. Since ultimately constant sequences are Cauchy sequences in every metric space, we are done. Thus, without restriction, we have $\perp \neq \gamma_n(s) \in T(\Omega)$ for all $n \in \mathbb{N}$.

Recall that $\sigma(x) \notin H$. It follows by induction that for $1 \leq m \leq n$ the trees $\gamma_m(s)$ and $\gamma_n(s)$ agree on all positions u where $|u| \leq m$; and moreover, $\gamma_m(s)(u) = \gamma_n(s)(u)$ for these positions. Thus, for all $1 \leq m \leq n$ the distance satisfies $d(\gamma_m(s), \gamma_n(s)) \leq 2^{-m}$. ◀

def:signeqes

► **Definition 5.** Let $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup \mathcal{X})}$ be a substitution and $\gamma \in \Gamma_{\perp}(s)$ such that $\gamma(u) \in \sigma(s(u)) \cup \{\perp\}$ for all $u \in \text{Pos}(u)$. Let $\gamma_{\infty}(s) = \lim_{n \rightarrow \infty} \gamma_n(s)$, and extend $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ to a substitution $\sigma_{io} : T(\Sigma \cup \mathcal{X}) \rightarrow 2^{T(\Sigma)}$ by defining $\sigma_{io}(s) = \Gamma(\sigma, s)$ where

$$\Gamma(\sigma, s) = \bigcup \{ \gamma_{\infty}(s) \mid \gamma \in \Gamma_{\perp}(s) \wedge \forall u \in \text{Pos}(s) : \gamma(u) \in \sigma(s(u)) \cup \{\perp\} \} \setminus \{\perp\}. \quad (7)$$

eq:defsio

By Lemma 4 every tree $\gamma_{\infty}(s)$ is either equal to \perp (hence, undefined) or a tree without variables or holes. Therefore, $\sigma_{io}(s)$ is a (possibly empty) set of trees in $T(\Sigma)$.

prop:gamsio

► **Proposition 6.** Let $s = x(s_1, \dots, s_r) \in T(\Sigma \cup \mathcal{X})$ be a finite tree and $\gamma \in \Gamma(s)$ be a choice function. Then $\gamma_{\infty}(s) = \gamma(\text{root}(s))[i_j \leftarrow (\gamma|_i)_{\infty}(s_i)]$.

Proof. The sequence $n \mapsto \gamma(\varepsilon)[i_j \leftarrow (\gamma|_i)_{n-1}(s_i)]$ is a Cauchy sequence. Its limit is $\gamma(\varepsilon)[i_j \leftarrow \lim_{n \rightarrow \infty} (\gamma|_i)_{n-1}(s_i)]$. Since $\gamma_{\infty}(s_i) = \lim_{n \rightarrow \infty} (\gamma|_i)_n(s_i)$ for all i , we obtain

$$\begin{aligned} \gamma(\text{root}(s))[i_j \leftarrow (\gamma|_i)_{\infty}(s_i)] &= \gamma(\text{root}(s))[i_j \leftarrow \lim_{n \rightarrow \infty} (\gamma|_i)_n(s_i)] \\ &= \gamma(\text{root}(s))[i_j \leftarrow \lim_{n \rightarrow \infty} (\gamma|_i)_{n-1}(s_i)] = \lim_{n \rightarrow \infty} \gamma(\text{root}(s))[i_j \leftarrow (\gamma|_i)_{n-1}(s_i)] = \lim_{n \rightarrow \infty} \gamma_n(s) \end{aligned}$$

By definition $\lim_{n \rightarrow \infty} \gamma_n(s) = \gamma_{\infty}(s)$. Hence, the result. ◀

cor:gamsio

► **Corollary 7.** Let $s = x(s_1, \dots, s_r) \in T(\Sigma \cup \mathcal{X})$ be a tree and σ_{io} defined according to Definition 5. Then we have $\sigma_{io}(s) = \{t_x[i_j \leftarrow t_i] \mid t_x \in \sigma(x) \wedge t_i \in \sigma_{io}(s_i)\}$.

In particular, for finite trees the new definition of $\sigma_{io}(s)$ agrees with Equation (5).

Proof. We use the notation of $\Gamma(\sigma, s')$ for $s' \in T(\Sigma \cup \mathcal{X})$ as given in Definition 5, Equation (7).

$$\begin{aligned} \sigma_{io}(s) &= \bigcup \{ \gamma(\text{root}(s))[i_j \leftarrow (\gamma|_i)_{\infty}(s_i)] \mid \gamma \in \Gamma(\sigma, s) \} && \text{by Proposition 6} \\ &= \{t_x[i_j \leftarrow (\gamma|_i)_{\infty}(s_i)] \mid t_x \in \sigma(x), \gamma|_i \in \Gamma(\sigma, s_i)\} && \text{trivial} \\ &= \{t_x[i_j \leftarrow t_i] \mid t_x \in \sigma(x) \wedge t_i \in \sigma_{io}(s_i)\} && \text{by Definition 5} \end{aligned}$$

Hence, $\sigma_{io}(s) = \{t_x[i_j \leftarrow t_i] \mid t_x \in \sigma(x) \wedge t_i \in \sigma_{io}(s_i)\}$ as desired. ◀

4 Games and alternating automata

4.1 Parity games

sec:pg

An *arena* is a directed graph $A = (V, E)$ with $E \subseteq V \times V$ such that the set of vertices is a disjoint union $V = V_0 \cup V_1$. We allow that an arena has *sinks* (vertices without outgoing edges). A *parity game* is defined by a pair (A, χ) where A is an arena and $\chi : V \rightarrow C$ is a mapping to a set of *colors* C . Without restriction, we assume that $C = \{1, \dots, |C|\}$ and that $|C|$ is odd. Let $p_0 \in V$. A *game at p_0* is a finite or infinite sequence p_0, p_1, \dots such that $(p_{i-1}, p_i) \in E$ for all p_i in the sequence where $i \geq 1$. Moreover, we require that a game is infinite unless it ends in a sink. There are two players $P = P_0$ (*Prover*) and $S = P_1$ (*Spoiler*). The rules of the game are as follows. It starts in some vertex p_0 . If for $m \geq 0$ a path p_0, \dots, p_m is already defined and $p_m \in V_i$, then player P_i chooses an outgoing edge $(p_m, p_{m+1}) \in E$. If there is no outgoing edge, then player P_i lost. If the game does not end in a sink, the mutual choices define an infinite sequence, unless it ends in a sink. Prover P_0 wins an infinite game if the least color which appears infinitely often in $\chi(p_0), \chi(p_1), \dots$ is even. Otherwise, Spoiler P_1 wins that game.

A *positional strategy* for player P_i is a subset of edges $E_i \subseteq V_i \times V \subseteq E$ such that for each $u \in V_i$ there is at most one edge $(u, v) \in E_i$. Each positional strategy defines a subarena

$A_i = (V, E_i \cup E \cap (V_{1-i} \times V))$. For the arena A_i the game becomes a solitaire game for player P_{1-i} : indeed, P_{1-i} wins in position p_0 if and only if there exists some path starting at p_0 which satisfies his winning condition. Let $W_i \subseteq V$ be the set of positions $p \in V$ where no path starting at p satisfies the winning condition of P_{1-i} . Then W_i contains those position where player P_i wins positional (also called *memoryless*) by choosing E_i . The set W_i is a set of *winning positions* for P_i , because player P_i wins, no matter how P_{1-i} decides for $p_m \in V_{1-i}$ on the next edge $(p_m, p_{m+1}) \in E$.

thm:GH82

► **Theorem 8** ([16]). *Let $E_i = V_i \times V \subseteq E$ be a positional strategy for player P_i and let $W_i \subseteq V$ be its set of winning positions. Then the opponent, player P_{1-i} has a positional strategy $E_{1-i} \subseteq V_{1-i} \times V \subseteq E$ such that the corresponding set of winning positions W_{1-i} satisfies $W_{1-i} = V \setminus W_i$. That is, V is the disjoint union of W_{1-i} and W_i .*

Theorem 8 implies that for parity games there is no better strategy than a positional one. The result is due to Gurevich and Harrington. Simplified proofs are in [14, 26].

4.2 Nondeterministic and alternating tree automata.

sec:rtlata

Let Δ be a finite alphabet with a rank function $\text{rk} : \Delta \rightarrow \mathbb{N}$. In the application: $\Delta = \Sigma \cup \mathcal{X} \cup H$. Regular tree languages in $T(\Delta)$ are represented either by parity-NTAs (= top-down nondeterministic tree automata with a parity acceptance condition) or by parity-ATAs (= alternating tree automata with a parity acceptance condition). A *parity-ATA* for Δ is a tuple $A = (Q, \Delta, \delta, \chi)$ where Q is a finite set of states, δ is the *transition relation*, and $\chi : Q \rightarrow C$ is a coloring with $C = \{1, \dots, |C|\}$. Without restriction $|C|$ is odd. For each $(p, f) \in Q \times \Delta$ there is exactly one transition which has the form (p, f, Φ) where Φ is an element in the free distributive lattice over the set $[\text{rk}(f)] \times Q$. Since every such Φ can be written in disjunctive normal form, we content ourselves to give a more specific form. Each $(p, f, \Phi) \in \delta$ is written as

$$(p, f, \bigvee_{j \in J} \bigwedge_{k \in K_j} (i_k, p_k)) \quad (8)$$

eq:ataPhi

where J and K_j are finite index sets and $(i_k, p_k) \in [\text{rk}(f)] \times Q$. By definition, a tree $t \in T(\Delta)$ is accepted at a state p if Prover P_0 has a strategy to win the following parity-game at vertex $(\text{root}(t), p)$. The vertices in V_0 belonging to P_0 are all pairs in $\text{Pos}(t) \times Q$. The color of (u, q) is $\chi(q)$, and there is a unique transition $(q, f, \bigvee_{j \in J} \bigwedge_{k \in K_j} (i_k, p_k))$ where $f = t(u)$. If J is empty, then prover lost. (An empty disjunction is “false”.) In the other case prover P_0 chooses an index $j \in J$. Then it is the turn of Spoiler P_1 . Formally, we are at a vertex (u, q, j) of the arena belonging to P_1 with color $\chi(q)$. If K_j is empty, then Spoiler lost. (An empty conjunction is “true”.) Otherwise, Spoiler chooses an index $k \in K_j$ and the game continues at the vertex $(u, i_k, p_k) \in V_0$. That is, the game continues at the position of the i_k -th child of u at state p_k . Prover wins an infinite game if and only if the least color seen infinitely often is even. We write $L(A, p)$ for the set of trees $t \in T(\Delta)$ where P_0 is able to win every game at $(\text{root}(t), p)$.

A *parity-NTA* A is a special instance of an alternating automaton. For convenience we use only the traditional form that δ is a collection of tuples $(p, f, p_1, \dots, p_{\text{rk}(f)})$. Therefore:

$$\delta \subseteq \bigcup_{f \in \Delta} Q \times \{f\} \times Q^{\text{rk}(f)}. \quad (9)$$

eq:ntadel

def:parityacc

► **Definition 9.** *Let A be a parity-NTA and let $t \in T(\Delta)$.*

- A run ρ of t is a relabeling of the positions of t by states which is consistent with the transitions. (See [14] for details.) If $\rho(\varepsilon) = p$, then we say that ρ is run of t at state p .
- A run ρ is accepting if on every infinite directed path (away from the root) (u_0, u_1, u_2, \dots) in $\text{Pos}(t)$ the number $\max \{ \min \{ \chi \rho(u_i) \mid i \geq k \} \mid k \in \mathbb{N} \}$ is even: for all infinite directed paths in the tree $\chi \rho$ it holds that the minimal color appearing infinitely often is even.
- By $\text{Run}_{\text{acc}}(t, p)$ we denote the set of accepting runs of t at state p .
- The accepted language of A at state p is given by $L(A, p) = \{ t \in T(\Delta) \mid \text{Run}_{\text{acc}}(t, p) \neq \emptyset \}$.

The main results of alternating tree automata by [20, 21] can be formulated as follows.

1. Let A be a parity-NTA. Viewing A as an alternating automaton with a game semantics or using (9) yields the same sets $L(A, p)$.
2. Parity-ATAs characterize the class of regular tree languages as defined e.g. in [14, 25]: if $L(A, p)$ is defined by a parity-ATA A with a state p , then we can construct effectively a parity-NTA B and a state q such that $L(A, p) = L(B, q)$.

def:G(gam,s)

► **Definition 10.** Let $s \in T(\Sigma \cup \mathcal{X})$, and $\gamma : \text{Pos}(s) \rightarrow T_{\perp}(\Sigma \cup H)$ be a choice function. The game $G(\gamma, s)$ is defined as follows. The set of vertices V of the arena is the disjoint union of the following two sets:

- $V_0 = \text{Pos}(s) \times Q$ belongs to prover $P = P_0$ and the color of (u, q) is given by $\chi(q)$.
- $V_1 = \{ (u, q, \rho) \mid u \in \text{Pos}(s), q \in Q, \rho \in \text{Run}_{\text{acc}}(\gamma(u), q) \} \cup \{ (u, q, \rho, i_j) \mid u \in \text{Pos}(s), q \in Q, \rho \in \text{Run}_{\text{acc}}(\gamma(u), q), i_j \in \text{leaf}_i(\gamma(u)) \}$.

The set V_1 belongs to spoiler $S = P_1$. The color of (u, q, ρ) is given by $\chi(q)$, and the color of (u, q, ρ, i_j) is the minimal color in the tree $\chi \rho : \text{Pos}(\gamma(u)) \rightarrow C$ which appears on the unique path from the root of $\gamma(u)$ to the leaf i_j .

The outgoing edges are defined as follows: Each vertex (u, q) has outgoing edges to all (u, q, ρ) where $\rho \in \text{Run}_{\text{acc}}(\gamma(u), q)$. Each vertex (u, q, ρ) has outgoing edges to all (u, q, ρ, i_j) where $i_j \in \text{leaf}_i(\gamma(u))$. For each vertex $(u, q, \rho, i_j) \in V_1$ there is exactly one outgoing edge to the vertex $(u, i, \rho(i_j)) \in V_0$. This defines the arena.

lem:game

► **Lemma 11.** Let $q_0 \in Q$ be a state of the parity-NTA B , $s \in T(\Sigma \cup \mathcal{X})$, and $\gamma \in \Gamma_{\perp}(s)$. Then $\gamma_{\infty}(s) \in L(B, q_0)$ if and only if prover P_0 has a (positional) winning strategy for the game $G(\gamma, s)$ in Definition 10 starting at the vertex (ε, q_0) .

Proof. First, let $\gamma_{\infty}(s) \in L(B, q_0)$. Then there is an accepting run at q_0 , say $\rho : \text{Pos}(\gamma_{\infty}(s)) \rightarrow Q$. After every move of spoiler and reaching a vertex $(u, q) \in \text{Pos}(s) \times Q$, knowing the history of the game, P_0 knows the corresponding position of a vertex $v \in \gamma_{\infty}(s)$. Prover chooses the local run ρ_v of $(\gamma_v)_{\infty} = \gamma(u)[i_j \leftarrow (\gamma_{|v, i_j})_{\infty}]$ at $\rho(v)$ induced by ρ . This defines the edge to vertex (u, q, ρ_v) chosen by P_0 . Since $\gamma_{\infty}(s) \in L(B, q_0)$, this yields a winning strategy for prover.

Second, assume $\gamma_{\infty}(s) \notin L(B, q_0)$ and that, by contradiction, P_0 can win the game. By Theorem 8, if P_0 can win, then he can win with a positional strategy. Thus, whenever prover is at a vertex (u, q) , prover has to choose the same run. Considering all possible moves of spoiler, we obtain a run ρ of $\gamma_{\infty}(s)$ at q_0 . The run is not accepting. Hence, there must be a non-accepting path. If this is a path where at some (u, q) there is no accepting run of $\gamma(u)$ at q , then prover lost. We are done. In the other case, the path is infinite. Spoiler can choose his leaves i_j according to that path. Spoiler wins. Contradiction. ◀

Throughout, $B = (Q, \Sigma, \delta, \chi)$ denotes a parity-NTA accepting trees over Σ (no variables and no holes). Given B , we define the automaton $B_H = (Q, \Sigma \cup H, \delta_H, \chi)$ by letting

$\delta_H = \delta \cup Q \times H$. Thus, if a leaf is labeled by a hole, then we “accept” that hole in every state. Let us introduce the *best-ordering* \preceq_{best} on the set $\{0, \dots, |C|\}$. Note that we explicitly include 0 which is not in $\chi(Q)$ and that $|C|$ is odd by our convention. We let

$$0 \preceq_{\text{best}} 2 \preceq_{\text{best}} \dots \preceq_{\text{best}} |C| - 1 \preceq_{\text{best}} |C| \preceq_{\text{best}} |C| - 2 \preceq_{\text{best}} \dots \preceq_{\text{best}} 3 \preceq_{\text{best}} 1$$

Thus, for P_0 even numbers are “better” than odd numbers. We have $m \preceq_{\text{best}} n \iff m \leq n$ for even numbers m, n and $m \preceq_{\text{best}} n \iff m \geq n$ for odd number m, n .

5 Tasks and profiles

def:tau

► **Definition 12.** A task is a tuple $\tau = (p, \psi_1, \dots, \psi_{|H|})$ such that $p \in Q$ and ψ_i is a function $\psi_i : Q \rightarrow \{0\} \cup \chi(Q)$ for $1 \leq i \leq |H|$. We say that a tree $t : \text{Pos}(t) \rightarrow \Sigma \cup H$ in $T(\Sigma \cup H)$ satisfies the task $(p, \psi_1, \dots, \psi_{|H|})$ if the following two assertions are satisfied:

1. The tree t has an accepting run ρ at p w.r.t. the NTA B_H . That is $\text{Run}_{\text{acc}}(B_H, p) \neq \emptyset$.
2. The run ρ guarantees the following condition for all leaves $i_j \in \text{leaf}_i(t)$.
 - If $c_\rho(i_j)$ denotes the minimal color on the path from the root to position i_j in the tree $\chi_\rho : \text{Pos}(t) \rightarrow C \in T(C)$, then we have $c_\rho(i_j) \preceq_{\text{best}} \psi_i(\rho(i_j))$.

If $t \in T(\Sigma \cup H)$ satisfies a task $\tau = (p, \psi_1, \dots, \psi_{|H|})$, then we write $t \models \tau$.

The profile $\pi(t)$ is the subset of tasks $\pi(t) = \{(p, \psi_1, \dots, \psi_{|H|}) \mid t \models (p, \psi_1, \dots, \psi_{|H|})\}$.

We let $\mathcal{P} = \{\pi(t) \mid t \in T(\Sigma \cup H)\}$ be the set of all profiles.

By \equiv_B we denote the equivalence relation which is defined by $t \equiv_B t' \iff \pi(t) = \pi(t')$.

For $\pi \in \mathcal{P}$ we write $t \models \pi$ if there is the equivalence: $t \models \tau \iff \tau \in \pi$.

We have $|\mathcal{P}| \leq 2^{|Q| \cdot (|C|+1)^{|Q \times H|}}$. In particular, \equiv_B is of finite index. The value 0 in the range of ψ_i plays the following role: If $t \models \tau$ and $\psi_i(p) = 0$, then there is no accepting run ρ such that $\rho(i_j) = p$. The condition is vacuously true if $\text{leaf}_i(t) = \emptyset$.

lem:taskreg

► **Lemma 13.** Let τ be a task. The set of trees $\{t \in T(\Sigma \cup H) \mid t \models \tau\}$ is effectively regular and hence, $\{t \in T(\Sigma \cup H) \mid \pi(t) = \pi\}$ is effectively regular for every profile π .

More precisely, if $\tau = (p, \psi_1, \dots, \psi_{|H|})$, then $\{t \in T(\Sigma \cup H) \mid t \models \tau\} = L(B_\tau, (p, \chi(p)))$ where $B_\tau = (Q \times C, \Sigma \cup H, \delta_\tau, \chi_\tau)$ is the following parity-NTA.

- If $p \in Q$ and $f \in \Sigma$ with $r = \text{rk}(f)$, then we define δ_τ via

$$((p, c), f, (p_1, \min\{c, \chi(p_1)\}), \dots, (p_r, \min\{c, \chi(p_r)\})) \in \delta_\tau \iff (p, f, p_1, \dots, p_r) \in \delta_B.$$

- If $p \in Q$ and $i_j \in \text{Pos}(t)$ is labeled by a hole i , then $(p, c) \in \delta_\tau$ if and only if $c \preceq_{\text{best}} \psi_i(p)$.
- $\chi_\tau((p, c)) = \chi(p)$.

Proof. There is a canonical one-to-one correspondence between runs ρ for t at state p w.r.t. B_H and runs ρ_τ for t at state $(p, \chi(p))$ w.r.t. B_τ . If ρ_τ is accepting, then ρ is accepting. For the other direction, let ρ be accepting. Then, by construction, ρ_τ labels a leaf $i_j \in \text{leaf}_i(t)$ with a state $(\rho(i_j), c_\rho(i_j))$ where $c_\rho(i_j)$ is the minimal color on a path from the root to the leaf i_j in the tree $\rho(t)$. By definition: $((\rho(i_j), c_\rho(i_j)), i) \in \delta_\tau$ if and only if ρ is a witness for $t \models \tau$ if and only if ρ_τ is accepting. ◀

prop:pireg

► **Proposition 14.** Let π be a profile, then $\{t \in T(\Sigma \cup H) \mid \pi(t) = \pi\}$ is effectively regular.

Proof. This is a direct consequence of Lemma 13 and the well-known fact that the class of regular tree languages over every $T(\Delta)$ forms an effective Boolean algebra [22]. ◀

prop:equivasl

► **Proposition 15.** *Let $q_0 \in Q$ be state of the NTA B , $s \in T(\Sigma \cup \mathcal{X})$, and $\gamma, \gamma' : \text{Pos}(s) \rightarrow T_\perp(\Sigma \cup H)$ be two choice functions for s such that $\gamma'(u) \equiv_B \gamma(u)$ for all $u \in \text{Pos}(s)$. If $\gamma_\infty(s) \in L(B, q_0)$, then $\gamma'_\infty(s) \in L(B, q_0)$, too.*

Proof. By Lemma 11, prover P_0 wins the game $G(\gamma, s)$ with a positional strategy. Thus, for each vertex $(u, p) \in V_0$, prover P_0 decides on an accepting run $\rho_{u,p}$ for $\gamma(u)$ at state p . The run $\rho_{u,p}$ defines a unique minimal task $\tau = \tau_{\rho_{u,p}}(p, \psi_1, \dots, \psi_{|H|})$ (w.r.t. the natural partial order on tasks defined by \preceq_{best}) such that $\gamma(u) \models \tau$. The minimality of τ implies that

$$\psi_i(q) = \sup_{\preceq_{\text{best}}} \{\chi(\rho_{u,p}(i_j)) \in \{0, \dots, |C|\} \mid \exists i_j : q = \rho_{u,p}(i_j)\}.$$

In particular, there is no leaf i_j in t_u where $q = \rho_{u,p}(i_j)$ if and only if $\psi_i(q) = 0$. Note that τ depends on the triple $(u, p, \rho_{u,p})$. In particular, the functions ψ_i depend on $(u, p, \rho_{u,p})$, too.

Since $\gamma'(u) \equiv_B \gamma(u)$, prover $P' = P'_0$ for the game $G(\gamma', s)$ chooses for each vertex $(u, p) \in V_0 = V'_0$ some run $\rho'_{u,p}$ for $\gamma'(u)$ such that $\gamma'(u) \models \tau$. We have to show that spoiler $S' = P'_1$ cannot win the game. If, by contradiction, S' won the game, then S' can do so by a positional strategy. The positional strategies of P' and S' define a unique path through the arena. Now, whenever such a path leads from $(u, p, \rho'_{u,p})$ to the vertex $(u, p, \rho'_{u,p}, i_{j'})$, then there is some minimal color c' which appears in $\chi \rho'_{u,p}$ on the unique path from position u to the leaf $i_{j'}$. We have $\gamma'(u) \models \tau$, hence $c' \preceq_{\text{best}} \psi_i(\rho'_{u,p}(i_{j'}))$. Since $\rho_{u,p}$ is a witness for $\gamma(u) \models \tau$ and τ was chosen to be minimal, $\gamma(u)$ has some leaf $i_j \in \text{leaf}_i(\gamma(u))$ such that the color of $(u, \rho_{u,p}, i_j)$ is equal to $\psi_i(\rho_{u,p}(i_j))$. In that way prover P translates the strategy of S' into a positional strategy of spoiler P_1 in the game $G(\gamma_\infty, s)$. However, P_0 wins that game and $c' \preceq_{\text{best}} \psi_i(\rho'_{u,p}(i_{j'}))$ for all positions. Thus, P'_0 wins, too. A contradiction. ◀

def:sat

► **Definition 16.** *Let $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ be a substitution. The B -saturation of σ is the substitution $\hat{\sigma} : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ defined by $\hat{\sigma}(x) = \{t' \in T(\Sigma \cup \mathcal{X}) \mid t' \equiv_B t \in \sigma(x)\}$.*

prop:satsig

► **Proposition 17.** *Let $q_0 \in Q$ be state of the NTA B and $L \subseteq T(\Sigma \cup \mathcal{X})$ be any subset. If $\sigma_{io}(L) \subseteq L(B, q_0)$, then $\hat{\sigma}_{io}(L) \subseteq L(B, q_0)$, too.*

Proof. By definition of σ_{io} , it is enough to show the following claim for each $s \in T(\Sigma \cup \mathcal{X})$. If γ and γ' are choice functions for s such that $\gamma(u) \equiv_B \gamma'(u)$ for all $u \in \text{Pos}(s)$, then $\gamma_\infty(s) \in L(B, q_0)$ implies $\gamma'_\infty(s) \in L(B, q_0)$. This is exactly the statement of Proposition 15. ◀

cor:maxioreg

► **Corollary 18.** *Let $R \subseteq T(\Sigma)$ be a regular tree language and $L \subseteq T(\Sigma \cup \mathcal{X})$ be any subset. Then for every $\sigma : \mathcal{X} \rightarrow T(\Sigma \cup H)$ such that $\sigma_{io}(L) = R$ (resp. $\sigma_{io}(L) \subseteq R$) there is some maximal substitution $\sigma' : \mathcal{X} \rightarrow T(\Sigma \cup H)$ such that $\sigma'_{io}(L) = R$ (resp. $\sigma'_{io}(L) \subseteq R$) and $\sigma \leq \sigma'$ (for the natural partial order by set inclusion for components). Every maximal substitution is regular; and the set of maximal substitutions satisfying $\sigma'_{io}(L) = R$ (resp. $\sigma'_{io}(L) \subseteq R$) is finite and effectively computable.*

Proof. (Sketch) Proposition 17 implies the first assertion in the corollary and that the number of maximal solutions is finite because the relation \equiv_B is of finite index. The remaining assertions follow from Proposition 14. ◀

The following result will be shown later in the general form in Proposition 23.

lem:SigProf

► **Lemma 19.** *Let $R \subseteq T(\Sigma)$ be a regular tree language and $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ be a substitution such that $\sigma(x)$ is a finite set of finite trees for all $x \in \mathcal{X}$. Then the set of trees*

$$\sigma_{io}^{-1}(R) = \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{io}(s) \subseteq R\}$$

is regular.

Proof. Without restriction, $R = L(B, q_0)$ for some parity-NTA $B = (Q, \Sigma, \delta, \chi : Q \rightarrow C)$. Let's construct an alternating tree automaton D with state set $Q \times C$ as follows. The states are the pairs $(p, c) \in Q \times C$. The transitions can be defined as follows

$$((p, c), x, \bigwedge_{t_x \in \sigma(x)} \bigvee_{\rho \in \text{Run}_{\text{acc}}(t_x, p)} \bigwedge_{i_j \in \text{leaf}(t_x)} (i, (\rho(i_j), c_\rho(i_j)))) \quad (10)$$

eq:autoD

Here, $c_\rho(i_j)$ is the minimal color in the tree $\rho(t_x)$ which appears on the unique path from the root to the leaf i_j . The coloring is defined by $\chi_D((p, c)) = c$.

The expression in (10) (in the free lattice) is not in disjunctive normal form, but the conjunctions and disjunctions are over finite sets. Hence, there is an equivalent disjunctive normal form and using that, the alternating tree automaton D is well-defined.

By [20, 21] the language $L(D, (q_0, \chi(q_0)))$ is regular. Moreover, by the same papers we have $s \in L(D, (q_0, \chi(q_0)))$ if and only if prover P_0 wins the following parity game.

At a state (p, c) when reading x , Spoiler P_1 chooses some term $t_x \in \sigma(x)$. (If $\sigma(x)$ is empty, then Spoiler lost.) After that Prover P_0 chooses some accepting run $\rho \in \text{Run}_{\text{acc}}(t_x, p)$. (If there is no such accepting run, then Prover lost.) Now, Spoiler P_1 chooses some leaf i_j in t_x . (If there is no leaf, then Spoiler lost.) The game continues at state $(\rho(i_j), c_\rho(i_j))$ by reading the symbol of the i -th child of x (which exists if the game is not finished). ◀

Let \mathcal{T} be the set of tasks of the parity-NTA $B = (Q, \Sigma, \delta, \chi)$. Each task has the form $\tau = (q, \psi_1, \dots, \psi_{|H|})$. We let $\text{Triple}(\tau) = \{(\psi_i(q), q, i) \in C \times Q \times H \mid i \in H \wedge \psi_i(q) \geq 1\}$. The next step defines finitely many finite trees $t_{\pi, q}$ and a parity-NTA $B_{\mathcal{P}}$ such that for all $q \in Q$ we have $t_{\pi, q} \in L(B_{\mathcal{P}}, q)$ if and only if there is some $t \in L(B_H, q)$. Moreover $t_{\pi, q} \models (q, \psi_1, \dots, \psi_{|H|}) \iff (q, \psi_1, \dots, \psi_{|H|}) \in \pi$. The NTA $B_{\mathcal{P}} = (Q', \Sigma_{\mathcal{P}}, \delta_{\mathcal{P}}, \chi)$ is an extension of B_H . We let $Q' = Q \cup \mathcal{T} \cup C \times Q \times H$ and $\Sigma \subseteq \Sigma_{\mathcal{P}}$ where the coloring χ is extended by $\chi(\tau) = |C|$ for $\tau \in \mathcal{T}$ and $\chi((c, q, i)) = c$ for $(c, q, i) \in C \times Q \times H$. We also add various symbols f_π , g_τ , and $\$$.

def:alpProf

► **Definition 20.** The alphabet $\Sigma_{\mathcal{P}}$ contains Σ and in addition the following symbols:

- There is a special symbol $\$$ with $\text{rk}(\$) = 1$.
- For each $\pi \in \mathcal{P}$ there is one symbol f_π . We let $\text{rk}(f_\pi) = 1$.
- For each $\tau = (q, \psi_1, \dots, \psi_{|H|}) \in \mathcal{T}$ there is one symbol g_τ . We let $\text{rk}(g_\tau) = |\text{Triple}(\tau)|$.

The set of transitions $\delta_{\mathcal{P}}$ of $B_{\mathcal{P}}$ contains all transitions from B_H (that is: $\delta_H = \delta \cup Q \times H$) and in addition the following set of tuples (written in the traditional form for NTAs):

- (q, f_π, τ) for all $\tau = (q, \psi_1, \dots, \psi_{|H|}) \in \pi$.
- $(\tau, g_\tau, t_1, \dots, t_k)$ where $\tau = (q, \psi_1, \dots, \psi_{|H|})$ and the t_j 's run over $\text{Triple}(\tau) \subseteq Q'$.
- $(\$, (c, q, i), q)$ for $(c, q, i) \in \text{Triple}(\tau)$ and $\tau \in \mathcal{T}$.

Since $Q \subseteq Q'$, the set \mathcal{T} is a subset of possible tasks for $B_{\mathcal{P}}$. Therefore, $t \models \tau$ and $t \models \pi$ are well-defined for a task $\tau \in \mathcal{T}$ and a profile $\pi \in \mathcal{P}$.

lem:BProfPi

► **Lemma 21.** Let $t \in T(\Sigma \cup H)$ and $\pi = \pi(t)$. Then for every $(q, \psi_1, \dots, \psi_{|H|}) \in \pi$ there exists a finite tree $t_{\pi, q}$ as on the left side in Figure 4 with $t_{\pi, q} \in L(B_{\mathcal{P}}, q)$. Conversely, if $t' \in T(\Sigma_{\mathcal{P}} \cup H) \cap L(B_{\mathcal{P}}, q')$ such that $\text{root}(t') = f_\pi$, then $q' \in Q$ and $t \in L(B_H, q')$.

Proof. $t_{\pi, q} \in L(B_{\mathcal{P}}, q)$ because $\pi = \pi(t)$ implies $f_\pi, g_\tau \in \Sigma_{\mathcal{P}}$, therefore $t_{\pi, q}$ exists, and ρ depicted on the right side of Figure 4 is an accepting run of $t_{\pi, q}$ in q . Note that for $\text{rk}(g_\tau) = 0$, this is just the term $f_\pi(g_\tau)$ which is accepted. For the converse, let $t' \in T(\Sigma_{\mathcal{P}} \cup H)$ be any tree with root f_π . If $t' \in L(B_{\mathcal{P}}, q')$, then this is due to a transition (q', f_π, τ') with $q' \in Q$ and $\tau' \in \pi$ where the first component of τ' is q' . Since $\pi = \pi(t)$, this implies $t \in L(B_H, q')$. ◀

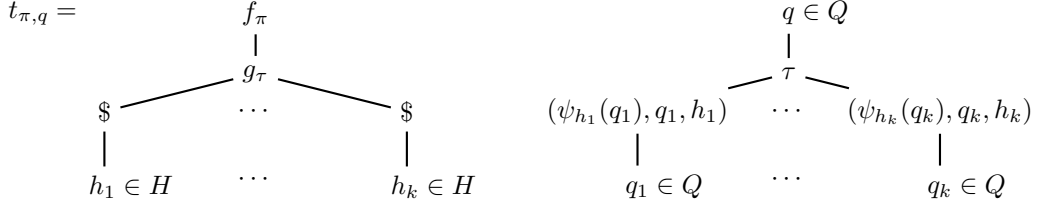


fig:RunProf

■ **Figure 4** $t_{\pi,q}$ and an accepting run ρ on $t_{\pi,q}$ for $\tau = (q, \psi_1, \dots, \psi_{|H|}) \in \pi$.

def:specsat

► **Definition 22.** Let $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ be a substitution and $\pi \in \mathcal{P}$ be a profile. The specialization $\check{\sigma} : \mathcal{X} \rightarrow 2^{T(\Sigma_{\mathcal{P}} \cup H)}$ w.r.t. $B_{\mathcal{P}}$ is defined by the substitution

$$\check{\sigma}(x) = \{t_{\pi,q} \in T(\Sigma_{\mathcal{P}} \cup H) \mid \exists t \in \sigma(x) : t \models \pi\}. \quad (11)$$

eq:spec

6 Proof of Theorem 1

prop:SigProf

► **Proposition 23.** Let $R \subseteq T(\Sigma)$ be a regular tree language and $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ be any substitution. Then the set of trees $\sigma_{io}^{-1}(R) = \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{io}(s) \subseteq R\}$ is regular.

Proof. Without restriction, $R = L(B, q_0)$ for some parity-NTA $B = (Q, \Sigma, \delta, \chi : Q \rightarrow C)$. Obviously, $\{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{io}(s) \subseteq L(B, q_0)\} = \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{io}(s) \subseteq L(B_{\mathcal{P}}, q_0)\}$. Proposition 17 says $\{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{io}(s) \subseteq L(B_{\mathcal{P}}, q_0)\} = \{s \in T(\Sigma \cup \mathcal{X}) \mid \hat{\sigma}_{io}(s) \subseteq L(B_{\mathcal{P}}, q_0)\}$. Note that $\hat{\sigma}_{io}(s)$ refers to $\hat{\sigma}(x) = \{t' \in T(\Sigma_{\mathcal{P}} \cup H) \mid \exists t \in \sigma(x) : t' \equiv_{B_{\mathcal{P}}} t\}$. That is to the saturation with respect to terms in $T(\Sigma_{\mathcal{P}} \cup H)$. Lemma 21 implies $\hat{\sigma}(x) = \check{\sigma}(x)$. Hence,

$$\begin{aligned} \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{io}(s) \subseteq L(B, q_0)\} &= \{s \in T(\Sigma \cup \mathcal{X}) \mid \hat{\sigma}_{io}(s) \subseteq L(B_{\mathcal{P}}, q_0)\} \\ &= \left\{s \in T(\Sigma \cup \mathcal{X}) \mid \check{\sigma}_{io}(s) \subseteq L(B_{\mathcal{P}}, q_0)\right\} = \{s \in T(\Sigma \cup \mathcal{X}) \mid \check{\sigma}_{io}(s) \subseteq L(B_{\mathcal{P}}, q_0)\}. \end{aligned}$$

Since $\check{\sigma}(x)$ is a finite set of finite trees, Lemma 19, yields the result: $\sigma_{io}^{-1}(R)$ is regular. ◀

prop:maxlist

► **Proposition 24.** Let $R \subseteq T(\Sigma)$ be a regular tree language. Without restriction, $R = L(B, q_0)$. Then the number of substitutions $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ such that $\sigma = \hat{\sigma}$ is bounded by $2^{|\mathcal{P} \times \mathcal{X}|}$. Moreover, for each $\kappa : \mathcal{X} \rightarrow 2^{\mathcal{P}}$ and $\# \in \{\subseteq, =\}$ we can compute a regular subset $K_{\kappa} \subseteq T(\Sigma \cup \mathcal{X})$ such that for every $L \subseteq T(\Sigma \cup \mathcal{X})$ the following equivalences hold:

$$\exists \sigma : \sigma_{io}(L) \# R \iff \exists \kappa : L \# K_{\kappa}. \quad (12)$$

eq:LsseK

$$\exists \sigma : \sigma_{io}(L) = R \iff \exists \kappa : L = K_{\kappa}. \quad (13)$$

eq:LeqK

Proof. Let σ and τ be two substitutions. Then $\hat{\sigma} = \hat{\tau}$ if and only if for each $x \in \mathcal{X}$ we have

$$\{\pi \in \mathcal{P} \mid \exists t \in \sigma(x) : t \models \pi\} = \{\pi \in \mathcal{P} \mid \exists t \in \tau(x) : t \models \pi\}.$$

On the other hand, given any function $\kappa : \mathcal{X} \rightarrow 2^{\mathcal{P}}$ we obtain a substitution σ_{κ} by:

$$\sigma_{\kappa}(x) = \{t \in T(\Sigma \cup H) \mid \forall \pi \in \mathcal{P} : t \models \pi \iff \pi \in \kappa(x)\}. \quad (14)$$

eq:sigkap

The substitution σ_{κ} is regular by Proposition 14. Moreover, by construction, we have $\hat{\sigma}_{\kappa} = \sigma_{\kappa}$. We can effectively compute the finite list of these $\sigma_{\kappa}(x)$. For each κ Proposition 23 allows to calculate a regular subset $K_{\kappa} = \sigma_{\kappa,io}^{-1}(R) = \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{\kappa,io}(s) \subseteq R\}$. The equivalences in (12) and (13) hold because for each substitution σ there is some κ with $\sigma \leq \sigma_{\kappa}$ and $\sigma_{io}(L) \# R \iff \sigma_{\kappa,io}(L) \# R$ for $\# \in \{\subseteq, =\}$. The result follows. ◀

cor:effreg

► **Corollary 25.** *Let $R \subseteq T(\Sigma)$ be a regular. If all $\sigma(x)$ belong to a class of languages where the intersection with regular sets is decidable, then $\sigma_{io}^{-1}(R)$ is effectively regular.*

Proof. We can compute $\check{\sigma}(x)$ and apply the construction in the proof of Lemma 19. ◀

cor:ClassCsse=

► **Corollary 26.** *Let $\# \in \{\subseteq, =\}$ and \mathcal{C} be a class of tree languages over $\Sigma \cup \mathcal{X}$ such that on input $L \in \mathcal{C}$ and a regular tree language K the problem “ $L \# K$?” is decidable. Then, on input $L \in \mathcal{C}$, a parity-NTA B with a state q_0 , and regular substitutions $\sigma_1, \sigma_2 : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$, the following problem is decidable, too:*

“ $\exists \sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ such that $\sigma_1 \leq \sigma \leq \sigma_2$ and $\sigma_{io}(L) \# L(B, q_0)$?”.

Proof. If $\sigma : \mathcal{X} \rightarrow 2^{T(\Sigma \cup H)}$ satisfies $\sigma_{io}(L) \# L(B, q_0) \wedge \forall x : \sigma_1(x) \subseteq \sigma(x) \subseteq \sigma_2(x)$, then there is some maximal substitution $\hat{\sigma}$ and $\tilde{\sigma}(x) = \hat{\sigma}(x) \cap \sigma_2(x)$ satisfies the same property. Each $\tilde{\sigma}$ is a regular substitution; and the set of all $\tilde{\sigma}$ satisfying $\forall x : \sigma_1(x) \subseteq \tilde{\sigma}(x)$ is finite and effectively computable. Now, $\tilde{\sigma}_{io}(L) \# L(B, q_0)$ is equivalent to $L \# \tilde{\sigma}_{io}^{-1} L(B, q_0)$. ◀

7 Conclusion and future work

sec:conop

Our results are more general than the corresponding results of Conway for regular word languages. We deal with non-linear substitutions. That is: $\sigma_{io}(L) \neq \sigma_{oi}(L)$, in general. The results in the present paper concern $\sigma_{io}(L)$. Future work should extend Theorem 1 to $\sigma_{oi}(L)$. Outside-in is more complicated, but there is no obvious obstacle to extend the results. The natural idea is to modify the notion of choice function: instead of choosing a single tree from some $\sigma(x)$ for each position labeled by a variable x , an “outside-in” choice function should choose a subset of trees from $\sigma(x)$.

Another open question is whether better results or easier proofs are possible if we restrict R (or L and R) for example to regular tree languages with Büchi acceptance. In this case, the parity condition on infinite paths is replaced by the condition that on every infinite path some repeated (or *final*) state appears infinitely often. Restrictions to other natural subclasses of regular tree languages are of interest, too.

Above, we mentioned Bala’s result [1] that for regular word languages the problem “ $\exists \sigma : \sigma(L) = R$?” is exponentially harder to decide than the question “ $\exists \sigma : \sigma(L) \subseteq R$?”. So, although $\exists \sigma : \sigma_{io}(L) = R$ is decidable for regular tree languages, the underlying computational complexity might be quite high. This is underlined by the fact that “ $\exists \sigma : \sigma_{io}(L) \subseteq R$?” is decidable if L is context-free and R is regular, but in the same setting “ $\exists \sigma : \sigma(L) = R$?” becomes undecidable for languages over finite words.

Another line of future research is to establish (matching) lower and upper complexity bounds for the various decision questions. This is wide open, even in the restricted case of regular languages over finite trees. For example, find a reasonable upper bound for the space complexity of the problem “ $\exists \sigma : \sigma_{io}(L) = R$?” where L, R are regular tree languages.

References

- 1 Sebastian Bala. Complexity of regular language matching and other decidable cases of the satisfiability problem for constraints between regular open terms. *Theory of Computing Systems*, 39(1):137–163, 2006.
- 2 Achim Blumensath. Regular tree algebras. *Logical Methods in Computer Science*, 16, February 2020. URL: <https://lmcs.episciences.org/6101>.
- 3 Mikołaj Bojańczyk. Recognisable languages over monads. *ArXiv e-prints*, abs/1502.04898, 2015. URL: <http://arxiv.org/abs/1502.04898>, arXiv:1502.04898.
- 4 J. Richard Büchi. On a decision method in restricted second-order arithmetic. In *Proc. Int. Congr. for Logic, Methodology, and Philosophy of Science*, pages 1–11. Stanford Univ. Press, 1962.
- 5 H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications, 2007. <http://www.grappa.univ-lille3.fr/tata>.
- 6 John Horton Conway. *Regular algebra and finite machines*. Chapman and Hall, London, 1971.
- 7 Bruno Courcelle. A representation of trees by languages. II. *Theoretical Computer Science*, 7(1):25–55, 1978. URL: [https://doi-org.docelec.u-bordeaux.fr/10.1016/0304-3975\(78\)90039-7](https://doi-org.docelec.u-bordeaux.fr/10.1016/0304-3975(78)90039-7), doi:10.1016/0304-3975(78)90039-7.
- 8 Carles Creus, Adrià Gascón, Guillem Godoy, and Lander Ramos. The HOM problem is EXPTIME-complete. *SIAM J. Comput.*, 45:1230–1260, 2016. URL: <https://doi.org/10.1137/140999104>, doi:10.1137/140999104.
- 9 Volker Diekert, Manfred Kufleitner, Gerhard Rosenberger, and Ulrich Hertrampf. *Discrete Algebraic Methods. Arithmetic, Cryptography, Automata and Groups*. Walter de Gruyter, 2016.
- 10 Joost Engelfriet and Erik Meineche Schmidt. IO and OI. I. *J. Comput. Syst. Sci.*, 15:328–353, 1977. URL: [https://doi.org/10.1016/S0022-0000\(77\)80034-2](https://doi.org/10.1016/S0022-0000(77)80034-2), doi:10.1016/S0022-0000(77)80034-2.
- 11 Joost Engelfriet and Erik Meineche Schmidt. IO and OI. II. *J. Comput. Syst. Sci.*, 16:67–99, 1978. URL: [https://doi.org/10.1016/0022-0000\(78\)90051-X](https://doi.org/10.1016/0022-0000(78)90051-X), doi:10.1016/0022-0000(78)90051-X.
- 12 Seymour Ginsburg and Thomas N. Hibbard. Solvability of machine mappings of regular sets to regular sets. *J. ACM*, 11:302–312, 1964. URL: <http://doi.acm.org/10.1145/321229.321234>, doi:10.1145/321229.321234.
- 13 Guillem Godoy and Omer Giménez. The HOM problem is decidable. *J. ACM*, 60:23:1–23:44, 2013. URL: <http://doi.acm.org/10.1145/2501600>, doi:10.1145/2501600.
- 14 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002. URL: <https://doi.org/10.1007/3-540-36387-4>, doi:10.1007/3-540-36387-4.
- 15 Irène Guessarian. Pushdown tree automata. *Mathematical Systems Theory*, 16(4):237–263, 1983. URL: <https://doi-org.docelec.u-bordeaux.fr/10.1007/BF01744582>, doi:10.1007/BF01744582.
- 16 Yuri Gurevich and Leo Harrington. Trees, automata, and games. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 60–65. ACM, 1982. URL: <https://doi.org/10.1145/800070.802177>, doi:10.1145/800070.802177.
- 17 Dexter Kozen. Lower bounds for natural proof systems. In *Proc. of the 18th Ann. Symp. on Foundations of Computer Science, FOCS’77*, pages 254–266, Providence, Rhode Island, 1977. IEEE Computer Society Press.
- 18 Michal Kunc. The power of commuting with finite sets of words. *Theory of Computing Systems*, 40:521–551, 2007.

bala2006complexity

Blumensath20lmcs

Bojanczyk15arxiv

buc62

tata2007

conway1971regular

Courcelle78-tcs2

CreusGGR16SIAMCOMP

edam16

EngelfrietS77

EngelfrietS78

GinsburgHibbard64

GodoyG13JACM

LNCS2500automata

Guessarian83

GurevichH82stoc

koz77

kunc2007power

- OkhotinKunc19
- 19 Michal Kunc and Alexander Okhotin. Language equations. In Jean-Éric Pin, editor, *Handbook of Automata*. European Mathematical Society, To Appear.
- ullerSchupp87tcs
- 20 David E. Muller and Paul E. Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987. URL: [https://doi.org/10.1016/0304-3975\(87\)90133-2](https://doi.org/10.1016/0304-3975(87)90133-2), doi:10.1016/0304-3975(87)90133-2.
- ullerSchupp95tcs
- 21 David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141:69–107, 1995. URL: [https://doi.org/10.1016/0304-3975\(94\)00214-4](https://doi.org/10.1016/0304-3975(94)00214-4), doi:10.1016/0304-3975(94)00214-4.
- rab69
- 22 Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.
- chimpf-Gallier85
- 23 Karl M. Schimpf and Jean H. Gallier. Tree pushdown automata. *Journal of Computer and System Sciences*, 30(1):25–40, 1985. URL: [https://doi-org.docelec.u-bordeaux.fr/10.1016/0022-0000\(85\)90002-9](https://doi-org.docelec.u-bordeaux.fr/10.1016/0022-0000(85)90002-9), doi:10.1016/0022-0000(85)90002-9.
- Seidl94
- 24 Helmut Seidl. Haskell overloading is DEXPTIME-complete. *Information Processing Letters*, 52:57–60, 1994.
- tho90handbook
- 25 Wolfgang Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 4, pages 133–191. Elsevier Science Publishers B. V., 1990.
- zielonka98tcs
- 26 Wiesław Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200:135–183, 1998. URL: [https://doi.org/10.1016/S0304-3975\(98\)00009-7](https://doi.org/10.1016/S0304-3975(98)00009-7), doi:10.1016/S0304-3975(98)00009-7.

sec:oldconwords

A Theorem 1 for finite and infinite words

The material of this section is not used in the paper. We only included it for readers who are not familiar with notion of *syntactic congruences* or *recognizing morphisms* or even more for those readers who are familiar with these notions in the case of finite words but less or not familiar with them in the corresponding set-up for infinite words. Let Σ be a finite set of constants, \mathcal{X} be a finite set of variables, $L \subseteq (\Sigma \cup \mathcal{X})^*$, and $R \subseteq \Sigma^*$ such that L and R are regular. The aim is to give a self-contained, short, and elementary proof of Conway's result (Theorem 1) with respect to $\# \in \{\subseteq, =\}$. Actually, let us give such a proof for “regular constraints” in the spirit of Corollary 26. (Once we have seen the case of finite words, we explain that essentially the same approach works for infinite words, too.)

1. Let σ_1, σ_2 be mappings from \mathcal{X} to 2^{Σ^*} such that each $\sigma_i(x)$ is regular. Then it is decidable whether the problem

$$“\sigma(L) \# R \wedge \forall x : \sigma_1(x) \subseteq \sigma(x) \subseteq \sigma_2(x)?” \quad (15)$$

eq:Confinword

has a solution $\sigma : \mathcal{X} \rightarrow 2^{\Sigma^*}$.

2. Define $\sigma \leq \sigma'$ by $\sigma(x) \subseteq \sigma'(x)$ for all $x \in \mathcal{X}$. Then every solution is upper bounded by a maximal solution; and the number of maximal solutions is finite.
3. If σ is maximal, then $\sigma(x)$ is regular for all $x \in \mathcal{X}$; and all maximal solutions are effectively computable.

Clearly, we may assume without restriction that $\sigma_1(x) \subseteq \sigma_2(x)$ for all x because otherwise there are no solutions. There is no need to define the notion of a syntactic congruence. Let us start with an NFA $A = (Q, \Sigma, \delta, I, F)$ with $\delta \subseteq Q \times \Sigma \times Q$ and $R = L(A)$. Without restriction, $Q = \{1, \dots, n\}$ with $n \geq 1$. For $p, q \in Q$ we denote by $L[p, q]$ the set of words accepted by the NFA $A_{p,q} = (Q, \Sigma, \delta, \{p\}, \{q\})$. Next, we consider the semiring of Boolean $n \times n$ matrices $\mathbb{B}^{n \times n}$. For each letter $a \in \Sigma$ let M_a be the matrix such that for all p, q we have $M_a(p, q) = 1 \iff a \in L[p, q]$. Since Σ^* is a free monoid, the M_a 's define a homomorphism $\mu : \Sigma^* \rightarrow \mathbb{B}^{n \times n}$ (to the multiplicative structure of $\mathbb{B}^{n \times n}$) such that for all $w \in \Sigma^*$ we have $M_w(p, q) = 1 \iff w \in L[p, q]$ where $M_w = \mu(w)$. The crucial, but easy to verify, observation is that $\mu^{-1}(\mu(R)) = R$.

We are almost done with the proof for finite words. Let $\sigma : \mathcal{X} \rightarrow 2^{\Sigma^*}$ be any substitution such that $\sigma(L) \# R$. Define $\hat{\sigma} : \mathcal{X} \rightarrow 2^{\Sigma^*}$ by $\hat{\sigma}(x) = \mu^{-1}(\mu(\sigma(x)))$. Then $\sigma \leq \hat{\sigma}$ and $\hat{\sigma}(L) \# R$, too. We have $|\mathbb{B}^{n \times n}| = 2^n$. Every list $(\mu^{-1}(S_x) \mid x \in \mathcal{X})$ where $S_x \subseteq \mathbb{B}^{n \times n}$ is a *candidate* for some solution as soon as $\sigma_1(x) \subseteq \mu^{-1}(S_x)$ for all x . Since each $\mu^{-1}(S_x)$ is regular, the list of candidates is computable. Moreover, there are at most $2^{2^{n^2}|\mathcal{X}|}$ candidates.

Since we assume $\sigma_1 \leq \sigma_2$, we compute for each candidate σ another substitution $\tilde{\sigma}$ by $\tilde{\sigma}(x) = \sigma(x) \cap \sigma_2(x)$. The point is that whenever the problem in (15) has any solution $\sigma : \mathcal{X} \rightarrow 2^{\Sigma^*}$, then there is a maximal solution $\tilde{\sigma}$ from the list of candidates which satisfies $\tilde{\sigma}(L) \# R$. The class of regular languages is closed under substitution of letters by regular sets. Hence, $\tilde{\sigma}(L)$ is regular; and we can decide $\tilde{\sigma}(L) \# R$.

Now, let us explain that the case of infinite words can be explained easily in a similar fashion. The starting point are two ω -regular languages $L \subseteq (\Sigma \cup \mathcal{X})^\omega$, and $R \subseteq \Sigma^\omega$. We use the fact that every ω -regular language can be accepted by a nondeterministic Büchi automaton A . The syntax of a Büchi automaton is as above: $A = (Q, \Sigma, \delta, I, F)$ where $\delta \subseteq Q \times \Sigma \times Q$. A word $w \in \Sigma^\omega$ is accepted if there are $p \in I$ and $q \in F$ such that there A allows an infinite path labeled by w which begins in p and visits the state q infinitely often. Instead of working over the Booleans \mathbb{B} we consider the three-element commutative

idempotent semiring $S = \mathbb{B} \cup \{\infty\}$ where we adjoin a new element ∞ to $\mathbb{B} = (\{0, 1\}, +, \cdot, 0, 1)$ with $+$ = max and \cdot = min. We let $\infty \cdot x = \infty$ for $x \neq 0$, $\infty \cdot 0 = 0$ and $\infty + x = \infty$ for all x . Note that 0 is a zero and 1 is neutral in $(S, \cdot, 1)$. The semiring S is a quotient of the semiring $(\mathbb{N} \cup \{\infty\}, +, \cdot, 0, 1)$ with $\infty \cdot 0 = 0$. The homomorphism from $\mathbb{N} \cup \{\infty\}$ to S identifies all positive numbers with 1.

Let us define for $a \in \Sigma$ the matrix $M_a \in S^{n \times n}$ by

$$M_a(p, q) = \begin{cases} 0 & \text{if } (p, a, q) \notin \delta, \\ 1 & \text{if } (p, a, q) \in \delta \text{ but } \{p, q\} \cap F = \emptyset, \\ \infty & \text{otherwise: if } (p, a, q) \in \delta \text{ and } \{p, q\} \cap F \neq \emptyset. \end{cases}$$

For simplicity let us concentrate on the only interesting case where substitutions are given by mappings $\sigma : \mathcal{X} \rightarrow 2^{\Sigma^+}$ (Hence $\sigma(x)$ is a set of nonempty finite words.) Then everything is essentially verbatim to the finite case if we substitute the semiring of Boolean matrices by the semiring $S^{n \times n}$. In particular, the matrices $M_a \in S^{n \times n}$ define a homomorphism $\mu : \Sigma^* \rightarrow S^{n \times n}$. For all $p, q \in Q$ and $w \in \Sigma^*$ the interpretation for $M_w = \mu(w)$ is as follows. We have $M_w(p, q) \neq 0$ if and only if there is path labeled by w from state p to q . Moreover, $M_w(p, q) = \infty$ if and only if there is path labeled by w from state p to q which visits a final state. (More details about that approach are, e.g., in the textbook [9].)

The tricky thing is that at the end we have to decide the problem “ $\tilde{\sigma}(L) \# R?$ ”. It is here where Büchi’s result comes into the play. Using Ramsey theory [4] shows decidability of that problem. If we take it as a blackbox, then Conway’s result for infinite words is essentially as easy as for finite words.

B Quotient metrics

Given a metric space (M, d) and an equivalence relation \sim on M , general topology provides a canonical definition of a quotient (pseudo)metric d_\sim on the quotient space M/\sim . Since the term “quotient metric” appears in our paper, let us explain the connection. For $x \in M$ let $[x] = \{x' \in M \mid x \sim x'\}$ denote its equivalence class. We associate to (M, d) a complete weighted graph with vertex set M/\sim and weight $g([x], [y]) = \inf \{d(x', y') \mid x \sim x' \wedge y \sim y'\}$. (So the weight might be 0 for $[x] \neq [y]$.) Then we define $d_\sim([x], [y])$ by the infimum over all weights of paths in the undirected graph connecting $[x]$ and $[y]$. Paths can be arbitrary long and still have weight 0. Clearly, d_\sim is a pseudometric satisfying

$$0 \leq d_\sim([x], [y]) \leq g([x], [y]) \leq d(x, y).$$

It is well-known that $(M/\sim, d_\sim)$ is characterized by the following universal property. Let (M', d') be a pseudometric space and $f : M \rightarrow M'$ be a mapping such that $d'(f(x), f(y)) \leq d(x, y)$ (that is: f is *metric*). If $x \sim x'$ implies $f(x) = f(x')$ for all x, x' , then the induced mapping $\bar{f} : M/\sim \rightarrow M'$ satisfies $d_\sim(\bar{f}([x]), \bar{f}([y])) \leq d(x, y)$ for all x, y . If for each $[x]$ there exists $x_0 \in [x]$ such that $g([x], [y]) = \inf \{d(x_0, y') \mid y \sim y'\}$ for all $y \notin [x]$, then $g([x], [y]) = d_\sim([x], [y])$ is a metric. This holds for $(T(\Omega \cup \{\perp\})/\sim, d_\sim)$ as defined in Section 2.1. The consequence is $(T(\Omega) \cup \{\perp\}, d) = (T(\Omega \cup \{\perp\})/\sim, d_\sim)$