Regular matching problems for infinite trees

Carlos Camino
FMI, Universität Stuttgart, Germany: cfcamino@gmail.com

Volker Diekert
FMI, Universität Stuttgart, Germany: diekert@fmi.uni-stuttgart.de

Besik Dundua
Kutaisi International University and VIAM, Tbilisi State University: bdundua@gmail.com

Mircea Marin
FMI, West University of Timişoara, Romania: mircea.marin@e-uvt.ro

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

Abstract

We study the matching problem of regular tree languages, that is, "$\exists\sigma : \sigma(L) \subseteq R$?" where L, R are regular tree languages over finite ranked alphabets $\mathcal{X}$ and Σ respectively, and σ is a substitution such that $\sigma(x)$ is a set of trees in $T(\Sigma \cup H) \setminus H$ for all $x \in \mathcal{X}$. Here, H denotes a set of "holes" which are used to define a "sorted" concatenation of trees. Conway studied this problem in the special case for languages of finite words in his classical textbook *Regular algebra and finite machines*. The book was published in 1971. Conway showed that if L and R are regular, then the problem "$\exists\sigma\forall x \in \mathcal{X} : \sigma(x) \neq \emptyset \wedge \sigma(L) \subseteq R$?" is decidable. Moreover, there are only finitely many maximal solutions, the maximal solutions are regular substitutions, and they are effectively computable. We extend Conway's results when L, R are regular languages of finite and infinite trees, and language substitution is applied inside-out, in the sense of Engelfriet and Schmidt (1977/78). More precisely, we show that if $L \subseteq T(\mathcal{X})$ and $R \subseteq T(\Sigma)$ are regular tree languages over finite or infinite trees, then the problem "$\exists\sigma\forall x \in \mathcal{X} : \sigma(x) \neq \emptyset \wedge \sigma_{\mathrm{io}}(L) \subseteq R$?" is decidable. Moreover, there are only finitely many maximal solutions σ, the maximal solutions are regular substitutions and effectively computable. The corresponding question for the outside-in extension σ_{oi} remains open, even in the restricted setting of finite trees.

2012 ACM Subject Classification Theory of computation → 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. Their publication [13] dating back to 1964 showed that it is decidable whether there is a generalized sequential machine which maps L onto R if L and R are regular languages of finite words. The paper also treats several variants of this problem. For example, the authors notice that the decidability cannot be lifted to context-free languages. Another paper in that area is [23]. It appeared 1997 and the authors, Prieur et al., address the problem whether there exists a sequential bijection from a finitely generated free monoid to a given rational set R. Earlier, in the late 1960's Conway studied regular matching problems in the following variant of [13]: Let $\mathcal{X}$, Σ be finite alphabets and $L \subseteq \mathcal{X}^*$, $R \subseteq \Sigma^*$. A substitution $\sigma : \mathcal{X} \to 2^{\Sigma^*}$ is called *solution* of the problem "$L \subseteq R$?" if $\sigma(L) \subseteq R$. In his textbook [6, Chapt. 6], Conway developed a *factorization theory* of formal languages. Thereby he found a nugget in formal language theory: Given regular word languages $L \subseteq \mathcal{X}^*$ and $R \subseteq \Sigma^*$, then it holds:

1. It is decidable whether there is a substitution $\sigma : \mathcal{X} \to 2^{\Sigma^*}$ such that $\sigma(L) \subseteq R$ and $\emptyset \neq \sigma(x)$ for all $x \in \mathcal{X}$.

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.

The original proof is rather technical and not easy to digest. On the other hand, using the algebraic concept of *recognizing morphisms*, elegant and simple proofs exist. Regarding the complexity, it turns out that the problem "$\exists \sigma : \sigma(L) \subseteq R$?" is PSPACE-complete if L and R are given by NFAs by [18]. (The hardness holds for DFAs and $\mathcal{X} = \Sigma$). The apparently similar problem "$\exists \sigma : \sigma(L) = R$?" is more difficult: Bala showed that it is EXPSPACE-complete [1].

Conway also asked whether the unique maximal solution of the language equation $Lx = xL$ is given by a substitution such that $\sigma(x)$ is regular[1]. This question was answered by Kunc in a highly unexpected way: there is a finite set L such that the unique maximal solution $\sigma(x)$ of $Lx = xL$ is co-recursively-enumerable-complete [19]. A recent survey on language equations is in [20].

The present paper generalizes Conway's result to regular tree languages. We consider finite and infinite trees simultaneously. We begin with finite ranked alphabets $\mathcal{X}$ of *variables* and Σ of *function symbols*. In order to define a notion of concatenation we also need a set of *holes* H. These are symbols of rank zero. For simplicity, throughout the set of holes is chosen as $H = \{1, \ldots, |H|\}$. We require $(\Sigma \cup \mathcal{X}) \cap H = \emptyset$. Trees, as considered here, are rooted and they can be written as terms $x(s_1, \ldots, s_r)$ where $r = \mathrm{rk}(x) \geq 0$ and s_i are trees. In particular, all symbols of rank 0 are trees. Words $a_1 \cdots a_n \in \Sigma^*$ (with $a_i \in \Sigma$) are encoded as terms $a_1(\cdots(a_n(\$))\cdots)$ where the a_i are function symbols of rank 1 and the only symbol of rank 0 is \$ which signifies "*end-of-string*". An infinite word $a_1 a_2 \ldots \in \Sigma^\omega$ is encoded as $a_1(a_2(\ldots))$ and no hole appears. In contrast to the case of classical term rewriting, substitutions are applied at inner positions, too. In the word case it is clear what to do. Let $w = xyx \in \mathcal{X}^*$ with $\sigma(x) = L_x$ and $\sigma(y) = L_y$, then we obtain $\sigma(w) = L_x L_y L_x$. Translated to term notation, we obtain $w = x(y(x(\$)))$, $\sigma(z) = \{u(1) \mid u \in L_z\}$ for $z \in \{x, y\}$ with the result $\sigma(w) = L_x L_y L_x(\$)$. On the other hand, in the tree case variables of any rank may exists. As a result, variables may appear at inner nodes as well as at leaves. Throughout, if t is any tree, then $\mathrm{leaf}_i(t)$ denotes the set of leaves labeled by the hole $i \in H$. By i_j we denote the elements of $\mathrm{leaf}_i(t)$.

In the following, a *substitution* means a mapping $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H) \setminus H}$ such that for each $x \in \mathcal{X}$ we have $\sigma(x) \subseteq T(\Sigma \cup \{1, \ldots, \mathrm{rk}(x)\})$. A *homomorphism* (resp. *partial homomorphism*) is a substitution σ such that $|\sigma(x)| = 1$ (resp. $|\sigma(x)| \leq 1$) for all $x \in \mathcal{X}$. We write $\sigma_1 \leq \sigma_2$ if $\sigma_1(x) \subseteq \sigma_2(x)$ for all $x \in \mathcal{X}$; and we say that σ is *regular*[2] if $\sigma(x)$ is regular for all $x \in \mathcal{X}$.

Trees are represented graphically, too. For example, $g(1, f(g(a, 1)))$ and $g(t, f(g(a, t)))$ are represented in Fig. 1. We obtain $g(t, f(g(a, t)))$ by replacing the positions labeled by hole 1 in $g(1, f(g(a, 1)))$ by any rooted tree t. As soon as σ is not a partial homomorphism, one has to distinguish between "Inside-Out" (IO for short) and "Outside-In" (OI for short) as advocated and defined in [11, 12]. Our positive decidability results concern IO, only. The IO-definition $\sigma_{\mathrm{io}}(s)$ for a given tree $s \in T(\Sigma \cup \mathcal{X})$ and a substitution σ has the following interpretation. First, we extend σ to a mapping from $\Sigma \cup \mathcal{X}$ to $2^{T(\Sigma \cup H)}$ by $\sigma(f) = \{f(1, \ldots, \mathrm{rk}(f))\}$ for $f \in \Sigma \setminus \mathcal{X}$. Second, we use a term notation $s = x(s_1, \ldots, s_r)$ (for finite and infinite trees s)

---

[1] There is a unique maximal solution since the union over all solutions is a solution.
[2] There are several equivalent definitions for regular tree languages, e.g. see [5, 24, 21, 22, 26].

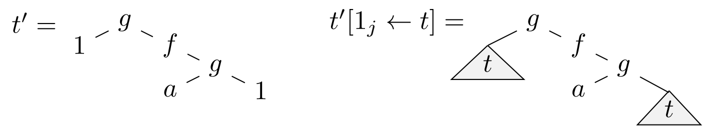

Figure 1 The left tree t' has two positions labeled with hole $1 \in H$. Holes define a composition of trees. The right tree is obtained by composing t' with a tree t over the hole 1 which is denoted by $t'[1_j \leftarrow t]$.

and we let $\sigma_{\text{io}}(s) \subseteq T(\Sigma)$ to be a certain fixed point of the language equation

$$\sigma_{\text{io}}(s) = \bigcup \{t[i_j \leftarrow t_i] \mid t \in \sigma(x) \wedge \forall\, 1 \leq i \leq r : \; t_i \in \sigma_{\text{io}}(s_i)\} \,. \tag{1}$$

The notation $t[i_j \leftarrow t_i]$ in Eq.(1) means that every leaf of t which is labeled by a hole i is replaced by the tree t_i. The idea to compute the elements in $\sigma_{\text{io}}(s)$ is a recursive procedure which at each call first selects the tree $t \in \sigma(x)$ (if x is the label of the root of s) and then it makes recursive calls for each hole i which appears as a label in t to compute the elements t_i. After that, all positions i_j in t which have the label i are replaced by the same tree t_i. For finite trees the procedure terminates. For example, $g(t, f(g(a,t))) = g(1, f(g(a,1)))[1_j \leftarrow t]$ in Fig. 1 represents an instance of an IO-substitution. For that, we let $s = g(x, f(g(a,x)))$ where x is a variable of rank zero and $\sigma(x) = \{t_1, t_2\}$. Then $\sigma_{\text{io}}(s) = \{g(t_1, f(g(a,t_1))), g(t_2, f(g(a,t_2)))\}$. For infinite trees, the recursion need not to stop, but running it for n recursive calls defines a Cauchy-sequence in an appropriate complete metric space $T_\perp(\Sigma \cup \mathcal{X} \cup H)$ where $\perp$ plays the role of undefined if the procedure cannot select a tree because the corresponding set $\sigma(x)$ happens to be empty.

As explained above, the feature of IO is that every leaf i_j in $t \in \sigma(x)$ labeled with a hole i is substituted with the same tree $t_i \in \sigma_{\text{io}}(s_i)$. If we remove this restriction (that is: in Eq.(1) we replace $t[i_j \leftarrow t_i]$ by $t[i_j \leftarrow t_{i_j}]$), then we obtain the OI-substitution $\sigma_{\text{oi}}(s)$ which can be much larger than $\sigma_{\text{io}}(s)$. For example, for $s = g(x, f(g(a,x)))$ and $\sigma(x) = \{t_1, t_2\}$ we obtain

$$\sigma_{\text{oi}}(s) = \{g(t_i, f(g(a, t_j))) \mid i, j \in \{1,2\}\} \,.$$

Clearly, $\sigma_{\text{io}}(s) = \sigma_{\text{oi}}(s)$ if σ is a partial homomorphism. Thus, for a partial homomorphism h we may write $h(s)$ without risking ambiguity. Another situation where $\sigma_{\text{io}}(s) = \sigma_{\text{oi}}(s)$ holds appears when duplications of holes do not appear. *Duplication mode* means that there are some x and $t \in \sigma(x)$ where a hole $i \in H$ appears at least twice in t. Duplications cannot appear in the traditional framework of words, but "duplication" is a natural and useful concept for trees.

Allowing duplications complicates the situation because it might happen that $\sigma_{\text{io}}(L)$ is not regular, although L and σ are regular. Actually, this may happen even if σ is defined by a homomorphism $h : \mathcal{X} \to T_{\text{fin}}(\Sigma \cup H)$. Here, and in the following, the notation $T_{\text{fin}}(\Sigma \cup H)$ refers to the subset of finite trees in $T(\Sigma \cup H)$. Examples of a homomorphism h where $h(L)$ is not regular are easy to construct. The classical example is $L = \{x^n(a) \mid n \in \mathbb{N}\}$ and $h(x) = f(1,1)$. Then $h(L)$ is not regular. The corresponding trees of height 3 are depicted in Fig. 2. This led to the *HOM-problem.* The inputs are a homomorphism h and a regular tree language L. The question is whether $h(L)$ is regular. The problem is decidable in the setting of finite trees by [14]. It is DEXPTIME-complete by [9].

Most of our work deals with regular tree languages. However, once the results are established for regular sets, we can easily push the results further to some border of decidability.

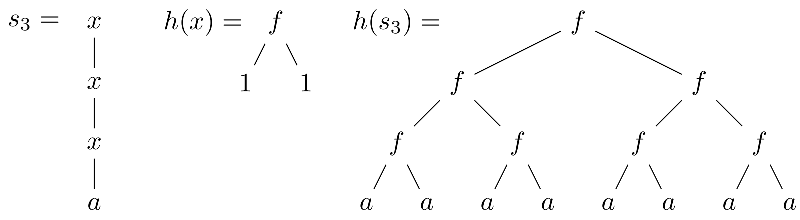

Figure 2 $L = L^*(a)$ is regular, but $h(L)$ is not.

We consider a class $\mathcal{C}$ of tree languages such that on input $L \in \mathcal{C}$ and a regular tree language K, the emptiness problem $L \cap K$ is decidable. For example, in the word case, the class $\mathcal{C}$ can be defined by the class of context-free languages, and then Conway's result for finite words still holds if L is context-free and R is regular.

We are now ready to formulate our main result. It is Thm. 30 and can be rephrased in two items. First, the following decision problem is decidable

- Input: Regular substitutions σ_1, σ_2 and tree languages $L \subseteq T(\mathcal{X})$, $R \subseteq T(\Sigma)$ such that R is regular and $L \in \mathcal{C}$.
- Question: Is there some substitution $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H) \setminus H}$ satisfying both, $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma_1 \leq \sigma \leq \sigma_2$?

Second, we can effectively compute the set of maximal substitutions σ satisfying $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma_1 \leq \sigma(x) \leq \sigma_2$. It is a finite set of regular substitutions.

Thanks to the constraint $\sigma_1 \leq \sigma(x) \leq \sigma_2$, a result for $L \subseteq T(\mathcal{X})$ implies the same result for $L \subseteq T(\Sigma \cup \mathcal{X})$. This will be become clear later.

We included some additional material in the appendix. There are three subsections. Sec. 8.1 is a kind of self-contained proof of Conway's results for finite and infinite words. This might serve as in introduction to deal with finite and infinite trees, too. Sec. 8.2 defines the outside-in extension σ_{oi} beyond what we need for the main body of the paper. The section might serve as a reference for future work to deal with OI rather than IO. Sec. 8.3 explains the standard topological notion of *quotient metric* which happens to be a metric in our setting, but denotes a pseudo-metric, in general.

1.1 Roadmap to prove our main result: Thm. 30

Our proof is a slow but essentially self-contained journey. The proof is designed to be accessible for readers who are familiar with basic results about metric spaces and regular languages over trees. In particular, we don't rely on the theory of monads[3]. The categorial theory of monads leads to general notion of *syntactic algebra*. For finite trees we refer to the arXiv-paper of Bojańczyk [3]. For infinite trees this was worked out by Blumensath [2]. In our case we use nondeterministic finite (top-down) parity-tree automata to define an appropriate *congruence* of finite index. This is conceptually simple but there is no free lunch: the index of the congruence might be not the smallest possible one[4].

[3] Conway's result for restricted to finite trees can be derived from the results presented in [3], too. Personal communication Mikołaj Bojańczyk, 2019.

[4] With respect to worst case complexity this is more an advantage than a problem. "Generically" it is debatable whether it makes sense to spend any efforts in computing the optimal index.

Having any convenient notion of congruence, the next step is to define $\sigma_{\mathrm{io}}(s)$ for finite and infinite trees such that $\sigma_{\mathrm{io}}(s)$ is indeed the intended fixed point for Eq.(1). For finite trees the set $\sigma_{\mathrm{io}}(s)$ can be defined by induction on the size of s. Then $\sigma_{\mathrm{io}}(s)$ becomes the unique least fixed point of Eq.(1) satisfying $\sigma_{\mathrm{io}}(x) = \sigma(x)$ for all symbols of rank zero. For infinite trees the definition is more subtle. Given a tree s and a substitution σ we introduce a notion of *choice function.* That is a function $\gamma : \mathrm{Pos}(s) \to T(\Sigma \cup H) \cup \{\perp\}$ where $\mathrm{Pos}(s)$ is the set of positions (that is: vertices) of s such that the following holds. If $u \in \mathrm{Pos}(s)$ is labeled by x, then γ selects a tree $\gamma(u) \in \sigma(x)$. If $\sigma(x) = \emptyset$, then $\gamma(u)$ is not defined which is denoted as $\gamma(u) = \perp$. To each choice function we will associate a Cauchy sequence $\gamma_n(s)$ in some complete metric space $T_\perp(\Sigma \cup \mathcal{X} \cup H)$; and we let $\gamma_\infty(s) = \lim_{n\to\infty} \gamma_n(s)$ be its limit. We can think of the space $T_\perp(\Sigma \cup \mathcal{X} \cup H)$ as a union of the usual Cantor space $T(\Sigma \cup \mathcal{X} \cup H)$ together with an isolated point $\perp$ which has distance 1 to every other point. Then we define

$$\sigma_{\mathrm{io}}(s) = \{\gamma_\infty(s) \mid \ \gamma \text{ is a choice function for } s \text{ and } \gamma_\infty(s) \neq \perp\}\,. \tag{2}$$

It turns out that this definition coincides with the natural definition for finite trees and it satisfies Eq.(1), too.

Another crucial step on the road to show Thm. 30 is the following result. If σ and R are regular, then the "inverse image" ${\sigma_{\mathrm{io}}}^{-1}(R) = \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{\mathrm{io}}(s) \subseteq R\}$ is a regular set of trees. In order to prove that fact we use two well-known results. First, the class of regular tree languages can be characterized by alternating parity-automata, and the semantics of these automata can be defined by parity-games [21, 22, 5]. Second, parity-games are determined and have positional (= memoryless) winning strategies, [17].

2 Notation and preliminaries

We let $\mathbb{N} = \{0, 1, \ldots\}$ denote the set of natural numbers, $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$, and $\mathbb{N}_+^*$ to be the monoid of finite sequences of natural numbers with operation "." and the neutral element ϵ. For $r \in \mathbb{N}$ we let $[r] = \{1, \ldots, r\}$. We write 2^S for the power set of S, and identify every element $x \in S$ with the singleton $\{x\}$.

A *rooted tree* is a nonempty, connected, and directed graph $t = (V, E)$ with vertex set V and without multiple edges such that there is exactly one vertex, the *root*, without any incoming edge and all other vertices have exactly one incoming edge. As a consequence, for every vertex $v \in V$ there is exactly one directed path from the root to v. Since there are no multiple edges we assume without restriction $E \subseteq \{(u, v) \mid u, v \in V, u \neq v\}$. If $(u, v) \in E$ is an edge, then we say that v is a *child* of u, and u is the *parent* of v. A *leaf* of t is a vertex without any children. Throughout, we restrict ourselves to directed graphs where the set of edges is (at most) countable. Hence, it is possible to encode the vertex set of a rooted tree as a subset of *positions* $\mathrm{Pos}(t) \subseteq \mathbb{N}_+^*$ satisfying the following conditions: $\varepsilon \in \mathrm{Pos}(t)$, and if $u.j \in \mathrm{Pos}(t)$, then both $u \in \mathrm{Pos}(t)$ and $u.i \in \mathrm{Pos}(t)$ for all $1 \le i \le j$. Using this, we have $\mathrm{root}(t) = \varepsilon$ and edge set $\{(u, u.i) \mid u, u.i \in \mathrm{Pos}(t)\}$. We are mainly interested in *ordered* trees: these are rooted trees where the children of a node are equipped with a "left-to-right" ordering. In such a case the "left-to-right breadth-first" ordering on position can be represented by the length-lexicographical ordering on $\mathbb{N}_+^*$. The *size* of a tree t is the cardinality of $\mathrm{Pos}(t)$. The *level* of a vertex u is the length of the longest directed path starting at the root and ending in u. Dually, the *height* of a vertex u is the length of the longest directed path starting at u. The *height of a tree* is the height of ε. Leaves have height 0.

Typically, and actually without restriction, every position in a tree has a label in some set Ω. Such a tree t can be defined therefore through a mapping $t : \mathrm{Pos}(t) \to \Omega$ where $\mathrm{Pos}(t)$

is the set of positions (or vertices) of the tree and $t(u) \in \Omega$ is the label of the position u. The set of all trees is denoted by $T(\Omega)$. Its subset of finite trees is denoted by $T_{\text{fin}}(\Omega)$.

Let $t, t' \in T(\Omega)$ and $u \in \text{Pos}(t)$. We define the trees $t|_u$ and $t[u \leftarrow t']$ as usual:

$$\text{Pos}(t|_u) = \left\{v \in \mathbb{N}_+^* \,\middle|\, u.v \in \text{Pos}(t)\right\} \text{ with labeling } t|_u(v) = t(u.v),$$
$$\text{Pos}(t[u \leftarrow t']) = \{u.u' \mid u' \in \text{Pos}(t')\} \cup \{v \in \text{Pos}(t) \mid u \text{ is not a prefix of } v\},$$
$$t[u \leftarrow t'](u.u') = t'(u') \text{ and } \; t[u \leftarrow t'](v) = t(v) \text{ if } u \text{ is not a prefix of } v.$$

A *ranked alphabet* is a nonempty finite set of labels Ω with a *rank* function $\text{rk} : \Omega \to \mathbb{N}$. We assume implicitly that $T(\Omega)$ is represented by a the set of *terms* over Ω, that is, ordered trees $t \in T(\Omega)$ which satisfy the following additional constraint:

- $\forall u \in \text{Pos}(t)$, if $t(u) = x$ and $\text{rk}(x) = n$, then $\{i \in \mathbb{N} \mid u.i \in \text{Pos}(t)\} = [n]$.

Henceforth, if not otherwise specified, we let $\Omega = \mathcal{X} \cup \Sigma \cup H$ be a finite ranked alphabet consisting of three sets: $\mathcal{X}$ is the set of *variables*, Σ is the set of *function symbols*, H is the set of *holes*. We require $(\mathcal{X} \cup \Sigma) \cap H = \emptyset$ but $\mathcal{X} \cap \Sigma \neq \emptyset$ is not forbidden. Symbols $a \in \Sigma$ with $\text{rk}(a) = 0$ are called *constants*. Holes are not constants, but they have rank 0, too. For simplicity, we assume $H = [r_{\max}]$ where $r_{\max} \in \mathbb{N}$ satisfies $\text{rk}(x) \leq r_{\max}$ for all $x \in \Sigma \cup H \cup \mathcal{X}$. We will also use the standard notation of terms to denote trees: $x(s_1, \ldots, s_r)$ represents the term s with $s(\varepsilon) = x$ and $s|_i = s_i$ for all $i \in [\text{rk}(x)]$.

Given a tree $t \in T(\Omega)$ we denote by $\text{leaf}_i(t)$ the set of leaves which are labeled by the hole $i \in H$. The length-lexicographical ordering of positions induced by $\mathbb{N}_+^*$ is a well-order. In particular, if $\emptyset \neq L_i \subseteq \text{leaf}_i(t)$, then there is a unique minimal element $\min(L_i)$ in that ordering: it is the leaf in L_i of minimal distance to the root and among these leaves of L_i it is the leftmost. Using this ordering we can speak about the j-th position in n $\text{leaf}_i(t)$. Phrased differently for each $i \in H$ there is a downward-closed subset $J_i \subseteq \mathbb{N}$ such that $\text{leaf}_i(t) = \{i_j \mid j \in J_i\}$ with $i_j \leq i_k \iff j \leq k$. Henceforth, the notation i_j is reserved to denote an element in $\text{leaf}_i(t)$.

The term notation is also convenient for a concise notation of infinite trees using fixed point equations. For example, let $f \in \Sigma$ be a function symbol of rank 2, then there is exactly one tree $t \in T(\Sigma \cup \{1\})$ which satisfies the equation $t = f(t, 1)$. It is depicted in Fig. 3. The set $\text{leaf}_1(t)$ is the set of all leaves. There is no leftmost leaf, but a rightmost leaf which is in turn the first one in the length-lexicographical ordering.

Figure 3 The tree $t = f(t, 1)$ has infinitely many holes labeled by 1, but no leftmost hole. The set $\text{Pos}(t) \subseteq \mathbb{N}_+^*$ is depicted on the right. Following our convention, the subset $\text{leaf}_1(t) \subseteq \text{Pos}(t)$ is written as $\{1_1, 1_2, 1_3, \ldots\}$.

Suppose that sets $T_x \subseteq T(\Sigma \cup [r])$ and $T_i \subseteq T(\Sigma)$ for $i \in [r]$ are defined. Then we define the set $T_x[i_j \leftarrow T_{i_j}] \subseteq T(\Sigma)$ as the union over all trees $t_x[i_j \leftarrow t_{i_j}]$ where $t_x \in T_x$ and $t_{i_j} \in T_{i_j}$. This is explained in more detail in Sec. 2.1. For example, $t_x[i_j \leftarrow t_{i_j}]$ corresponds to an "OI-mode", in a more restricted "IO-mode" we require additionally $t_{i_j} = t_{i_k}$ for all j, k.

2.1 Substitutions: outside-in and inside-out for finite trees

The ranked alphabet $\Sigma \cup \mathcal{X}$ contains function symbols and variables. As mentioned in the introduction we allow $\Sigma \cap \mathcal{X} \neq \emptyset$. In the following, if $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)}$ is a mapping which is specified on the set of variables, then we extend it to a mapping $\sigma : \Sigma \cup \mathcal{X} \to 2^{T(\Sigma \cup H)}$ by letting

$$\sigma(f) = \{f(1, \ldots, \mathrm{rk}(f))\} \text{ for all } f \in H \cup \Sigma \setminus \mathcal{X}. \tag{3}$$

We say that a mapping $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)}$ is a *substitution* if σ satisfies the following property

$$\sigma(x) \subseteq T(\Sigma \cup [\mathrm{rk}(x)]) \setminus H \text{ for all } x \in \mathcal{X}. \tag{4}$$

Note that (3) and (4) together imply that $t \in T(\Sigma \cup [\mathrm{rk}(x)]) \setminus H$ for all $t \in \sigma(x)$ and for all $x \in \Sigma \cup \mathcal{X}$. For all elements $x \in \Sigma \cup \mathcal{X}$ of rank zero we have $\sigma(x) \subseteq T(\Sigma)$. The set of substitutions is a partial order by letting $\sigma \leq \sigma'$ if $\sigma(x) \subseteq \sigma'(x)$ for all $x \in \mathcal{X}$. A substitution σ is called a *homomorphism* (resp. *partial homomorphism*) if $|\sigma(x)| = 1$ (resp. $|\sigma(x)| \leq 1$) for all $x \in \mathcal{X}$. Since we identify elements and singletons we can also say that a partial homomorphism is specified by a mapping $\sigma : \mathcal{X} \to T(\Sigma \cup H) \setminus H$. Homomorphisms σ such that $\sigma(x) \notin H$ for all $x \in \mathcal{X}$ are called *non-erasing* by Courcelle in [8, page 117]. The paper of Courcelle treats homomorphisms, only.

Recall the following notation. If $t \in T(\Sigma \cup H)$, then $t[i_j \leftarrow t_{i_j}]$ denotes the tree produced from s by replacing every position $i_j \in \mathrm{leaf}_i(s)$ with the tree t_{i_j}. Our goal is to extend σ to a mapping from $T(\Sigma \cup \mathcal{X} \cup H)$ to $2^{T(\Sigma \cup H)}$ such that $T(\Sigma \cup \mathcal{X})$ is mapped to $2^{T(\Sigma)}$. In this section we restrict ourselves to map finite trees to subsets of $T(\Sigma \cup H)$. (These subsets may contain infinite trees, too.) The general definition for an extension including infinite trees relies on a notion of "choice functions" and it is postponed to Sec. 2.4.

According to [11, 12] there are two natural ways to extend a substitution σ to $T_{\mathrm{fin}}(\Sigma \cup \mathcal{X})$: *outside-in* (*OI* for short) and *inside-out* (*IO* for short). The corresponding notations are σ_{oi} and σ_{io} respectively.

For a finite tree $s = x(s_1, \ldots, s_r) \in T_{\mathrm{fin}}(\mathcal{X})$, the sets of trees $\sigma_{\mathrm{io}}(s_i) \subseteq \sigma_{\mathrm{oi}}(s_i) \subseteq T(\Sigma)$ are defined by induction on the size of s. For $r = 0$ we let $\sigma_{\mathrm{oi}}(s) = \sigma_{\mathrm{io}}(s) = \sigma(x) \subseteq T(\Sigma)$. For $r \geq 1$, sets $\sigma_{\mathrm{io}}(s_i) \subseteq \sigma_{\mathrm{oi}}(s_i) \subseteq T(\Sigma)$ are defined by induction for all $i \in [r]$. Hence, we can define

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

$$\sigma_{\mathrm{oi}}(s) = \left\{t_x[i_j \leftarrow t_{i_j}] \,\middle|\, t_x \in \sigma(x) \wedge t_{i_j} \in \sigma_{\mathrm{oi}}(s_i)\right\} \tag{6}$$

It may happen that $\sigma_{\mathrm{oi}}(s) = \emptyset$, for example if $\sigma(x) = \emptyset$ (which is not forbidden). It also happens if $\sigma(x) \neq \emptyset$ and $\mathrm{leaf}_i(s) \neq \emptyset$ but $\sigma_{\mathrm{oi}}(s_i) = \emptyset$ for some $i \in [r]$. In all other cases we see (by induction) that $\sigma_{\mathrm{oi}}(s) \neq \emptyset$. Moreover, $\sigma_{\mathrm{oi}}(s) \neq \emptyset$ implies $\sigma_{\mathrm{io}}(s) \neq \emptyset$.

It is clear that for every tree $s \in T_{\mathrm{fin}}(\mathcal{X})$ we have $\sigma_{\mathrm{io}}(s) \subseteq \sigma_{\mathrm{oi}}(s) \subseteq 2^{T(\Sigma)}$. There is more flexibility in OI than in IO because positions $i_j \neq i_k$ of a hole $i \in H$ may be substituted with σ_{oi} by different trees t_{i_j} and t_{i_k}, whereas with σ_{io} they are substituted by the same tree $t_{i_j} = t_{i_k}$. This means the i-th child of a position u in s is duplicated since there is some $t \in \sigma(s(u))$ where $|\mathrm{leaf}_i(t)| \geq 2$. Hence, $\sigma_{\mathrm{io}}(s) \subsetneq \sigma_{\mathrm{oi}}(s)$ is possible because of "duplication". Fig. **??** depicts "the" smallest example for $\sigma_{\mathrm{io}}(s) \neq \sigma_{\mathrm{oi}}(s)$. Here, x is a variable of rank 1 and z is a variable of rank 0 (playing the role of "end-of-file"). Note that in this situation (with $\sigma(x) = f(1,1)$ and $\sigma(z) = \{a, b\}$) neither $\sigma_{\mathrm{io}}(x^*(z))$ nor $\sigma_{\mathrm{oi}}(x^*(z))$ is regular, since every tree in $\sigma_{\mathrm{io}}(x^n z)$ and in $\sigma_{\mathrm{oi}}(x^n z)$ is a full binary tree with 2^n leaves. Reading the labels of the leaves from left-to-right reveals the difference. For $n \geq 1$ and $a \neq b$ the "leaf-language" of $\sigma_{\mathrm{io}}(x^n z)$ is the two-element set $\{a^{2^n}, b^{2^n}\}$ whereas the "leaf-language" of $\sigma_{\mathrm{io}}(x^n z)$ has 2^n elements. It is equal to $\{a, b\}^{2^n}$.

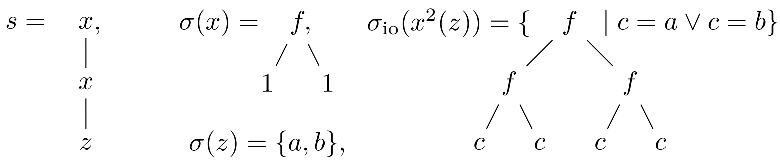

Figure 4 A duplication of the hole leads to $|\sigma_{\text{io}}(x^n(z))| = |\{a, b\}|$ and $|\sigma_{\text{oi}}(x^n(z))| = |\{a, b\}|^{n+1}$.

A modification of the example above with $\Sigma = \{f, a\}$, $H = \{1\}$, and $\mathcal{X} = \{x, z\}$ leads to a striking situation where $\sigma_{\text{io}}(x^*(z))$ is not regular but $\sigma_{\text{oi}}(x^*(z))$ is the set of all finite trees over Σ although $\sigma(z) = \emptyset$. Define $\sigma(x) = \{f(1, 1), a\}$ and $\sigma(z) = \emptyset$. Then $\sigma_{\text{io}}(x^n(z))$ is the set of all full binary trees with 2^m leaves for all $0 \leq m < n$. The set $\sigma_{\text{io}}(x^3(z))$ is depicted in Fig. 5. All leaves are labeled by a and all inner nodes are labeled by f. In contrast, $\sigma_{\text{oi}}(x^n z)$ is the set of all trees with height less than n. Hence, $\sigma_{\text{oi}}(x^* z) = T_{\text{fin}}(\Sigma)$.

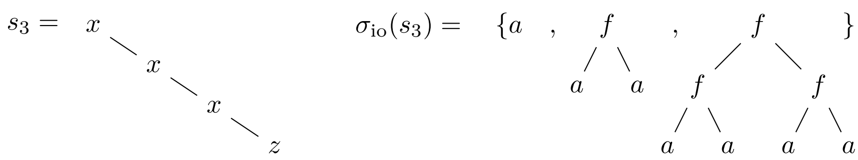

Figure 5 $L = x^*(a)$ and $\sigma(x) = \{f(1, 1), a\}$. Then $\sigma_{\text{io}}(L)$ is not regular, but $\sigma_{\text{oi}}(L) = T(\{f, a\})$. The assertion is valid for every choice of $\sigma(z)$.

Yet another variant is given by $\Sigma = \{f, a, b\}$, $H = \{1\}$, and $\mathcal{X} = \{x, y\}$ where $\text{rk}(f) = 2$, $\text{rk}(x) = \text{rk}(y) = \text{rk}(a) = 1$, and $\text{rk}(b) = 0$. Let $\sigma(x) = t$ where $t = f(t, 1)$ as in Fig. 3 and $\sigma(y) = a^*(b)$. Then we obtain the following situation as depicted in Fig. 6.

Figure 6 For all $n \in \mathbb{N}$ there exists $t_n \in \sigma_{\text{io}}(xy(b))$ as shown on the left. For all sequences $(n_i)_{i \in \mathbb{N}}$ there is a tree $t \in \sigma_{\text{oi}}(xy(b)$ as on the right. The set $\sigma_{\text{oi}}(xy(b))$ is regular, but $\sigma_{\text{io}}(xy(b))$ is not regular.

We will show that σ_{io} can be extended to $T(\Sigma \cup \mathcal{X})$ such that equation (5) still holds. The extension is based on defining two auxiliary concepts: complete metric spaces and choice functions. The formal definition is in Sec. 2.4. We neither need nor use the (possible) outside-in extension σ_{oi} in full generality for all trees $T(\Sigma \cup \mathcal{X})$ such that σ_{oi} satisfies Eq.(6). Details how to define σ_{oi} are therefore left to the interested reader or can be found in Sec. 8.2.

2.2 About the existence of maximal solutions

The material in this subsection is not used elsewhere as our focus in on infinite trees. For example, even in the restricted case of infinite words, Prop. 1 does not hold, in general. See Ex. 2. The importance of that example is however that the *regularity* hypothesis on R cannot be removed from our main results. Prop. 3 shows a positive result for the outside-in extension σ_{oi}. It is more restrictive than Prop. 1 because it applies to substitutions σ where no $\sigma(x)$ contains a tree with infinitely many holes. Ex. 4 shows that Prop. 3 does not carry over to the case where this restriction on σ is removed.

Throughout this section $T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$ and $T_{H\text{-fin}}(\Sigma \cup H)$ denote the set of trees with only a finite number of occurrences of variables (resp. holes). Moreover, for $s \in T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$ we use also use Eq.(5) to define $\sigma_{\mathrm{io}}(s)$ by induction on the sum $\sum_{u \in \{\mathrm{Pos}(s) \mid s(u) \in \mathcal{X}\}} \mathrm{level}(u)$. A similar induction shows that we can use Eq.(6) to define $\sigma_{\mathrm{oi}}(s)$ if $s \in T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$ and $\sigma(x) \subseteq T_{H\text{-fin}}(\Sigma \cup H) \setminus H$ for all $x \in \mathcal{X}$. The formula looks more complicated but it follows the same scheme:

$$\sum_{u \in \{\mathrm{Pos}(s) \mid s(u) \in \mathcal{X}\}} \min\{ \sum_{i_j \in \{\mathrm{Pos}(t) \mid i_j \in \mathrm{leaf}_i(t) \wedge i \in H\}} \mathrm{level}(i_j) \mid t \in \sigma(s(u))\}. \tag{7}$$

▶ **Proposition 1.** *Let $L \subseteq T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$ and $R \subseteq T(\Sigma)$. Then the following statement holds for the inside-out extension σ_{io} and for both $\# \in \{\subseteq, =\}$. For every substitution $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H}$ such that $\sigma_{\mathrm{io}}(L) \# R$ there exists a maximal substitution σ' such that $\sigma \leq \sigma'$ and $\sigma'_{\mathrm{io}}(L) \# R$.*

Proof. Let us show that $\leq$ is an *inductive ordering* on the set of σ such that $\sigma_{\mathrm{io}}(L) \# R$. By Zorn's lemma, it follows that every solution is upper-bounded by some maximal solution. To see that $\leq$ is an inductive ordering, let K be a nonempty totally ordered index set $\{\sigma^{(k)} : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H} \mid k \in K\}$ of substitutions satisfying $\sigma^{(k)} \leq \sigma^{(\ell)} \iff k \leq \ell$ and

$$\{\sigma^{(k)} : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H} \mid k \in K\} \subseteq \{\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H} \mid \sigma_{\mathrm{io}}(L) \# R\}.$$

Let us show, by an induction on the height of trees $s \in T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$, that if $s \in L$ and $t \in \sigma'(s)$, then $t \in \sigma^{(k)}(s)$ for some $k \in \mathbb{N}$. For that we fix any index $k_0 \in K$. Now, let $s = x(s_1, \ldots, s_r) \in L$ and $t = t_x[i_j \leftarrow t_i] \in \sigma'_{\mathrm{io}}(s)$ such that $t_x \in \sigma'(x)$ and $t_i \in \sigma'_{\mathrm{io}}(s_i)$. By induction we may assume that there are indices k_i such that $t_i \in \sigma_{\mathrm{io}}^{(k_i)}(s_i)$ for all $1 \leq i \leq \mathrm{rk}(x)$. (This is true for $\mathrm{rk}(x) = 0$.) Since $t_x \in \sigma'(x)$, there is some index k_x such that $t_x \in \sigma^{(k_x)}(x)$. Let $k_t = \max\{k_x, k_i \mid 0 \leq i \leq \mathrm{rk}(x)\}$. Then we have $k_t \geq k_0$ and $t \in \sigma_{\mathrm{io}}^{(k_t)}(s)$: the induction step is achieved. Therefore, $\sigma'_{\mathrm{io}}(s) \subseteq R$.

Finally, if $\sigma_{\mathrm{io}}^{(k)}(L) = R$ for all $k \in K$, then for each $t \in R$ there is some $s \in L$ such that $t \in \sigma_{\mathrm{io}}^{(k_t)}(s)$. This implies $R \subseteq \bigcup\{\sigma_{\mathrm{io}}^{(k)}(s) \mid s \in L \wedge k \in \mathbb{N}\} = \bigcup\{\sigma'_{\mathrm{io}}(s) \mid s \in L\} \subseteq R$. Hence, $\bigcup\{\sigma'_{\mathrm{io}}(s) \mid s \in L\} = R$: every solution is upper-bounded by some *maximal* solution. ◀

Ex. 2 shows that even in the case where L is a regular language of infinite words, the statement of Prop. 1 might fail if R is not regular.

▶ **Example 2.** Consider $\Sigma = \{a, b\}$ and $\mathcal{X} = \{x\}$ where all three symbols have rank one. Hence, $H = \{1\}$ and an infinite tree over $\Sigma \cup \mathcal{X}$ encodes an infinite word in $(\Sigma \cup \mathcal{X})^\omega$ and vice versa. We let $L = (ax)^\omega$, and $R = \left\{u \in \Sigma^\omega \,\middle|\, \exists k \geq 1, b^k \text{ is not factor of } u\right\}$. The set L is regular, but R is not regular! Note that we have $\sigma_{\mathrm{io}}(L) = \sigma_{\mathrm{oi}}(L)$ for every substitution $\sigma(x) \subseteq T(\Sigma \cup H) \setminus H$. There are many substitutions σ such that $\sigma_{\mathrm{io}}(L) \subseteq R$, for example $\sigma(x) = a(1)$. But there is no maximal substitution σ such that $\sigma_{\mathrm{io}}(L) \subseteq R$. Indeed, given any substitution $\sigma(x) \subseteq T_{\mathrm{fin}}(\{a, b, 1\}) \setminus H$ such that $\sigma_{\mathrm{io}}(L) \subseteq R$, we can define $n = \max\{k \geq 1 \mid b^k \text{ is a factor of } \sigma(x)\}$. Then $\sigma'(x) = \sigma(x) \cup \{b^{n+1}(1)\}$ is strictly larger than σ, and it satisfies $\sigma'_{\mathrm{io}}(L) \subseteq R$, too.

▶ **Proposition 3.** *Let $L \subseteq T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$ and $R \subseteq T(\Sigma)$ be any set of trees. Let $E \in \{\mathrm{io}, \mathrm{oi}\}$ and $\# \in \{\subseteq, =\}$. Then for every substitution $\sigma : \mathcal{X} \to 2^{T_{H\text{-fin}}(\Sigma \cup H)\setminus H}$ such that $\sigma_E(L) \# R$ there exists a maximal substitution $\sigma' : \mathcal{X} \to 2^{T_{H\text{-fin}}(\Sigma \cup H)\setminus H}$ such that $\sigma \leq \sigma'$ and $\sigma'_E(L) \# R$.*

Proof. For the inside-out extension $E = \mathrm{io}$ the assertion is a special case of Prop. 1. Thus, it is enough to consider the outside-in extension $E = \mathrm{oi}$. Similar to the proof of Prop. 1, with the help of Zorn's Lemma, it is enough to show that $\leq$ is an inductive ordering. Let K be any nonempty totally ordered index set $\{\sigma^{(k)} : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H} \mid k \in K\}$ a set of substitutions satisfying $\sigma^{(k)} \leq \sigma^{(\ell)} \iff k \leq \ell$ and

$$\{\sigma^{(k)} : \mathcal{X} \to 2^{T_{H\text{-fin}}(\Sigma \cup H)} \mid k \in K\} \subseteq \{\sigma : \mathcal{X} \to 2^{T_{H\text{-fin}}(\Sigma \cup H)\setminus H} \mid \sigma_{\mathrm{oi}}(L) \mathrel{\#} R\}.$$

Recall that $\sigma_{\mathrm{oi}}(s)$ satisfies Eq.(6) for $s \in T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$ and $\sigma : \mathcal{X} \to 2^{T_{H\text{-fin}}(\Sigma \cup H)\setminus H}$. As above, define $\sigma'(x) = \bigcup \left\{\sigma^{(k)}(x) \,\middle|\, k \in K\right\}$. Note that every $t \in \sigma'(x)$ satisfies $t \in T_{H\text{-fin}}(\Sigma \cup H) \setminus H$. We fix any index $k_0 \in K$ and we note that $\sigma'(x) = \sigma^{(k_0)}(x)$ for all $x \in \Sigma \setminus \mathcal{X}$. Now, let $s = x(s_1, \ldots, s_r) \in L$ and $t = t_x[i_j \leftarrow t_{i_j}] \in \sigma'_{\mathrm{oi}}(s)$ such that $t_x \in \sigma'(x)$ and $t_{i_j} \in \sigma'_{\mathrm{oi}}(s_i)$. We want to show the existence of some $k_0 \leq k \in K$ such that $t \in \sigma^{(k)}(s)$. We can think that t is defined by a (non-effective) recursive procedure which first initializes an index $k \in K$ by $k = k_0$. Next, on input $s = x(s_1, \ldots, s_r) \in T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$ and $k \in K$, it does the following as long as a variable $x \in \mathcal{X}$ appears somewhere in s:

- It selects a tree $t_x \in \sigma'(x)$, outputs the tree t_x and updates k to $k := \max\{k, k_x\}$ such that $t_x \in \sigma^{(k)}(x)$.
  Recall that we need the existence of updated k, only. No effectiveness is required.
- For all positions i_j where $\mathrm{leaf}_i(t_x) \neq \emptyset$, one after another, the procedure makes a recursive call with the input s_i and the current value of k.
  Recall that each $\mathrm{leaf}_i(t_x)$ is finite. Thus, we have termination.
- Delete t_x, but remember the actual value of k.

We "call" the procedure for $s \in T_{\mathcal{X}\text{-fin}}(\Sigma \cup \mathcal{X})$ and where each $t \in \sigma'(x)$ satisfies $t \in T_{H\text{-fin}}(\Sigma \cup H) \setminus H$. Therefore, the procedure terminates with a final value $k \in K$, and we define $k_t = k$. We conclude that $t \in \sigma_{\mathrm{io}}^{(k_t)}(s)$. We are done as in the proof of Prop. 1. ◀

▶ **Example 4.** Consider $\Sigma = \{f, a, b\}$ and $\mathcal{X} = \{x, y\}$ where $\mathrm{rk}(f) = 2$, $\mathrm{rk}(x) = \mathrm{rk}(y) = \mathrm{rk}(a) = 1$, and $\mathrm{rk}(b) = 0$. We are interested in the finite term $x(y(b))$. Let $\sigma(x) = t$ where $t = f(t, 1)$ (as in the example used for Fig. 6) and $\sigma(y) = a(b)$. The infinite tree $\sigma(x)$ is regular and the accepted language of a deterministic top-down tree automaton with a Büchi-acceptance condition as defined for example in [26]. $\sigma_{\mathrm{oi}}(x(y(b)))$ is a singleton which is depicted in Fig. 6 on the left with $n = 1$ and $\sigma(y) = a(b)$ is a singleton, too. Similar as in Ex. 2 we let

$$R = \left\{t \in T(\{f, a, b\}) \,\middle|\, \exists k \geq 1, a^k \text{ is not factor of any path in } t\right\}.$$

Then $\sigma_{\mathrm{oi}}(x(y(b))) \subseteq R$. However, there is no maximal substitution σ' such that $\sigma \leq \sigma'$ and $\sigma_{\mathrm{oi}}(x(y(b))) \subseteq R$. Indeed, given any such σ' there is some $n \in \mathbb{N}$ such that $\sigma''(y) = \sigma'(y) \cup \{a^n(b)\}$ is strictly larger than σ' and σ'' still satisfies $\sigma''_{\mathrm{oi}}(x(y(b))) \subseteq R$. If we define $\sigma''(y) = a^*(b)$, then $\sigma''_{\mathrm{oi}}(x(y(b)))$ is regular, but it is not a subset of R.

2.3 Complete metric spaces with $\perp$ for "undefined"

In order to handle the case that $\sigma_{\mathrm{io}}(s) = \emptyset$, it is convenient to introduce a special constant $\perp \notin T(\Omega)$ which plays the role of "undefined". The idea is that an empty set $\sigma_{\mathrm{io}}(s)$ is replaced by the singleton $\{\perp\}$ (or simply $\perp$ as we don't distinguish between elements and the corresponding singleton sets).

The set of trees $T(\Omega \cup \{\perp\})$ is a metric space by defining a distance d as follows. The distance makes sure that $T(\Omega)$ becomes a clopen (= open and closed) subspace in

$(T(\Omega \cup \{\perp\}), d)$. Hence, each of $T(\Omega)$, $\{\perp\} \cup T(\Omega)$, and their complements with respect to $T(\Omega \cup \{\perp\}) \setminus T(\Omega)$ are clopen subsets in $(T(\Omega \cup \{\perp\}), d)$.

$$d(s,t) = \begin{cases} 2^{-\inf\{|u| \mid u \in \mathrm{Pos}(s) \cap \mathrm{Pos}(t) \wedge s(u) \neq t(u)\}} & \text{if either } s,t \in T(\Omega) \text{ or both, } s,t \notin T(\Omega), \\ 1 & \text{otherwise: } s \in T(\Omega) \iff t \notin T(\Omega). \end{cases}$$

The convention is as usual $2^{-\infty} = 0$. The restriction to the clopen subspace $\{\perp\} \cup T(\Omega)$ yields a compact ultra-metric space such that

$$d(s,t) = 2^{-\inf\{|u| \mid u \in \mathrm{Pos}(s) \cap \mathrm{Pos}(t) \wedge s(u) \neq t(u)\}} \tag{8}$$

holds for all $s, t \in \{\perp\} \cup T(\Omega)$. Indeed, $d(\perp, t) = 1$ for all $t \in T(\Omega)$. The ambient metric space $(T(\Omega \cup \{\perp\}), d)$ is not complete and therefore not compact since the Cauchy-sequence $(a^n(\perp))_{n \in \mathbb{N}}$ does not have any limit in $(T(\Omega \cup \{\perp\}), d)$. Here we assume $\mathrm{rk}(a) = 1$. However, as $\perp$ represents "undefined", we wish that $\lim_{n\to\infty} a^n(\perp) = \perp$. There is a way to achieve that: we identify the clopen set $T(\Omega \cup \{\perp\}) \setminus T(\Omega)$ with the constant $\perp$. Thereby $\perp$ becomes an isolated point. To be precise, let us define the equivalence relation $\sim$ on $T(\Omega \cup \{\perp\})$ which is induced by letting $\perp \sim t$ for all $t \in T(\Omega \cup \{\perp\}) \setminus T(\Omega)$. Now, we have $a^n(\perp) \sim \perp$ for all n. Hence, the sequence $(a^n(\perp))_{n \in \mathbb{N}}$ is equal in the quotient space to the constant sequence $(\perp)_{n \in \mathbb{N}}$ which has the obvious limit $\perp$.

Having this, the natural embedding of $\{\perp\} \cup T(\Omega)$ into the quotient space $T(\Omega \cup \{\perp\})/\sim$ induces an isometry between $(T(\Omega) \cup \{\perp\}, d)$ and $(T(\Omega \cup \{\perp\})/\sim, d_\sim)$ where $d_\sim$ is the canonical quotient metric[5] on $T(\Omega \cup \{\perp\})/\sim$. This means, the equivalence class of $\perp$ has distance 1 to every other point, and the restriction to $(T(\Omega) \cup \{\perp\})$ is just the usual Cantor-space. Hence, we identify the metric spaces $(T(\Omega) \cup \{\perp\}, d)$ and $(T(\Omega \cup \{\perp\})/\sim, d_\sim)$. Thus, $T_\perp(\Omega)$ comes with two equivalent interpretations: either as $(T(\Omega) \cup \{\perp\}, d)$ or as $(T(\Omega \cup \{\perp\})/\sim, d_\sim)$; and every equivalence class in $T(\Omega \cup \{\perp\})/\sim$ is represented by a single tree in $T(\Omega) \cup \{\perp\}$ such that the constant $\perp$ represents all trees in $T(\Omega \cup \{\perp\}) \setminus T(\Omega)$. As a consequence, whenever $s = (s_1, s_2, \ldots)$ is an infinite sequence with $s_i \in T(\Omega \cup \{\perp\})$ and $s' = (s'_1, s'_2, \ldots)$ is the derived sequence where every s_i is replaced by $\perp$ if $s_i \notin T(\Omega)$, then the sequence s is a Cauchy-sequence in $(T(\Omega \cup \{\perp\})/\sim, d_\sim)$ if and only if s' is a Cauchy-sequence in $(T(\Omega) \cup \{\perp\}, d)$.

2.4 Choice functions and the definition of σ_{io} for infinite trees

A *choice function* for $s \in T(\mathcal{X})$ is a mapping $\gamma : \mathrm{Pos}(s) \to \{\perp\} \cup T(\Sigma \cup H)$ such that

- $\gamma(x) \in \{\perp\} \cup T(\Sigma \cup [\mathrm{rk}(x)]) \setminus H$ for all $x \in \mathcal{X}$.

For $u \in \mathrm{Pos}(s)$ we let $\gamma|_u : \mathrm{Pos}(s|_u) \to \{\perp\} \cup T(\Sigma \cup H)$ be the mapping defined by $\gamma|_u(u') = \gamma(u.u')$. Note that $\gamma|_u$ is a choice function for the subtree $s|_u$ whenever $u \in \mathrm{Pos}(s)$ and γ is a choice function for s. For every $s \in T(\Sigma \cup \mathcal{X})$ and choice function γ for s, we define the sequence of trees $(\gamma_n(s))_{n \in \mathbb{N}}$ in $T(\Sigma \cup H \cup \{\perp\})$ as follows:

$$\gamma_0(s) = \gamma(\varepsilon) \quad \text{and} \quad \gamma_n(s) = \gamma(\varepsilon)[i_j \leftarrow (\gamma|_i)_{n-1}(s|_i)] \text{ if } n > 0.$$

$(\gamma_n(s))_{n \in \mathbb{N}}$ is a Cauchy sequence in the complete metric space $T_\perp(\Sigma \cup H, d)$. (Recall that it is the space where all trees with an occurrence of $\perp$ are identified with the constant $\perp$.) Therefore $\lim_{n\to\infty} \gamma_n(s) \in \{\perp\} \cup T(\Sigma \cup H)$ always exists. Moreover, the fact that choice functions don't map tree positions to holes implies that $\lim_{n\to\infty} \gamma_n(s) \in \{\perp\} \cup T(\Sigma)$.

[5] We don't need the general definition of *quotient metric* which is only a pseudo metric, in general.

▶ **Definition 5.** *Let $\sigma : \mathcal{X} \to 2^{T(\Sigma\cup H)\setminus H}$ be a substitution and $s \in T(\Sigma \cup \mathcal{X})$. Then we define*

1. *The tree $\gamma_\infty(s) = \lim_{n\to\infty} \gamma_n(s) \in \{\perp\} \cup T(\Sigma)$.*
2. *The set $\Gamma(\sigma, s)$ of choice functions for s satisfying for all $u \in \mathrm{Pos}(s)$ with $x = s(u)$ the following: if $\sigma(x) \neq \emptyset$, then $\gamma(u) \in \sigma(x)$, otherwise $\gamma(u) = \perp$.*
3. *The set $\sigma_{\mathrm{io}}(s) = \{\gamma_\infty(s) \mid \gamma \in \Gamma(\sigma, s)\}$.*

Note that $\sigma_{\mathrm{io}}(s)$ need not to be closed. Indeed, let $s = x(s_1)$ where $x \in \mathcal{X}$ is a variable of rank 1 and s_1 is any (finite or infinite) tree in $T(\mathcal{X}\cup\Sigma\cup H\cup\{\perp\})$. Let $\sigma(x) = \{a^n(b) \mid n \in \mathbb{N}\}$ with a constant b and $\mathrm{rk}(a) = 1$. Then $\sigma_{\mathrm{io}}(s) = \{a^n(b) \mid n \in \mathbb{N}\}$ is a finite subset of $T(\{a, b\})$ which is open but not closed because $a^\omega \notin \sigma_{\mathrm{io}}(s)$.

▶ **Proposition 6.** *Let $\sigma : \mathcal{X} \to 2^{T(\Sigma\cup H)\setminus H}$ be a substitution, $s = x(s_1, \ldots, s_r) \in T(\Sigma \cup \mathcal{X})$ be a tree, and $\gamma \in \Gamma(\sigma, s)$ be a choice function. Then $\gamma_\infty(s) = \gamma(\varepsilon)[i_j \leftarrow (\gamma_{|i})_\infty(s_i)]$.*

Proof. Since $\gamma_\infty(s) = \lim_{n\to\infty} \gamma_n(s)$ and $(\gamma_{|i})_\infty(s_i) = \lim_{n\to\infty}(\gamma_{|i})_n(s_i)$ for all i, we obtain

$$\begin{aligned}
\gamma(\varepsilon)[i_j \leftarrow (\gamma_{|i})_\infty(s_i)] &= \gamma(\varepsilon)[i_j \leftarrow \lim_{n\to\infty}(\gamma_{|i})_n(s_i)] \\
&= \gamma(\varepsilon)[i_j \leftarrow \lim_{n\to\infty}(\gamma_{|i})_{n-1}(s_i)] \\
&= \lim_{n\to\infty} \gamma(\varepsilon)[i_j \leftarrow (\gamma_{|i})_{n-1}(s_i)] \\
&= \lim_{n\to\infty} \gamma_n(s) = \gamma_\infty(s).
\end{aligned}$$

◀

▶ **Corollary 7.** *Let $s = x(s_1, \ldots, s_r) \in T(\Sigma \cup \mathcal{X})$ be a tree. Then we have $\sigma_{\mathrm{io}}(s) = \{t_x[i_j \leftarrow t_i] \mid t_x \in \sigma(x) \wedge t_i \in \sigma_{\mathrm{io}}(s_i)\}$. In particular, for finite trees the new definition of $\sigma_{\mathrm{io}}(s)$ in Def. 5 agrees with the earlier one given in Eq.(5).*

Proof. We use the notation as given in Def. 5.

$$\begin{aligned}
\sigma_{\mathrm{io}}(s) &= \bigcup \left\{\gamma(\varepsilon)[i_j \leftarrow (\gamma_{|i})_\infty(s_i)] \,\middle|\, \gamma \in \Gamma(\sigma, s)\right\} && \text{by Prop. 6} \\
&= \left\{t_x[i_j \leftarrow (\gamma_{|i})_\infty(s_i)] \,\middle|\, t_x \in \sigma(x), \gamma_{|i} \in \Gamma(\sigma, s_{|i})\right\} && \text{trivial} \\
&= \{t_x[i_j \leftarrow t_i] \mid t_x \in \sigma(x) \wedge t_i \in \sigma_{\mathrm{io}}(s_i)\} && \text{by Def. 5}
\end{aligned}$$

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

Every infinite tree s can be written as a limit $\lim_{n\to\infty} s_n$ where s_n are finite trees. The following examples show that Cauchy-sequences $(t_n)_{n\in\mathbb{N}}$ with $t_n \in \sigma_{\mathrm{io}}(s_n)$ need not converge to any tree in $\sigma_{\mathrm{io}}(s)$. In these examples x is a variable of rank one, $\Sigma = \{f, a, b\}$ with a constant b, $\mathrm{rk}(a) = 1$, $\mathrm{rk}(f) = 2$, and where only the first hole $1 \in H$ is used. In the first example we have $s = s_n$ for all n.

▶ **Example 8.** Consider $s = x(b)$ and $\sigma(x) = \{a^n(1) \mid n \in \mathbb{N}\}$. Then $s = \lim_{n\to\infty} s_n$ where $s_n = s$. We define $t_n = \gamma^{(n)}(s) = a^n(b)$ where $\gamma^{(n)} \in \Gamma(s, \sigma)$, $\gamma^{(n)}(\varepsilon) = a^n(1)$, and note that $(t_n)_{n\in\mathbb{N}}$ is a Cauchy sequence with every $t_n \in \sigma_{\mathrm{io}}(s_n)$, but

$$\lim_{n\to\infty} t_n = a^\omega \notin \sigma_{\mathrm{io}}(s) = \{a^n(b) \in \mathbb{N}\}.$$

The next example is a modification of Ex. 8 such that s becomes infinite and the sequence s_n is increasing such that $s_{n+1} = s_n[1 \leftarrow x(1)]$.

▶ **Example 9.** Let $s = f(a^\omega, x(b))$ and σ be defined by $\sigma(x) = \{a^n(1) \mid n \in \mathbb{N}\}$. Then $s = \lim_{n\to\infty} s_n$ where $s_n = f(a^n(1), x(b))$. We have $t_n = f(a^n(1), a^n(b)) \in \sigma_{\mathrm{io}}(s_n)$ since for each n we may use the choice function $\gamma^{(n)}$ with $\gamma^{(n)}(u) = a^n(b)$ where $u = 2 \in \mathrm{Pos}(s)$. The t_n's form a Cauchy-sequence with $t = f(a^\omega, a^\omega) = \lim_{n\to\infty} t_n$ but $t \notin \sigma_{\mathrm{io}}(s) = \{f(a^\omega, a^n(b)) \mid n \in \mathbb{N}\}$.

3 Regular tree languages

There are several equivalent definitions for regular languages of finite and infinite trees, see [5, 15, 24, 26]. It is well-known classical fact that the class of regular tree languages forms an effective Boolean algebra [24]. Here, we will consider regular languages of finite and infinite trees from $T(\Sigma \cup \mathcal{X} \cup H \cup \{\perp\})$. Note that for every subset $\Delta \subseteq \Sigma \cup \mathcal{X} \cup H \cup \{\perp\}$, the set $T(\Delta)$ is regular in $T(\Sigma \cup \mathcal{X} \cup H \cup \{\perp\})$. Moreover, there are isometrical embeddings of $T(\Delta) \subseteq T(\Sigma \cup \mathcal{X} \cup H \cup \{\perp\})$ and $T_\perp(\Delta) \subseteq T_\perp(\Sigma \cup \mathcal{X} \cup H)$. The element $\perp \in T_\perp(\Sigma \cup \mathcal{X} \cup H)$ is isolated since it has distance at least 1 to any other element.

3.1 Nondeterministic tree automata with parity acceptance

We use parity-NTAs (nondeterministic tree automata with a parity acceptance condition) as the basic reference to characterize regular tree languages. The parity condition is without any meaning for acceptance of finite trees. It is used to accept infinite trees, too. In our definition a parity-NTA accepts sets of finite and infinite trees. Alternating tree automata with a parity acceptance condition will be considered later.

Let Δ be a finite ranked alphabet with the rank function $\mathrm{rk} : \Delta \to \mathbb{N}$. A *parity-NTA* over Δ is specified by a tuple $A = (Q, \Delta, \delta, \chi)$ where Q is a nonempty finite set of states, δ is the *transition relation*, and $\chi : Q \to C$ is a coloring with $C = \{1, , \ldots , |C|\}$. Without restriction, we assume that $|C|$ is odd. We have

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

Thus, δ is a set of tuples $\big(p, f, p_1, \ldots , p_r\big)$ where $r = \mathrm{rk}(f)$. Note that for $r = 0$ there are no children and whether we accept f at state p depends whether or not $(p, f) \in \delta$. In order to define the acceptance of a tree we use the following.

▶ **Definition 10.** *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. That is, $\rho : \mathrm{Pos}(t) \to Q$ is a mapping such that for all $u \in \mathrm{Pos}(t)$ with $t(u) = f$ there is a transition $(p, f, p_1, \ldots , p_{\mathrm{rk}(f)}) \in \delta$ such that $\rho(u) = p$ and $\rho(u.j) = p_j$ for all $j \in [\mathrm{rk}(f)]$. (See [15] for more details.)*
  If $\rho(\varepsilon) = p$, then we say that ρ is a run of t at state p.
- *Let ρ be a run and $(u_0, u_1, u_2, \ldots)$ be a finite or infinite path in $\mathrm{Pos}(t)$ such that u_{j+1} is a child of u_j. The path is* accepting *either if it is finite and ending in a leaf or if the number $\max\{\min\{\chi\rho(u_i) \mid i \geq k\} \mid k \in \mathbb{N}\}$ is even. This means that for all infinite directed paths in the tree $\chi\rho : \mathrm{Pos}(t) \to C$, the minimal color appearing infinitely often is even.*
- *By $\mathrm{Run}_A(t, p)$ we denote the set of **accepting** runs of t at state p. (If the context to A is clear, we might simply write $\mathrm{Run}(t, p)$.)*
- *If $p \in Q$, then we let $L(A, p) = \{t \in T(\Delta) \mid \mathrm{Run}_A(t, p) \neq \emptyset\}$. It is the* accepted language of A at state p.
- *For $P \subseteq Q$ we define $L(A, P)$ by*

$$L(A, P) = \bigcap \{L(A, p) \mid p \in P\}. \tag{10}$$

The set $L(A, P) \subseteq T(\Delta)$ is called the accepted language at the set P. *The convention is as usual: $L(A, \emptyset) = T(\Delta)$.*

The following fact is well-known, see e. g. [15]. Thus, we can use the characterization in Prop. 11 as the working definition for the present paper.

▶ **Proposition 11.** *A language $L \subseteq T(\Delta)$ is regular if and only if there is a parity-NTA $A = (Q, \Delta, \delta, \chi)$ and a state $p \in Q$ such that $L = L(A, p)$.*

Since regular tree languages form an effective Boolean algebra, Prop. 11 implies that for each parity-NTA A and all subsets $P, P' \subseteq Q$ we can effectively construct a parity-NTA B with a single initial state q such that $L(B, q) = L(A, P) \setminus L(A, P')$.

4 Tasks and profiles

In this section we introduce the notions of *task* and *profile*. Throughout, we let $H = \{1, \ldots, |H|\}$ be a set of holes and Σ be a ranked alphabet such that $\mathrm{rk}(f) \le |H|$ for all $f \in \Sigma$. We also fix a parity-NTA $B = (Q, \Sigma, \delta, \chi)$ such that $L(B, p) \subseteq T(\Sigma)$ for all $p \in Q$. Every such B is extended to a parity-NTA B_H by allowing more transitions such that all holes $i \in H$ are accepted at all states. Thus, we let $B_H = (Q, \Sigma, \delta_H, \chi)$ where $\delta_H = \delta \cup (Q \times H)$. Clearly, $L(B_H, p)$ is regular and therefore the complement $T(\Sigma \cup H) \setminus L(B_H, p)$ is regular, too. For the rest of this section, a run ρ of a tree t refers to a tree $t \in T(\Sigma \cup H)$ and the NTA B_H. Thus $\mathrm{Run}(t, p) = \mathrm{Run}_{B_H}(t, p)$ if not stated explicitly otherwise. Let ρ be any run of a tree $t : \mathrm{Pos}(t) \to \Sigma$ (not necessarily accepting) and let $\alpha = (u_0, u_1, \ldots)$, $\alpha' = (u'_0, u'_1, \ldots)$ be infinite directed paths in $\mathrm{Pos}(t)$ with $\varepsilon = u_0 = u'_0$. (Recall that directed paths in trees are directed away from the root. Thus, u_{i+1} is a child of u_i and u'_{i+1} is a child of u'_i for all $i \in \mathbb{N}$.) The paths α and α' define infinite words over Σ and the run ρ defines infinite words $\rho(\alpha)$ and $\rho(\alpha')$ over Q. Here, Σ and Q are viewed as finite alphabets. Suppose each path is cut into infinitely many finite and nonempty pieces w_i and w'_i such that we can write $\rho(\alpha) = \rho(w_0)\rho(w_1)\cdots)$ and $\rho(\alpha') = \rho(w'_0)\rho(w'_1)\cdots)$ where all $\rho(w_i)$ and $\rho(w'_i)$ are in Q^+. Assume that α' satisfies the parity condition. The idea is to find a condition based on all factors $\rho(w_i)$ and $\rho(w'_i)$ such that the condition is sufficient to imply that α satisfies the parity condition, too.

Naturally, such a condition is related to colors which appear in $\chi\rho(w_i)$ and $\chi\rho(w'_i)$. More precisely, let c_i (resp. c'_i) denote the minimal color in $\chi\rho(w_i)$ (resp. $\chi\rho(w'_i)$) for $i \in \mathbb{N}$. We obtain two infinite words $(c_0, c_1, \ldots)$ and $(c'_0, c'_1, \ldots)$ over the alphabet C of colors. If we compare locally each color c_i with the corresponding color c'_i, then intuitively, with respect to the parity condition, an even color is better than an odd color. A small even color is better than a large even color. However, a large odd color is better than a small odd color. Based on that intuition, let us introduce the *best-ordering* $\preceq_{\mathrm{best}}$ on the set $\{0, \ldots, |C|\}$. Note that we explicitly include 0 which is not in $\chi(Q)$ and that $|C|$ is odd by our convention. We let

$$0 \preceq_{\mathrm{best}} 2 \preceq_{\mathrm{best}} \cdots \preceq_{\mathrm{best}} |C| - 1 \preceq_{\mathrm{best}} |C| \preceq_{\mathrm{best}} |C| - 2 \preceq_{\mathrm{best}} \cdots \preceq_{\mathrm{best}} 3 \preceq_{\mathrm{best}} 1 \tag{11}$$

Indeed, in the linear order $\preceq_{\mathrm{best}}$ all even numbers are “better” than the odd numbers. Among even numbers smaller is better. Among odd numbers larger is better.

As a consequence, if $c_i \preceq_{\mathrm{best}} c'_i$ for all $i \in \mathbb{N}$ and if the parity condition holds for α', then it holds for α.

▶ **Definition 12.** *A* task *is a tuple $(p, \psi_1, \ldots, \psi_{|H|})$ with $p \in Q$ and $\psi_i : Q \to \{0\} \cup \chi(Q)$ for $1 \le i \le |H|$. A* profile *is a set of tasks.*

The best-ordering defines a partial order on the set of tasks: we write $(p, \psi_1, \ldots, \psi_{|H|}) \le (p', \psi'_1, \ldots, \psi'_{|H|})$ if first, $p = p'$ and second, $\psi_i(q) \preceq_{\mathrm{best}} \psi'_i(q)$ for all $q \in Q$ and $i \in H$.

▶ **Definition 13.** *Let $\tau = (p, \psi_1, \ldots, \psi_{|H|})$ be a task and $t, t' \in T(\Sigma \cup H \cup \{\bot\})$ be trees.*

1. *A run $\rho \in \mathrm{Run}_{B_H}(t, p)$* satisfies *$\tau$ if the following condition holds for all leaves $i_j \in \mathrm{leaf}_i(t)$:*
   - *If $c_\rho(i_j)$ denotes the minimal color on the path from the root of t to position $i_j \in \mathrm{leaf}_i(t)$ in the tree $\chi\rho : \mathrm{Pos}(t) \to C$, then we have $c_\rho(i_j) \preceq_{best} \psi_i(\rho(i_j))$.*

   We also write $\rho \models \tau$ in that case. Moreover, we write $t \models \tau$ if there exists some $\rho \in \mathrm{Run}_{B_H}(t, p)$ such that $\rho \models \tau$.
2. *A run $\rho \in \mathrm{Run}_{B_H}(t, p)$* defines *$\tau$ if first, $\rho \models \tau$ and second, for all $q \in Q$ and $i \in H$ we have $\psi_i(q) \geq 1 \iff \exists i_j \in \mathrm{leaf}_i(t) : \psi_i(q) = c_\rho(i_j)$.*
3. *The set of all tasks τ such that there is a tree $t \in T(\Sigma \cup H \cup \{\bot\})$ with $t \models \tau$ is denoted by $\mathcal{T} = \mathcal{T}_B$.*
4. *By $\pi(t)$ we denote the set of tasks τ such that $t \models \tau$.*
5. *The set of all profiles π such that there is a tree $t \in T(\Sigma \cup H \cup \{\bot\})$ with $\pi = \pi(t)$ is denoted by $\mathcal{P} = \mathcal{P}_B$.*
6. *By $\equiv_B$ we denote the equivalence relation which is defined by $t \equiv_B t' \iff \pi(t) = \pi(t')$.*

Note that $\pi(t) = \emptyset$ if and only if $\mathrm{Run}_{B_H}(t, p) = \emptyset$ for all $p \in Q$. Since $\bot \notin T(\Sigma \cup H)$, we have $\pi(\bot) = \emptyset$. The index of $\equiv_B$ is finite since it is bounded by $|\mathcal{P}|$. More precisely,

$$|\{\{t \in T(\Sigma \cup H) \mid t \equiv_B t'\} \mid t' \in T(\Sigma \cup H)\}| \leq |\mathcal{P}_B| \leq 2^{|Q| \cdot (|C|+1)^{|H \times Q|}}. \tag{12}$$

The value 0 in the range of ψ_i plays the following role: Let $q \in Q$ and $\rho \models (p, \psi_1, \ldots, \psi_{|H|})$ with $\rho \in \mathrm{Run}(p, t)$. Then $\psi_i(q) = 0$ implies $\{i_j \in \mathrm{leaf}_i(t) \mid \rho(i_j) = q\} = \emptyset$. The condition is vacuously true if $\mathrm{leaf}_i(t) = \emptyset$. The following lemma says that $\{t \in T(\Sigma \cup H) \mid t \models \tau\}$ is effectively regular for every task $\tau \in \mathcal{T}_B$. The proof of Prop. 14 is based on a product construction where the first component simulates the NFA B_H and the second component remembers the minimal color appearing on paths in runs ρ of B_H at states p.

▶ **Proposition 14.** *Let $\tau = (p, \psi_1, \ldots, \psi_{|H|}) \in \mathcal{T}_B$ be any task. Then the set of trees $\{t \in T(\Sigma \cup H \cup \bot) \mid t \models \tau\}$ is effectively regular. More precisely, $\{t \in T(\Sigma \cup H) \mid t \models \tau\} = L(B_\tau, (p, \chi(p)))$ where $B_\tau = (Q \times C, \Sigma \cup H, \delta_\tau, \chi_\tau)$ is the following parity-NTA. For a state $q \in Q$ in B, a function symbol $f \in \Sigma$ with $r = \mathrm{rk}(f)$, and a hole $i \in H$ we define δ_τ by:*

$$\big((q, c), f, (p_1, \min\{c, \chi(p_1)\}), \ldots, (p_r, \min\{c, \chi(p_r)\})\big) \in \delta_\tau \iff (q, f, p_1, \ldots, p_r) \in \delta_B \tag{13}$$

$$\big((q, c), i\big) \in \delta_\tau \iff c \preceq_{best} \psi_i(q) \tag{14}$$

The color of a pair (q, c) is defined by $\chi_\tau(q, c) = \chi(q)$.

Proof. Let $\mathrm{pr}_1 : Q \times C \to Q, (p, c) \mapsto p$ be the projection onto the first component. The definition of δ_τ says that $\mathrm{pr}_1\rho' : \mathrm{Pos}(t) \to Q$ is a run in $\mathrm{Run}_{B_H}(t, p)$ for every $t \in T(\Sigma \cup H)$ and every run $\rho' \in \mathrm{Run}_{B_\tau}(t, (p, \chi(p)))$. Consider any nonempty directed path $\varepsilon, u_1, \ldots, u_k$ in t, then, by induction on k, we see that $\rho'(u_k) = (q_k, c_k)$ satisfies $c_k = \min\{c_1, \ldots, c_k\}$. The definition of $\delta_\tau((q, c), i)$ implies $\mathrm{pr}_1\rho' \models \tau$ where, as usual, $\mathrm{pr}_1\rho' = \mathrm{pr}_1 \circ \rho'$ denotes the composition. Thus, the projection onto the first component defines a mapping

$$\mathrm{pr}_1 : \mathrm{Run}_{B_\tau}(t, (p, \chi(p))) \to \{\rho \in \mathrm{Run}_{B_H}(t, p) \mid \rho \models \tau\}, \ \rho' \mapsto \mathrm{pr}_1(\rho') = \mathrm{pr}_1\rho'. \tag{15}$$

Let us also define a mapping $\rho \mapsto \rho_\tau$ in the other direction from $\{\rho \in \mathrm{Run}_{B_H}(t, p) \mid \rho \models \tau\}$ to $\mathrm{Run}_{B_\tau}(t, (p, \chi(p)))$. Given $\rho \in \mathrm{Run}_{B_H}(t, p)$ such that $\rho \models \tau$, we define the run ρ_τ top-down beginning at the root with $\rho_\tau(\varepsilon) = (p, \chi(p))$. Now, assume that $\rho_\tau(u) = (q, c)$ is defined for $u \in \mathrm{Pos}(u)$ such that $\rho(u) = q$. Suppose $t(u) = f$ and $\mathrm{rk}(f) = r$ Then, thanks to $\rho \in \mathrm{Run}_{B_H}(t,p)$, there is some $(q,f,p_1,\dots,p_r) \in \delta_{B_H}$ such that $\rho(u.i) = p_i$ for all $1 \le i \le r$. However, since $\rho_\tau(u) = (q,c)$ is fixed there is exactly one corresponding transition $\big((q,c), f, (p_1, \min\{c,\chi(p_1)\}),\dots,(p_r,\min\{c,\chi(p_r)\})\big) \in \delta_\tau$ and we obtain $\rho_\tau(u.i) = \min\{c,\chi(p_i)\}$ for all $1 \le i \le r$. This is clear for $f \in \Sigma$ by (13) and $f = i \in H$ by (14) because $\rho \models \tau$. Moreover, since every infinite path in the tree ρ satisfies the parity condition, the same is true for the run ρ_τ because the color of a state $\chi(q,c)$ is defined by the first component: $\chi(q,c) = \chi(q)$. Thus, $\rho_\tau \in \mathrm{Run}_{B_\tau}(t,(p,\chi(p)))$ as desired. A straightforward inspection shows $(\mathrm{pr}_1\rho')_\tau = \rho'$ for all $\rho' \in \mathrm{Run}_{B_\tau}(t,(p,\chi(p)))$ and $\mathrm{pr}_1(\rho_\tau) = \rho$ for all $\rho \in \mathrm{Run}_{B_H}(t,p)$ where $\rho \models \tau$. This shows that the mapping pr_1 defined in (15) is bijective. We conclude $\{t \in T(\Sigma \cup H) \mid t \models \tau\} = L(B_\tau,(p,\chi(p)))$. This shows the proposition. ◀

▶ **Corollary 15.** *Let π be a profile, then $\{t \in T(\Sigma \cup H \cup \{\bot\}) \mid \pi(t) = \pi\}$ is effectively regular.*

Proof. Note that $\{t \in T(\Sigma \cup H \cup \{\bot\}) \mid \pi(t) = \pi\}$ coincides with

$$\bigcap_{\tau\in\pi} \{t \in T(\Sigma \cup H \cup \{\bot\}) \mid t \models \tau\} \cap \bigcap_{\tau\notin\pi} \{t \in T(\Sigma \cup H \cup \{\bot\}) \mid t \not\models \tau\} \tag{16}$$

which is a finite intersection of languages, or complements of languages, from the set of languages $\mathcal{L} = \{\{t \in T(\Sigma \cup H \cup \{\bot\}) \mid t \models \tau\} \mid \tau \in \tau_B\}$. By Prop. 14, $\mathcal{L}$ consists of regular tree languages. Since the class of regular tree languages in $T(\Sigma \cup H \cup \{\bot\})$ is an effective Boolean algebra [24], we conclude that $\{t \in T(\Sigma \cup H \cup \{\bot\}) \mid \pi(t) = \pi\}$ is effectively regular too. ◀

▶ **Proposition 16.** *Let B be a parity NTA with state set Q such that $L(B,p) \subseteq T(\Sigma)$ for all $p \in Q$ and $s \in T(\Sigma \cup \mathcal{X})$ be tree. If $\gamma,\gamma' : \mathrm{Pos}(s) \to T_\bot(\Sigma \cup H)$ are two choice functions for s such that $\gamma'(u) \equiv_B \gamma(u)$ for all $u \in \mathrm{Pos}(s)$, then $\gamma_\infty(s) \in L(B,p_0) \iff \gamma'_\infty(s) \in L(B,p_0)$ for all states p_0 of B.*

Proof. Let $s = x(s_1,\dots,s_r)$ and $\gamma_\infty(s) \in L(B,p_0)$. By symmetry, it is enough to show $\gamma'_\infty(s) \in L(B,p_0)$. Recall that $\gamma_\infty(s)$ is the limit of trees $\gamma_n(s)$. Fig. 7 depicts the basic relationship between paths in $\gamma_n(s)$ and paths in s, and that the positions $u \in \mathrm{Pos}(s)$ appearing on such a path in s can identified with positions in $\gamma_n(s)$. In the picture, it might happen that t_n is without any leaf labeled by a hole i. In this case, the position $u_n.i_{n+1}$ might still exist. However, if the leaf $i_j \in \mathrm{leaf}(\gamma_n(s))$ exists, then s has the position $u_n.i_{n+1}$. Whenever $p \in Q$ and $t = \gamma(u)$ for some $u \in \mathrm{Pos}(s)$, then each run $\rho \in \mathrm{Run}(\gamma(u),p)$ defines a minimal[6] task $\tau_\rho = (p,\psi_1,\dots,\psi_{|H|})$ which it satisfies, by

$$\psi_i(q) = \sup\nolimits_{\preceq_{\mathrm{best}}} \{c_\rho(i_j) \in \{0,\dots,|C|\} \mid \exists i_j : q = \rho(i_j)\}\,. \tag{17}$$

In particular, minimality of τ_ρ implies that there is some leaf i_j in t_u where $q = \rho(i_j)$ if and only if $\psi_i(q) \ge 1$. Moreover, the existence of ρ implies that $\gamma(u) \models \tau_\rho$. Since $\gamma(u) \equiv_B \gamma'(u)$, there exists for each $u \in \mathrm{Pos}(s)$, $p \in Q$, and $\rho \in \mathrm{Run}(\gamma(u),p)$ some $\rho' \in \mathrm{Run}(\gamma'(u),p)$ such that $\rho' \models \tau_\rho$. It follows that for all $(u,p) \in \mathrm{Pos}(s) \times Q$ there is a mapping

$$\theta : \mathrm{Run}(\gamma(u),p) \to \mathrm{Run}(\gamma'(u),p), \quad \rho \mapsto \rho' \tag{18}$$

such that $\rho' = \theta(\rho)$ satisfies $\rho' \models \tau_\rho$.

The definition of θ is crucial to prove $\gamma'_\infty(s) \in L(B,p_0)$. Note that the shape of $t = \gamma(u)$ and $t' = \gamma'(u)$ might be very different. For example, t can be finite whereas t' is infinite, or

[6] Minimal w.r.t. the natural partial order on tasks defined by $\preceq_{\mathrm{best}}$

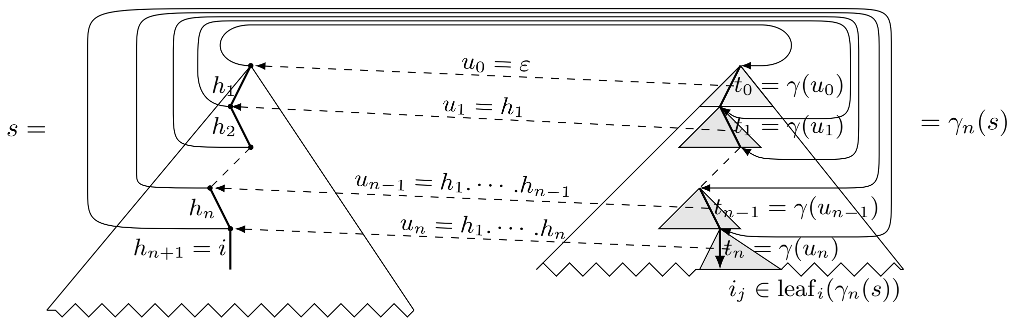

Figure 7 A directed path in $\gamma_n(s)$ starting at the root maps to a unique sequence of positions $\varepsilon, h_1, h_1.h_2 \ldots$ in s (indicated by dashed arrows). These positions of the sequence can be identified with positions in $\gamma_n(s)$: the top positions of the small triangles (indicated by plain arrows from positions in s to positions in $\gamma_n(s)$). The depicted directed path ends in a leaf i_j, but it could also stay inside t_n or end at a leaf of t_n which is a constant.

vice versa. The trees share however $\mathrm{leaf}_i(t) \neq \emptyset \iff \mathrm{leaf}_i(t') \neq \emptyset$ because $\gamma(u) \equiv_B \gamma'(u)$. Still, the cardinalities of $\mathrm{leaf}_i(t)$ and $\mathrm{leaf}_i(t')$ might differ drastically.

The proof uses an additional concept of IO-prefixes: Let $t, t_\varepsilon, t_i \in T_\perp(\Sigma \cup H)$ be trees for each $i \in H$ where $\mathrm{leaf}_i(t_\varepsilon) \neq \emptyset$. We say that t_ε *is an IO-prefix of* t if we can write $t = t_\varepsilon[i_j \leftarrow t_i]$.

In order to see $\gamma'_\infty(s) \in L(B, p_0)$ we fix a run $\rho \in \mathrm{Run}(\gamma_\infty(s), p_0))$ which certifies $\gamma_\infty(s) \in L(B, p_0)$. Using this fixed ρ we construct a set of (finite or infinite) sequences $\alpha = (u_0, p_0, \rho_0) \cdots (u_k, p_k, \rho_k)$ and $\beta = (u_0, p_0, \rho'_0) \cdots (u_k, p_k, \rho'_k)$ such that for all i we have $u_i \in \mathrm{Pos}(s)$, $p_i \in Q$, and $\rho_i \in \mathrm{Run}(\gamma(u_i), p_i)$ (resp. $\rho'_i \in \mathrm{Run}(\gamma'(u_i), p_i)$). We require that u_{j+1} is child of u_j for all $0 \leq j < k$. The set of sequences α (resp. β) form themselves a rooted tree where the root is the empty sequence. The parent of a nonempty sequence $w_0 \cdots w_{k-1} w_k$ of length k is the prefix $w_0 \cdots w_{k-1}$. Each node in the tree defined by α-sequences has at most $|H \times C|$ children whereas a node in the tree defined by β-sequences may have infinitely many children. The number of children is actually equal to the size of the index sets I_α resp. I_β as defined in (19) and (20) below.

Every such sequence $\alpha = (u_0, p_0, \rho_0) \cdots (u_k, p_k, \rho_k)$ will define a unique position $\nu(\alpha)$ in $\gamma_\infty(s)$ such that the subtree of $\gamma_\infty(s)$ at position $\nu(\alpha)$ has the tree $\gamma(u_k)$ as an IO-prefix as defined above. The same will be true for a sequence β with respect to $\gamma'_\infty(s)$. The position of the empty sequences is ε which is the root of $\gamma_\infty(s)$ and $\gamma'_\infty(s)$. We define a sequence $\alpha_1 = (u_0, p_0, \rho_0) = (\varepsilon, p_0, \rho_0)$ such that $\rho_0 \in \mathrm{Run}(\gamma(\varepsilon), p_0)$ is induced by $\rho \in \mathrm{Run}(\gamma_\infty(s), p_0)$. This yields $\beta_1 = (\varepsilon, p_0, \rho'_0)$ with $\rho'_0 = \theta(\rho_0)$ is defined by (18).

Now, let $k \geq 1$ and assume that a sequence $\alpha = (u_0, p_0, \rho_0) \cdots (u_k, p_k, \rho_k)$ is already defined. By induction, we also know a position $\nu(\alpha) \in \mathrm{Pos}(\gamma_\infty(s))$. Let

$$I_\alpha = \{(i, q) \in H \times Q \mid \exists i_j \in \mathrm{leaf}_i(\gamma(u_k)) : \rho_k(i_j) = q\}\,. \tag{19}$$

For each $(i, q) \in I_\alpha$ we define a triple $(u_k.i, q, \rho(i, q))$. Here, $u_k.i$ is the ith child of $u_k \in \mathrm{Pos}(s)$. Before we define the run $\rho(i, q)$, let us define the position $\nu(\alpha \cdot (u_k, q, \rho(i, q))) \in \mathrm{Pos}(\gamma_\infty(s))$. The subtree at position $\nu(\alpha) \in \mathrm{Pos}(\gamma_\infty(s))$ has $\gamma(u_k)$ as an IO-prefix; and every leaf $i_\ell \in \mathrm{leaf}_i(\gamma(u_k))$ defines a unique position in $\gamma_\infty(s)$ in the subtree at position $\nu(\alpha)$. Therefore, it is enough to choose a leaf in $\gamma(u_k)$. For that we choose any leaf $i_j \in \mathrm{leaf}_i(\gamma(u_k))$ such that $c_{\rho_k}(i_j) = \psi_i(q)$. For example, we can choose i_j to the first one in the length-lexicographical ordering of leaves. Hence, the minimal color on the path from the root in $\gamma(u_k)$ to the chosen leaf i_j induced by the run ρ_k is the greatest among all these colors among all $i_\ell \in \mathrm{leaf}_i(\gamma(u_k))$ with respect to best ordering. Since the root of $\gamma(u_k)$ is $\nu(\alpha)$ we can define $\nu(\alpha_k \cdot (u_k.i, q, \rho(i,q)))$ by the corresponding position of the chosen position i_j. Thus, for $\alpha' = \alpha \cdot (u_k.i, q, \rho(i,q))$ the position $\nu(\alpha') \in \mathrm{Pos}(\gamma_\infty(s))$ is defined. The subtree of $\gamma_\infty(s)$ at position $\nu(\alpha)$ has the tree $\gamma(u_k.i)$ as an IO-prefix. Hence, ρ induces a run $\rho(i,q) \in \mathrm{Run}(\gamma(u_k.i), q)$; and $(u_k.i, q, \rho(i,q))$ is defined by $(i,q) \in I_\alpha$. Note that α has a child α' if and only if $I_\alpha \neq \emptyset$. If $I_\alpha = \emptyset$ the sequence α is a leaf in the corresponding tree. The number of children is bounded by $|H \times Q|$.

The definition of the tree of β-sequences is defined analogously. We let $k \geq 1$ and assume that a sequence $\beta = (u_0, p_0, \rho'_0) \cdots (u_k, p_k, \rho'_k)$ is already defined such that $\rho'_i = \theta(\rho_i)$ for all $1 \leq i \leq k$. By induction, we also know a position $\nu(\beta) \in \mathrm{Pos}(\gamma'_\infty(s))$. The difference to α will be that β may have infinitely many children. More precisely, we define

$$I_\beta = \{(i, q, i_j) \in H \times Q \times \mathrm{leaf}_i(\gamma'(u_k)) \mid i_j \in \mathrm{leaf}_i(\gamma'(u_k)) \wedge \rho'_k(i_j) = q\}\,. \tag{20}$$

The set I_β is in a canonical bijection with the union $\bigcup\{\mathrm{leaf}_i(\gamma'(u_k)) \mid i \in H\}$. For each $(i, q, i_j) \in I_\beta$ we define the run $\rho'(i,q) = \theta(\rho(i,q))$. (The mapping θ was defined in (18) and the run $\rho(i,q)$ was defined above for the α-sequence by choosing a particular leaf.) Thereby we obtain a triple $(u_k.i, q, \rho'(i,q))$. For $\beta' = \beta \cdot (u_k.i, q, \rho'(i,q))$ the position $\nu(\beta') \in \mathrm{Pos}(\gamma'_\infty(s))$ is given in analogy to the position $\nu(\alpha')$. That is, we consider the position $\nu(\beta) \in \mathrm{Pos}(\gamma'_\infty(s))$. The subtree $\nu(\beta)$ has $\gamma'(u_k)$ as an IO-prefix; and we let $\nu(\beta') = \nu(\beta(i, q, i_j))$ the position which corresponds to the leaf i_j, Since I_β is in a canonical bijection with the set of all leaves in $\gamma'(u_k)$ labeled by some hole $i \in H$, we see that we actually have defined runs $\rho'_n : \mathrm{Pos}(\gamma'_n(s)) \to Q$ for all $n \in \mathbb{N}$, hence a run $\rho' : \mathrm{Pos}(\gamma'_\infty(s)) \to Q$.

It remains to show that the run ρ' is accepting. Since it is a run, it is enough to consider infinite paths w'. These paths result from a finite or infinite sequence $\beta = (u_0, p_0, \rho'_0)(u_1, p_1, \rho'_1) \cdots$. If the infinite path w' is defined by some finite prefix of β, then it is accepting[7] because every finite prefix of β defines an accepting path of $\gamma'_\infty(s)$ at p_0. So, we may assume that β is infinite, too. Hence, w' can be cut into infinitely many finite paths $w' = w'_0 w'_1 \cdots$ where each w'_i is a path inside $\gamma'(u_i)$. Each finite path belongs to a run $\rho'_{v'}$ of $\gamma'(u)$ at some position $v' \in \mathrm{Pos}(\gamma'_\infty(s))$. It corresponds to the some run ρ_v of $\gamma(u)$ at some position $v \in \mathrm{Pos}(\gamma_\infty(s))$. In this way we obtain an infinite path in $w = w_0 w_1 \cdots$ in $\gamma_\infty(s)$ where each w_i is a path in $\gamma(u_i)$. Recall that β was defined together with a corresponding sequence $\alpha = (u_0, p_0, \rho_0)(u_1, p_1, \rho_1) \cdots$ such that $\rho'_i = \theta(\rho_i)$. Every path in tree $\rho' : \mathrm{Pos}(\gamma'_\infty(s))$ can be mapped to some accepting path in $\rho : \mathrm{Pos}(\gamma_\infty(s))$. This situation is depicted in Fig. 8. The task τ_v defined by ρ_v is satisfied by $\rho'_{v'}$. Hence, the minimal color seen on the path from the root in $\gamma'(u)$ to a leaf i_j is never greater in the best-ordering than the minimal color seen on the path from the root in $\gamma(u)$ to a leaf i_j. As a consequence, the minimal color seen infinitely often on the run defined by the path w' is not greater in the best-ordering than the one defined by w. Since the minimal color seen infinitely often defined by w is even, the same is true for the minimal color seen infinitely often defined by w'. ◀

4.1 The saturation of σ

We continue with a parity-NTAs $B = (Q, \Sigma, \delta, \chi)$ and $B_H = (Q, \Sigma \cup H, \delta_H, \chi)$ as above where $\delta_H = \delta \cup (Q \times H)$.

[7] Accepting paths were defined in Def. 10.

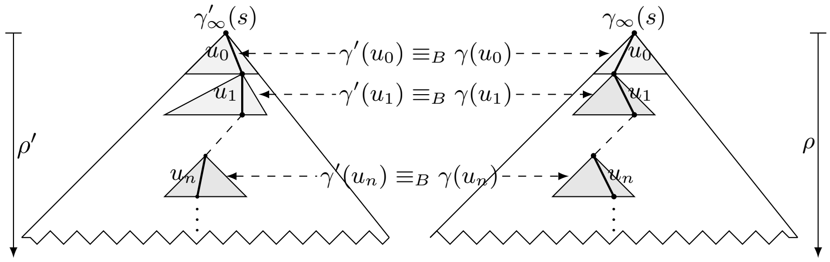

Figure 8 The $u_i \in \mathrm{Pos}(s)$ are the top positions of the small triangles according to Fig. 7. The run ρ defines a run ρ'. It satisfies the parity condition because every path in tree ρ has this property.

▶ **Definition 17.** *Let $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H}$ be a substitution. The* saturation of σ *(w.r.t. B) is the substitution $\widehat{\sigma} : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H}$ defined by*

$$\widehat{\sigma}(x) = \{t' \in T(\Sigma \cup \mathcal{X}) \setminus H \mid \exists t \in \sigma(x) : t' \equiv_B t\}.$$

▶ **Proposition 18.** *Let $q_0 \in Q$ be a state of the NTA B and $L \subseteq T(\Sigma \cup \mathcal{X})$ be any subset. Then $\widehat{\sigma}(x)$ is regular for all $x \in \mathcal{X}$. Moreover, if $\sigma_{\mathrm{io}}(L) \subseteq L(B, q_0)$, then $\widehat{\sigma}_{\mathrm{io}}(L) \subseteq L(B, q_0)$.*

Proof. We have $\widehat{\sigma}(x) = \bigcup_{\pi \in \{\pi(t) \mid t \in \sigma(x)\}} \{t' \in T(\Sigma \cup H) \mid \pi(t') = \pi\}$. This is a finite union. Thus, $\widehat{\sigma}(x)$ is regular by Cor. 15. To prove the second part of the proposition it suffices to show that, if $s \in T(\Sigma \cup \mathcal{X})$ and $\sigma_{\mathrm{io}}(s) \in L(B, q_0)$ then $\widehat{\sigma}_{\mathrm{io}}(s) \in L(B, q_0)$, that is, $\gamma'_\infty(s) \in L(B, q_0)$ for all $\gamma' \in \Gamma(s, \widehat{\sigma})$. Let $\gamma' \in \Gamma(s, \widehat{\sigma})$. Then, by definition of $\widehat{\sigma}$, for every $u \in \mathrm{Pos}(s)$, where $\sigma(s(u)) \neq \emptyset$, we can choose a tree $\gamma(u) = t \in \sigma(s(u))$ such that $\gamma'(u) \equiv_B \gamma(u)$. Thus, $\gamma \in \Gamma(s, \sigma)$ and $\gamma_\infty(s) \in L(B, q_0)$ because $\sigma_{\mathrm{io}}(s) \subseteq L(B, q_0)$. By Prop. 14, we have $\gamma'_\infty(s) \in L(B, q_0)$ too. Hence, $\widehat{\sigma}_{\mathrm{io}}(s) \subseteq L(B, q_0)$ as desired. ◀

The following corollary is another crucial step towards Thm. 30.

▶ **Corollary 19.** *Let $\sigma_1, \sigma_2 : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H}$ be regular substitutions such that $\sigma_1 \leq \sigma_2$ (for the natural partial order by set inclusion for components), $R \subseteq T(\Sigma)$ be a regular tree language, and $L \subseteq T(\Sigma \cup \mathcal{X})$ be any subset. Then there is an effectively computable finite set S_2 of regular substitutions such that for every $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H}$ satisfying both, $\sigma_1 \leq \sigma \leq \sigma_2$ and $\sigma_{\mathrm{io}}(L) \subseteq R$ (resp. $\sigma_{\mathrm{io}}(L) = R$), there is some maximal substitution $\sigma' \in S_2$ such that $\sigma'_{\mathrm{io}}(L) \subseteq R$ (resp. $\sigma'_{\mathrm{io}}(L) = R$) and $\sigma \leq \sigma' \leq \sigma_2$.*

Proof. We may assume that $R = L(B, q_0)$. Throughout the proof, given any substitution σ, the notation $\widehat{\sigma}$ refers to its saturation according to Def. 17. Moreover, by Prop. 18, every $\widehat{\sigma}$ is a regular substitution and if $\sigma_{\mathrm{io}}(L) \subseteq R$, then $\sigma_{\mathrm{io}}(L) \subseteq \widehat{\sigma}_{\mathrm{io}}(L) \subseteq R$. Based on these facts, we define in a fist step the following set of substitutions.

$$S_0 = \left\{\widehat{\sigma} \;\middle|\; \sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H} \text{ is a substitution}\right\}. \tag{21}$$

The set S_0 is finite and effectively computable. Its cardinality is bounded by $2^{|\mathcal{P}_B \times \mathcal{X}|}$. Since σ_1 and $\widehat{\sigma}$ are regular for every $\widehat{\sigma} \in S_0$, we can effectively compute the following subset S_1 of S_0, too.

$$S_1 = \left\{\widehat{\sigma} \in S_0 \;\middle|\; \sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H} \wedge \sigma_1 \leq \sigma\right\} = \{\widehat{\sigma} \in S_0 \mid \sigma_1 \leq \widehat{\sigma}\}. \tag{22}$$

The second equation in (22) uses $\widehat{\widehat{\sigma}} = \widehat{\sigma}$. Now, for every substitution $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H)\setminus H}$ satisfying $\sigma_1 \leq \sigma \leq \sigma_2$ we have $\widehat{\sigma} \in S_1$. But $\widehat{\sigma} \leq \sigma_2$ might fail. However, since σ_2 is regular, we can calculate for each $\widehat{\sigma} \in S_1$ the substitution σ' such that $\sigma'(x) = \widehat{\sigma}(x) \cap \sigma_2(x)$ for all $x \in \mathcal{X}$. Hence, the following finite set is effectively computable:

$$S_2 = \left\{ \sigma' : \mathcal{X} \to 2^{T(\Sigma \cup H) \setminus H} \,\middle|\, \exists \widehat{\sigma} \in S_1 \, \forall x \in \mathcal{X} : \sigma'(x) = \widehat{\sigma}(x) \cap \sigma_2(x) \right\}. \tag{23}$$

An easy reflection shows that for all σ such that $\sigma_1 \leq \sigma \leq \sigma_2$ and $\sigma_{\mathrm{io}}(L) \subseteq R$ there is some substitution $\sigma' \in S_2$ such that $\sigma \leq \sigma' \leq \sigma_2$ and $\sigma'_{\mathrm{io}}(L) \subseteq R$. Since S_2 is finite, there is a maximal element σ' in S_2 such that $\sigma \leq \sigma' \leq \sigma_2$ and $\sigma'_{\mathrm{io}}(L) \subseteq R$. The assertion of Cor. 19 follows. ◀

▶ Remark 20. Note that Cor. 19 does not tell us how to decide whether there exists any substitution σ such that $\sigma_{\mathrm{io}}(L) \subseteq R$ or $\sigma_{\mathrm{io}}(L) = R$. It just tells us that if there exists such a substitution σ, then there there exist such a σ in an effectively computable finite set of regular substitutions.

4.2 The specialization of σ

The roadmap in this section is as follows. We enlarge the set Σ by various new function symbols in order to obtain a larger ranked alphabet $\Sigma_{\mathcal{P}}$. In particular, for each $\pi \in \mathcal{P}_B$ there is a function symbol f_π and a finite tree t_π with $t_\pi(\varepsilon) = f_\pi$. Simultaneously we define an extension $B_{\mathcal{P}}$ of B_H such that for each $\pi = \pi(t)$ (w.r.t. the NTA B_H) we have $\pi = \pi(t) = \pi(t_\pi)$ where $\pi(t)$ and $\pi(t_\pi)$ are defined with respect to $B_{\mathcal{P}}$.

The formal definitions of $\Sigma_{\mathcal{P}}$ and $B_{\mathcal{P}}$ are in Def. 21 and Def. 22. Since every symbol f_π has a rank which only depends on π, but which will be defined through a tree t where $\pi = \pi(t)$, it is important to note that $\emptyset \neq \pi(t) = \pi(t')$ implies $\mathrm{leaf}_i(t) \neq \emptyset \iff \mathrm{leaf}_i(t') \neq \emptyset$ for all holes $i \in H$. This follows because if $\emptyset \neq \pi(t)$, then $\mathrm{leaf}_i(t) \neq \emptyset$ if and only if for all tasks $(p, \psi_1, \ldots, \psi_{|H|}) \in \pi$ there is some $q \in Q$ such that $\psi_i(q) > 0$. Indeed, let $\rho \in \mathrm{Run}(t,p)$ such that $\rho \models \tau$. Then for all $i_j \in \mathrm{leaf}_i(t)$ the run ρ yields some state $\rho(i_j) \in Q$. Since $\chi\rho(i_j) > 0$ we must have $\psi_i(\rho(i_j)) > 0$.

▶ **Definition 21.** 1. *For each $t \in T(\Sigma \cup H)$ we denote by $H(t)$ the subset of holes $i \in H$ such that* $\mathrm{leaf}_i(t) \neq \emptyset$.

2. *The alphabet $\Sigma_{\mathcal{P}}$ contains Σ and, in addition, a new function symbol $\$_i$ of rank 1 for each $i \in H$ and for each $t \in T(\Sigma \cup H)$ and each profile $\emptyset \neq \pi = \pi(t) \in \mathcal{P}$ it contains a new function symbol f_π of rank $|Q \times H(t)|$. In particular, if t is without holes, then f_π is a constant. We also include a new constant $f_\emptyset$ in case $\emptyset \in \mathcal{P}_B$ (but we will make sure that no run at any state of $B_{\mathcal{P}}$ accepts $f_\emptyset$).*
3. *For each $\emptyset \neq \pi = \pi(t) \in \mathcal{P}$ we define the finite tree t_π as follows. There are $k = |Q \times H(t)|$ holes, all of them belong to $H(t)$, and each hole $i \in H(t)$ appears exactly $|Q|$ times (in some fixed order). Moreover, the parent node of each leaf labeled by a hole i is labeled by $\$_i$. For $H(t) \neq \emptyset$ we have $k \geq 1$ and the corresponding tree is depicted in Fig. 9. If t is without holes, then $k = 0$ and $t_\pi = f_\pi$ is a constant of rank zero.*

We are ready to define an extension $B_{\mathcal{P}}$ of the automaton B_H such that for all $t \in T(\Sigma \cup H)$ the finite tree t_π satisfies $t \equiv_{B_{\mathcal{P}}} t_\pi$ if and only if $\pi(t) = \pi$.

▶ **Definition 22.** *We let $B_{\mathcal{P}} = (Q', \Sigma_{\mathcal{P}}, \delta_{\mathcal{P}}, \chi)$ denote the following parity-NTA. It is an extension of B_H. The set of states is $Q' = Q \cup (C \times Q \times H)$. The coloring χ is extended to states in $C \times Q \times H$ by $\chi'((c,q,h)) = \chi(c)$.*

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, for each $\emptyset \neq \pi(t) \in \mathcal{P}_B$ the following set of tuples. The definition of these *tuples uses the set of holes $H(t)$, but this is well-defined because $\emptyset = \pi(t) = \pi(t')$ implies $H(t) = H(t')$.*

- $\big(p, f_\pi, (t_{s,j})_{(s,j)\in Q\times H(t)}\big)$ *for* $\tau = (p, \psi_1, \dots, \psi_{|H|}) \in \pi$ *such that*

$$\{t_{s,j} \,|\, (s,j) \in Q \times H(t)\} = \{(\psi_i(q), q, i) \,|\, q \in Q, i \in H(t), \psi_i(q) \neq 0\}\,,$$

- $\big((c, q, i), \$_i, q\big)$ *for all* $(c, q, i) \in C \times Q \times H(t)$.

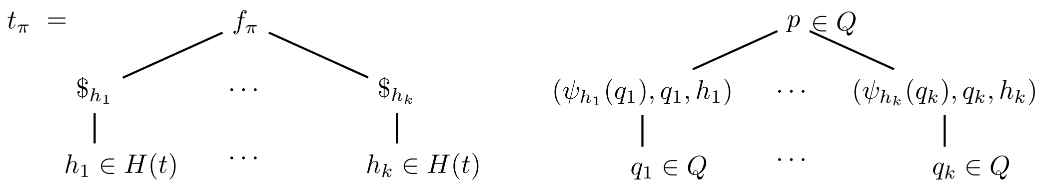

Figure 9 The tree t_π where $k = \mathrm{rk}(f_\pi)$ and an accepting run ρ on t_π for $\tau = (p, \psi_1, \dots, \psi_{|H|}) \in \pi$.

The cardinality of the set $\{(\psi_i(q), q, i) \,|\, q \in Q, i \in H(t), \psi_i(q) \neq 0\}$ varies. It is not necessarily the same for all $\tau \in \pi$. It can be any number between $|H(t)|$ and $\mathrm{rk}(f_\pi)$ which only depends on π and which was defined as $\mathrm{rk}(f_\pi) = |Q \times H(t)|$. Thus, using triples $(\psi_i(q), q, i)$ several times we can make sure that for every $\tau \in \pi$ there exists some transition $\big(p, f_\pi, (t_{s,j})_{(s,j)\in Q\times H(t)}\big)$ in $\delta_{\mathcal{P}}$.

▶ **Lemma 23.** *Let $t \in T(\Sigma \cup H)$, $\pi = \pi(t) \in \mathcal{P}_B$, and t_π as defined above. Then for all tasks $\tau \in \mathcal{T}_{B_\mathcal{P}}$ we have $t_\pi \models \tau \iff \tau \in \pi$. In particular, $t \equiv_{B_\mathcal{P}} t_\pi$ and $\pi(t_\pi) = \pi$.*

Proof. For each $\tau = (p, \psi_1, \dots, \psi_{|H|}) \in \pi$ the NTA $B_\mathcal{P}$ admits for t_π a successful run at p. This is clear for $H(t) = \emptyset$ because then t_π is a constant and $(p, t_\pi) \in \delta_\mathcal{P}$. For $H(t) \neq \emptyset$ the accepting run is depicted on the right of Fig. 9. It is accepting because $Q \times H \subseteq \delta_H \subseteq \delta_\mathcal{P}$. Hence, $\tau \in \pi$ implies $t_\pi \models \tau$. The converse follows from the definition of $\delta_\mathcal{P}$. In particular, $t_\pi \models \tau$ implies $\tau \in \mathcal{T}_B$. Since $\Sigma \subseteq \Sigma_\mathcal{P}$ and $Q \subseteq Q'$, we have $t \in T(\Sigma_\mathcal{P} \cup H)$ and $\mathcal{T}_B \subseteq \mathcal{T}_{B_\mathcal{P}}$ and therefore, $t \equiv_{B_\mathcal{P}} t_\pi$. ◀

Prop. 25 reduces the computation of $\sigma_{\mathrm{io}}^{-1}(R)$ to compute of $\check{\sigma}_{\mathrm{io}}^{-1}(R)$ where $\check{\sigma}(x)$ is defined next for all $x \in \mathcal{X}$. It is a subset of the finite set of finite trees $\{t_\pi \,|\, \pi \in \mathcal{P}_B\}$.

▶ **Definition 24.** *Let $\sigma : \mathcal{X} \to 2^{T(\Sigma\cup H)}$ be a substitution and $\pi \in \mathcal{P}$ be a profile. The* specialization *$\check{\sigma} : \mathcal{X} \to 2^{T(\Sigma_\mathcal{P}\cup H)}$ w.r.t. $B_\mathcal{P}$ is defined by the substitution*

$$\check{\sigma}(x) = \{t_\pi \in T(\Sigma_\mathcal{P} \cup H) \,|\, \exists t \in \sigma(x) : \pi = \pi(t)\}\,. \tag{24}$$

Here, t_π is the tree defined according to Def. 21. That is, if $\pi = \pi(t)$ for some t without holes, then $t_\pi = f_\pi$. Otherwise $k \geq 1$ and t_π is depicted in Fig. 9 on the left-side.

Note that $\sigma(x) = \emptyset \iff \check{\sigma}(x) = \emptyset$ and $t_\emptyset \in \check{\sigma}(x) \iff \exists t \in \sigma(x) : t \models \emptyset$.

▶ **Proposition 25.** *Let $R = L(B, q_0) \subseteq T(\Sigma)$ and let $\Sigma_\mathcal{P}$ and $B_\mathcal{P} = (Q', \Sigma_\mathcal{P} \cup H, \delta_\mathcal{P}, \chi')$ according to Def. 22. Let $\sigma : \mathcal{X} \to 2^{T(\Sigma\cup H)\setminus H}$ be any substitution and $\check{\sigma} : \mathcal{X} \to 2^{T(\Sigma_\mathcal{P}\cup H)}$ its specialization according to Def. 24. Then we have*

$$\{s \in T(\Sigma \cup \mathcal{X}) \,|\, \sigma_{\mathrm{io}}(s) \subseteq L(B, q_0)\} = \{s \in T(\Sigma \cup \mathcal{X}) \,|\, \check{\sigma}_{\mathrm{io}}(s) \subseteq L(B_\mathcal{P}, q_0)\}\,. \tag{25}$$

Proof. The equality $\{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{\mathrm{io}}(s) \subseteq L(B, q_0)\} = \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{\mathrm{io}}(s) \subseteq L(B_{\mathcal{P}}, q_0)\}$ follows directly from the definition of $B_{\mathcal{P}}$. We apply Prop. 18 twice, once to B and once to $B_{\mathcal{P}}$. We obtain $\{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{\mathrm{io}}(s) \subseteq L(B_{\mathcal{P}}, q_0)\} = \{s \in T(\Sigma \cup \mathcal{X}) \mid \widehat{\sigma}_{\mathrm{io}}(s) \subseteq L(B_{\mathcal{P}}, q_0)\}$ as well as $\left\{s \in T(\Sigma \cup \mathcal{X}) \,\middle|\, \widehat{\check{\sigma}}_{\mathrm{io}}(s) \subseteq L(B_{\mathcal{P}}, q_0)\right\} = \{s \in T(\Sigma \cup \mathcal{X}) \mid \check{\sigma}_{\mathrm{io}}(s) \subseteq L(B_{\mathcal{P}}, q_0)\}$.

Note that $\widehat{\sigma}_{\mathrm{io}}(s)$ refers to $\widehat{\sigma}(x) = \{t' \in T(\Sigma_{\mathcal{P}} \cup H) \mid \exists t \in \sigma(x) : t' \equiv_{B_{\mathcal{P}}} t\}$. That is to the saturation with respect to terms in $T(\Sigma_{\mathcal{P}} \cup H)$. By Lem. 23, we have $t_\pi \equiv_{B_{\mathcal{P}}} t$ for every $t \in \sigma(x)$ and profile $\pi = \pi(t)$. Hence, $\{s \in T(\Sigma \cup \mathcal{X}) \mid \widehat{\sigma}_{\mathrm{io}}(s) \subseteq L(B_{\mathcal{P}}, q_0)\} = \left\{s \in T(\Sigma \cup \mathcal{X}) \,\middle|\, \widehat{\check{\sigma}}_{\mathrm{io}}(s) \subseteq L(B_{\mathcal{P}}, q_0)\right\}$. Putting everything together we see

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

The result follows. ◀

5 Parity games

A *directed (multi-)graph* is a tuple $G = (V, E, s, t)$ where V is the set of vertices and the sets s and t are functions from E to V. Here, $s(e)$ denotes the source of e and $t(e)$ denotes the target of e. In our paper an *arena* is any tuple $A = (V_0, V_1, E, s, t)$ such that (V, E, s, t) is a directed graph where $V = V_0 \cup V_1$ and V_0, V_1 are disjoint.

Phrased differently, every vertex is in exactly one subset V_i of V and we allow multiple edges between vertices. The maps s and t are extended to all paths of the graph A in the natural way: if $w = p_0, e_1, p_1, \ldots, e_m, p_m$ is a path of A where $m \geq 0$, then we let $s(w) = p_0$ and $t(w) = p_m$. A *parity game* (or perhaps more accurately *the board of a game*) is defined by a pair (A, χ) where A is an arena and $\chi : V \to C$ is a mapping to a set of *colors* C.

Without restriction, we assume that $C = \{1, \ldots, |C|\}$ and that $|C|$ is odd. Let $p_0 \in V$. A *game at* p_0 is a finite or infinite sequence $p_0, e_1, p_1, e_2 \ldots$ such that $p_{i-1} = s(e_1)$, $p_i = t(e_i)$ and $e_i \in E$ for $i \geq 1$. Moreover, we require that a game is infinite unless it ends in a sink. A *sink* is a vertex without outgoing edges[8]. There are two players: P_0 (*Prover*) and P_1 (*Spoiler*). The rules of the game are as follows. It starts in the vertex p_0. Let $m \geq 0$ such that a path $p_0, e_1, p_1, \ldots, e_m, p_m$ is already defined with $p_m \in V_i$. If p_m is a sink, then player P_i loses. In the other case player P_i chooses an outgoing edge $e_{m+1} \in E$ such that $p_m = s(e)$ and $p_{m+1} = t(e)$. The game continues with $p_0, \ldots, p_m, e_{m+1}, p_{m+1}$.

If the game does not end in a sink, then the mutual choices define an infinite sequence. Prover P_0 wins an infinite game if the least color which appears infinitely often in $\chi(p_0), \chi(p_1), \ldots$ is even. Otherwise, Spoiler P_1 wins that game.

A *positional strategy* for player P_i is a subset $E_i \subseteq E$ such that for each $u \in V_i$ there is at most one edge $e \in E_i$ with $s(e) = u$ and $t(e) = v$. Hence, every $e \in E_i$ is an edge of the arena. However, the property that E_i is *winning* depends on the pair $(s(e), t(e))$, only. Therefore, we can assume $E_i \subseteq V_i \times V$. Each positional strategy defines a subarena $A_i = (V, E_i \cup \{e \in E \mid s(e) \in V_{1-i}\})$. In the arena A_i player P_{1-i} wins at a vertex p if and only if there exists some path starting at p which satisfies his winning condition defined above. Let $W_i \subseteq V$ be the set of vertices $p \in V$ of A_i where no path starting at p satisfies the winning condition of P_{1-i}. Then W_i contains all vertices 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

[8] Frequently in the literature an arena is not allowed to have sinks. This is an unnecessary restriction.

in the original arena A, because player P_i is able to win, no matter how P_{1-i} decides for $p_m \in V_{1-i}$ on the next outgoing edge $e \in E$ with $s(e) = p_m$.

▶ **Theorem 26** ([17]). *There exist positional strategies $E_i = V_i \times V \subseteq E$ for both players P_i such that their sets of winning positions $W_i \subseteq V$ (w.r.t. the subarenas A_i) satisfy $W_{1-i} = V \setminus W_i$. That is, V is the disjoint union of W_{1-i} and W_i.*

The theorem implies that for parity games there is no better strategy than a positional one. Thm. 26 is due to Gurevich and Harrington. Simplified proofs are e. g. in [15, 27].

5.1 Alternating tree automata

Let Δ be a finite ranked alphabet with the rank function $\mathrm{rk} : \Delta \to \mathbb{N}$. Nondeterministic tree automata are special instances of alternating tree automata. ATAs were introduced in [21] and further investigated in [22]. 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 function*, and $\chi : Q \to C$ is a coloring with $C = \{1, , \ldots, |C|\}$ and where, without restriction, $|C|$ is odd. For each $(p, f) \in Q \times \Delta$ there is exactly one transition which has the form $\big(p, f, \Phi\big)$ where Φ is a positive Boolean formula over the set $[\mathrm{rk}(f)] \times Q$ and, moreover, Φ is written in disjunctive normal form[9]. This means that $\big(p, f, \Phi\big) \in \delta$ is written as

$$\big(q, x, \bigvee_{j \in J} \bigwedge_{k \in K_j} (i_k, p_k)\big) \tag{26}$$

where J and K_j are finite sets and $(i_k, p_k) \in [\mathrm{rk}(x)] \times Q$. The first component $i_k \in [\mathrm{rk}(f)]$ is also called a *direction*. With each $j \in J$ there is an associated finite set of indices K_j. We use a syntax for Boolean formulas defined by a context-free grammar, but in the notation redundant brackets are typically removed as in (26). The syntax still allows repetitions of pairs (i_k, p_k). For example, we may have $K_j = \{k, \ell\}$ with a conjunction $\big((i_k, p_k) \wedge (i_\ell, p_\ell)\big)$ where $(i_k, p_k) = (i_\ell, p_\ell)$. This will have no influence on the semantics which we define next, but it will introduce "multiple edges" in the corresponding game-arena, which will make the set of choices for Spoiler larger [10].

By definition, let $p \in Q$, then $L(A, p)$ is the set of trees $t \in T(\Delta)$ such that Prover P_0 has a winning strategy the following parity-game in the arena $G(A, s)$ at vertex (ε, p).

The set V_0 of vertices belonging to Prover of the arena $G(A, s)$ is defined by $V_0 = \mathrm{Pos}(s) \times Q$. The color of (u, q) is $\chi(q)$. Next we define the set E_0 of outgoing edges at $(u, q) \in V_0$ by set of all (u, q, j) with $j \in J$ and where J appears under the disjunction in (26) of the unique transition $\big(q, s(u), \bigvee_{j \in J} \bigwedge_{k \in K_j} (i_k, p_k)\big) \in \delta$. We let $s(u, q, j) = (u, q)$ and we let $t(u, q, j) = (u, q, j)$. We define the set of positions belonging to Spoiler by $V_1 = t(E_0)$ and we let $\chi(u, q, j) = \chi(q)$. In that way the function $t : E_0 \to V_1$ becomes surjective. Every vertex in V_1 has an incoming edge. Note that for $(q, c) \in V_0$ and any $t \in V_1$ there is at most one edge $e \in E$ with $s(e) = (u, q)$. No multiple edges are necessary here. This changes when we define the outgoing edges for vertices $(u, q, j) \in V_1$. The outgoing edges are the quadruples (u, q, j, k) where $k \in K_j$ for index sets under the conjunction in the transition $\big(q, s(u), \bigvee_{j \in J} \bigwedge_{k \in K_j} (i_k, p_k)\big) \in \delta$. We define $s(u, q, j, k) = (u, q, j)$. The index k defines a

[9] Let S be any (finite) set. Then the set of *positive Boolean formulae* $\mathbb{B}_+(S)$ is defined inductively. 1. The symbols $\perp$ and $\top$ and all $s \in S$ belongs to $\mathbb{B}_+(S)$. (The elements of S are the *atomic propositions*.) 2. If $\varphi, \psi \in \mathbb{B}_+(S)$, then $(\varphi \vee \psi) \in \mathbb{B}_+(S)$ and $(\varphi \wedge \psi) \in \mathbb{B}_+(S)$.

[10] This larger set of choices is explicitly used in the definition of w_g in the proof of Lemma 28

pair (i_k, p_k) and then we let $t(u, q, j, k) = (u.i_k, p_k)$. Thus, there can be many edges with the source $(u, q, j) \in V_1$ and target $(u.i_k, p_k)$. Thanks to the condition $(i_k, p_k) \in [\mathrm{rk}(x)] \times Q$ we are sure that the position $u.i_k$ exists as soon as $K_j \neq \emptyset$.

Let us phrase the definition of the arena from the perspective of the players if the game starts at some vertex $(u, q) \in V_0$. Then Prover chooses, if possible, an index $j \in J$ according to set E_0. If J is empty, then Prover loses. (An empty disjunction is "false".) In the other case, it is the turn of Spoiler P_1, and the game continues at the vertex (u, q, j) belonging to P_1 with color $\chi(q)$. If K_j is empty, then Spoiler loses. (An empty conjunction is "true".) Otherwise, Spoiler chooses an index $k \in K_j$ and the game continues at the vertex $(u.i_k, p_k) \in V_0 = \mathrm{Pos}(s) \times Q$ (which exists). 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 occurring infinitely often is even.

A *parity-NTA* A is a special instance of an alternating automaton, where every transition as in Eq.(26) has the special form $\left(p, f, \bigvee_{j \in J} \bigwedge_{i \in [\mathrm{rk}(f)]} (i, p_i)\right)$. For convenience we use only the traditional syntax that δ is a set of tuples

$$\left(p, f, p_1, \ldots, p_{\mathrm{rk}(f)}\right). \tag{27}$$

The main results of alternating tree automata can be formulated as follows.

▶ **Theorem 27** ([21, 22]). *Let A be a parity-NTA and p be a state in the notation as above. Then the following assertions hold.*

1. *Viewing A as a special instance of a parity-ATA or using the traditional syntax (27) and semantics according to Def. 10 yields the same set $L(A, p)$.*
2. *Parity-ATAs characterize the class of regular tree languages: if $L(A, p)$ is defined by a parity-ATA A at a state p, then we can construct effectively a parity-NTA B and a state q such that $L(A, p) = L(B, q)$.*

5.2 Alternating tree automata for languages of type ${\sigma_{\mathrm{io}}}^{-1}(R)$

Thanks to Prop. 25, it is enough to consider the concept of ATAs in the setting where there is a finite set $T \subseteq T_{\mathrm{fin}}(\Sigma \cup H) \setminus H$ of finite trees such that $\bigcup \{\sigma(x) \mid x \in \mathcal{X}\} \subseteq T$. This is not essential but allows to use the standard definition in (26) for transitions where the index sets J and K_j are assumed to be finite.

▶ **Lemma 28.** *Let $R \subseteq T(\Sigma)$ be a regular tree language and $\gamma : \mathcal{X} \to T_{\mathit{fin}}(\Sigma \cup H) \setminus H$ be a homomorphism to the set of finite trees. Then the set of trees*

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

is effectively regular.

Proof. Since γ a homomorphism, there is for every $s \in T(\Sigma \cup \mathcal{X})$ a unique choice function γ^s defined by $\gamma^s(u) = \gamma(s(u))$ for all $u \in \mathrm{Pos}(s)$. Moreover, $\gamma^s(u) \neq \bot$ and hence, $\gamma_{\mathrm{io}}(s) = \{\gamma^s_\infty(s)\}$ is a singleton and $\gamma^s_\infty(s) \in T(\Sigma)$ for all $s \in T(\Sigma \cup \mathcal{X})$. In the following we use the abbreviation $\gamma_\infty(s) = \gamma^s_\infty(s)$ for simpler reading.

We may assume that $R = L(B, q_0)$ for some parity-NTA $B = (Q, \Sigma, \delta, \chi)$ and $q_0 \in Q$. Thus, it is enough to show that the set $\{s \in T(\Sigma \cup \mathcal{X}) \mid \gamma_\infty \in L(B, p)\}$ is effectively regular for all $p \in Q$. As usual, we assume that $\chi : Q \to \{1, \ldots, |C|\}$ where $|C|$ is odd. Let us define a parity-ATA $A = (Q \times C, \Sigma \cup \mathcal{X}, \delta_A, \chi_A)$ (where $\chi_A(q,c) = \mathrm{pr}_2(q,c) = c$) such that for all $p \in Q$ we have

$$\{s \in T(\Sigma \cup \mathcal{X}) \mid \gamma_\infty \in L(B,p)\} = L(A, (p, \chi(p))). \tag{28}$$

The set δ_A is defined by the following transitions

$$\Big((p,c), x, \bigvee_{\rho \in \mathrm{Run}_{B_H}(\gamma(x),p)} \bigwedge_{i_j \in K_\rho} \big(i, (\rho(i_j), c_\rho(i_j))\big)\Big) \tag{29}$$

The notation in (29) is as follows: $x \in \Sigma \cup \mathcal{X}$ where $\gamma(x) = x(1, \dots, \mathrm{rk}(x))$ for $x \in \Sigma \setminus \mathcal{X}$ and

$$K_\rho = \{i_j \in \mathrm{Pos}(\gamma(x)) \mid \exists\, i \in H : \, i_j \in \mathrm{leaf}_i(\gamma(x))\} \tag{30}$$

Recall that for a tree $t \in T(\Sigma \cup H)$ and a run $\rho \in \mathrm{Run}_{B_H}(t,p)$, the color $c_\rho(i_j)$ denotes the minimal color in the tree $\rho(t)$ which appears on the unique path from the root ε to the leaf i_j. If $\mathrm{Run}_{B_H}(\gamma(x),p) = \emptyset$, then the disjunction is over the empty set, hence the transition in (29) becomes (p, x, false). If $\mathrm{Run}_{B_H}(\gamma(x),p) \neq \emptyset$ but $\gamma(x)$ is a tree without holes, then the transition in (29) becomes (p, x, true).

Using Thm. 27 it is enough to show that for all $p \in Q$ we have $\gamma_\infty(s) \in L(B,p)$ if and only if Prover P_0 has a winning strategy in the associated parity game for the arena $G(A,s)$ at vertex $(\varepsilon, (p, \chi(p)))$ where A is the parity-ATA above. Note that the index set K_ρ allows Spoiler to pick any hole which appears in $\gamma(x)$.

One direction is easy. If $\gamma_\infty(s) \in L(B,p)$, then there exists an accepting run $\rho \in \mathrm{Run}_B(\gamma_\infty(s), p)$ and Prover can react to all choices of Spoiler by considering the run ρ. Note that due to the second component $c_\rho(i_j)$ in the states, every infinite game α corresponds to a unique infinite directed path in $\rho(\gamma_\infty(s))$ such that the minimal color occurring infinitely often on that infinite path is the same color as the the minimal color occurring infinitely often in game α.

For the other direction, let us assume that Prover has a winning strategy in the arena $G(A,s)$ at vertex $(\varepsilon, (p, \chi(p)))$. Then Prover P_0 has a positional winning strategy by [17].

We will show that this winning strategy of P_0 defines an accepting run $\rho_0 \in \mathrm{Run}_B(\gamma_\infty(s), p)$. The positional strategy chosen by Prover defines for each vertex belonging to P_0 at most one outgoing edge, all other outgoing edges at that vertex are deleted from the arena. After the deletion of edges, Prover is released and without any further interaction in the game. The game becomes a solitaire game where the outcome depends only on the choices of Spoiler.

Henceforth, all games are defined in a subarena $G'(A,s)$ where all games at vertex $(\varepsilon, (p, \chi(p)))$ are won by Prover. Moreover, we may assume that all vertices in $G'(A,s)$ are reachable from the vertex $(\varepsilon, (p, \chi(p)))$. Thus, without restriction, every vertex in $G'(A,s)$ belonging to Prover has exactly one outgoing edge. In particular, since all nodes are reachable from $(\varepsilon, (p, \chi(p)))$, all finite games starting at any vertex in $G'(A,s)$ are won by Prover, and if α is any directed infinite path starting at any vertex in $G'(A,s)$, then the minimal color occurring infinitely often at vertices of α is even.

Whenever a game reaches a vertex $(u, (q,c))$ via a finite directed path in $G'(A,s)$ starting at $(\varepsilon, (p, \chi(p)))$, then $\mathrm{Run}_{B_H}(\gamma(s(u)), q) \neq \emptyset$, and the unique outgoing edge defines a run $\rho \in \mathrm{Run}_{B_H}(\gamma(s(u)), q) \neq \emptyset$ which depends on (u,q,c), only. For better reading we we denote the run ρ as $\rho(u,q,c)$, too[11]. In particular, if $(q,c) = (\rho(i_j), c_\rho(i_j))$, then the run $\rho(u,q,c)$

[11] Thus, the same letter ρ denotes a function or the specific value of that function. The context makes clear what we mean.

does not reveal the position of the leaf i_j, in general. The outgoing edge (which defines ρ) ends at the vertex $(u,(q,c),\rho)$ belonging to spoiler.

In the following we assume without restriction that all games α start at vertices belonging to P_0. Moreover, if α is finite, then α ends in a vertex belonging to Spoiler.

Our aim is construct an accepting run of $\gamma_\infty(s)$. The run is constructed top-down by induction. We use the following notation. Let $u \in \mathrm{Pos}(s)$. We say that $\rho \in \mathrm{Run}_{B_H}(\gamma(s(u)),q)$ is a *partial run* of the subtree $\gamma_\infty(s|_u)$ if the run labels a prefix closed subset of positions in $\gamma_\infty(s|_u)$ with states. The partial run labels the root with q. (Recall that $(s|_u)$ denotes the subtree of s rooted at u.) A *path sequence* (or *choice sequence*) for Spoiler is a finite or infinite sequence $g(\alpha) = (g_1, g_2, \ldots)$ where each g_ℓ is of the form $g_\ell = (i_\ell)_{j_\ell}$ and which results from a game α in $G'(A,s)$ starting at $(\varepsilon,(p,\chi(p))$ recording the choices of Spoiler. Thus, if α is a game, then it defines a unique path sequence $g(\alpha)$ and knowing $g(\alpha)$ we can recover α due to the fact that every edge chosen by prover is fully determined by its source, and the assumption that the last vertex on the path defined by a finite game belongs to Spoiler.

Let $C_0(s)$ be set of all path sequences of Spoiler. The index 0 reflects that the definition of $C_0(s)$ depends on the positional strategy of Prover. In the next step we define for each finite prefix $g = (g_1, \ldots, g_k)$ of a path sequence $g(\alpha) \in C_0(s)$ several items.

- A path w_g in $\gamma_\infty(s)$. The other items are defined through the path w_g.
- A position $v(w_g)$ in $\gamma_\infty(s)$.
- A "terminal" position $t(w_g) = (u,(q,c))$ in $V_0 = \mathrm{Pos}(s) \times (Q \times C)$ such that the tree $\gamma(s(u))$ appears as an IO-prefix of the subtree $\gamma_\infty(s|_u)$.
- For $t(w_g) = (u,(q,c))$ a partial run $\rho_g \in \mathrm{Run}_{B_H}(\gamma(s(u)),q)$ of the subtree $\gamma_\infty(s|_u)$.

If g is the empty sequence, then no choice of Spoiler has been recorded. We let w_g be the empty path, $v(w_g) = \varepsilon \in \mathrm{Pos}(\gamma_\infty(s))$, and $t(w_g) = (\varepsilon,(p,\chi(p)) \in V_0$. The game starts at that vertex $(\varepsilon,(p,\chi(p))$. The set E_0 defines a run $\rho_1 \in \mathrm{Run}_{B_H}(\gamma(s(\varepsilon)),p)$. If no hole appears in $\gamma(s(\varepsilon))$, then $\gamma_\infty(s)$ is equal to $\gamma(s(\varepsilon))$. We are done in this case: with $\rho_\varepsilon = \rho_1$ all four items are defined if g is the empty sequence.

In the other case there exist some $i \in H$ and $i_j \in \mathrm{leaf}_i(\gamma(s(u)))$. Let $g = (g_1, \ldots, g_k)$ being a prefix of $g' = (g_1, \ldots, g_{k+1})$. By induction, assume that w_g is defined for $g = (g_1, \ldots, g_k)$. Let $t(w_g) = (u,(q,c))$, and $x = s(u)$. Hence, there is a tree $t_u = \gamma(x)$, and, by induction, we may assume that t_u is an IO-prefix of the subtree $\gamma_\infty(s)|_v$ of $\gamma_\infty(s)$ rooted at the position $v = v(w_g)$ in $\gamma_\infty(s)$. In particular, every position v in t_u can be identified with a unique position $\mathrm{pos}(v) \in \mathrm{Pos}(\gamma_\infty(s))$ by the endpoint of the path $w_g.w_u(v)$. Here, $w_u(v)$ denotes the unique directed path in t_u from the root to the position v in t_u. The choice g_{k+1} of spoiler defines a position $i_j \in \mathrm{leaf}_i(t_u)$ for some $i \in [\mathrm{rk}(x)]$. We let $w_{g'}$ be the unique path which starts at the the root has w_g as prefix and stops in $\mathrm{pos}(i_j) \in \mathrm{Pos}(\gamma_\infty(s))$. Then we define $v(w_{g'}) = \mathrm{pos}(i_j)$. It is the endpoint of the path $w_{g'}$. The unique outgoing edge at $t(w_g) = (u,(q,c))$ defines a run $\rho_{g'} \in \mathrm{Run}_{B_H}(\gamma(s(u)),q)$. Since g_{k+1} is defined, the position $u.i \in \mathrm{Pos}(s)$ exists. and we let $t(w_{g'}) = (u.i,(\rho'(i_j), c_{\rho'}(i_j)))$ where $\rho' = \rho_{g'}$. Note that the tree $\gamma(s(u.i))$ appears as an IO-prefix of the subtree $\gamma_\infty(s|_{u.i})$. Thus, all desired items are defined for the sequence g'. Moreover, the root position is labeled by some element in Σ.

See Fig. 7 how a finite directed path w in $\gamma^s_\infty(s)$ starting at the root defines a corresponding path in s. The figure also shows that w defines a unique position in $v(w) \in \mathrm{Pos}(\gamma^s_\infty(s))$ via the arrows which start in $\gamma^s_\infty(s)$, leave the tree on the left, and make a clockwise turn upwards to enter the tree from the right. This visualizes the formal definition of a path w_g and its position $v(w_g)$ in $\gamma_\infty(s)$.

Suppose that $t(w) = (u,(q,c))$ and $t(w') = (u,(q',c'))$ where $w = w_g$ and $w' = w_{g'}$ are finite prefixes of path sequences for Spoiler. We claim that if $(q,c) \neq (q',c')$, then the positions of $v(w)$ and $v(w')$ are incomparable. That is, there is no directed path between them. The claim can be easily shown by induction on $|w|$. The runs $\rho(u,q,c) \in \mathrm{Run}_{B_H}(\gamma(s(u)),q)$ and $\rho(u,q',c') \in \mathrm{Run}_{B_H}(\gamma(s(u)),q)$ might be different, for example because $c \neq c'$. Since the positions $v(w)$ and $v(w')$ are incomparable, we can use $\rho(u,q,c)$ to define a partial run of the subtree rooted at $v(w) \in \mathrm{Pos}(\gamma_\infty(s))$ and we can use $\rho(u,q',c')$ to define a partial run of the subtree rooted at $v(w')$.

Let $n \in \mathbb{N}$. Consider all finite directed paths w in $\gamma_\infty(s)$ beginning at the root such that $|w| \leq n$, and among them those which are induced by finite prefixes of path sequences g of Spoiler. Let call them w_g. Then the the runs ρ_g constructed above for w_g yield partial runs $\widetilde{\rho}_n$ of $\gamma_\infty(s)$). Note that these partial runs are compatible: for every common positions of two such partial runs, the labels (being states) are the same. This follows because $w_g = w_{g'}$ implies $\rho_g(v(w_g)) = \rho_{g'}(v(w_{g'}))$. Moreover, the sequence $(\widetilde{\rho}_n)_{n\in\mathbb{N}}$ has a well-defined limit $\rho_0 = \lim_{n\to\infty}\widetilde{\rho}_n$. Indeed, for every $k \geq 0$ there is some n_k such that $\widetilde{\rho}_n(v) = \widetilde{\rho}_{n_k}(v)$ for all $n \geq n_k$ and all positions $v \in \mathrm{Pos}(\gamma_\infty(s))$ having a distance less than k to the root.

Therefore, the set of all directed paths in the arena $G'(A,s)$ starting at $(\varepsilon,(p,\chi(p)))$ yields ρ_0 as a (totally defined) run $\rho_0 : \mathrm{Pos}(\gamma_\infty) \to Q$ such that $\rho_0(\varepsilon) = p$. (We write ρ_0 because its definition depends on the positional winning strategy of P_0.)

We have to show that the run ρ_0 is accepting. For that we have to consider all directed paths in the tree defined by ρ_0. If a path ends at a leaf, then it is accepting because ρ_0 is a run. Every infinite path in ρ_0 is due to some infinite game α starting at $(\varepsilon,(p,\chi(p)))$. The game defines a path sequence $g(\alpha)$ and the run ρ_0 labels all vertices on the unique infinite direct path β which visits all positions $v(w_{g_k})$ where g_k runs over all finite prefixes of g. By construction, the minimal color $c(\alpha)$ appearing infinitely often in the game α is the same color as occurring infinitely often on β. Since Prover wins all games, $c(\alpha)$ is even. Hence, the run ρ_0 is accepting. Thus, the assertion of the lemma. ◀

6 Main results

The following lemma generalizes Lem. 28 by switching from a homomorphism γ to a regular substitution σ.

▶ **Lemma 29.** *Let $R \subseteq T(\Sigma)$ be regular and $\sigma : \mathcal{X} \to 2^{T(\Sigma\cup H)\setminus H}$ be a regular substitution. Then the set $\sigma_{\mathrm{io}}^{-1}(R) = \{s \in T(\Sigma \cup \mathcal{X}) \mid \sigma_{\mathrm{io}}(s) \subseteq R\}$ is effectively regular.*

Proof. Let us begin to prove the lemma with the corresponding notation Σ', R', and σ'. Then, without restriction, there is some constant $a \in \Sigma$ such that $\Sigma' \subseteq \Sigma \setminus \{a\}$ and $R' \cup \sigma'(x) \subseteq T(\Sigma' \cup H) \setminus (H \cup \{a\})$ for all $x \in \mathcal{X}$. Define $\sigma(x) = \sigma'(x) \cup \{a\}$ for all $x \in \mathcal{X}$ and $R = R' \cup T(\Sigma) \setminus T(\Sigma')$. Then for all $s \in T(\Sigma')$ we have

$$\sigma'_{\mathrm{io}}(s) \subseteq R' \iff \sigma_{\mathrm{io}}(s) \subseteq R \iff s \in \sigma_{\mathrm{io}}^{-1}(R).$$

Moreover, if $\sigma_{\mathrm{io}}^{-1}(R)$ is regular, then $\sigma_{\mathrm{io}}^{-1}(R) \cap T(\Sigma')$ is regular, too. Thus, it is enough to prove the lemma under the assumptions that first, $\sigma(x) \neq \emptyset$ for all $x \in \mathcal{X}$ and second, there is a constant $a \in \Sigma$ such that $T(\Sigma) \setminus T(\Sigma \setminus \{a\}) \subseteq R$.

Let $R = L(B,q_0)$ for some parity-NTA B. According to Sec. 4.2 we embed the NTA B into the parity-NTA $B_{\mathcal{P}}$ and $T(\Sigma)$ into $T(\Sigma_{\mathcal{P}})$ such that both, $R = L(B_{\mathcal{P}},q_0)$ and for every profile $\pi = \pi(t) \in \mathcal{P}_B$ there is some finite tree $t_\pi \in T_{\mathrm{fin}}(\Sigma_{\mathcal{P}})$ satisfying $\pi(t_\pi) = \pi$. Since $T(\Sigma) \setminus T(\Sigma \setminus \{a\}) \subseteq R$, we may assume without restriction that $a \in L(B,p)$ for all states p. Hence, we have $\pi(a) = \mathcal{T}_B$: all tasks are satisfied by the constant a. It follows $\mathcal{T}_B \in \mathcal{P}_B \subseteq \mathcal{P}_{B_{\mathcal{P}}}$.

In the following let $\widetilde{R} = L(B_{\mathcal{P}}, q_0)$. Since $\sigma(x)$ is regular for all $x \in \mathcal{X}$ we can compute the specialization $\check{\sigma}$ of σ as defined in Def. 24. Using Eq.(25) in Prop. 25 we can state

$$\forall s \in T(\Sigma \cup \mathcal{X}):\ \sigma_{\text{io}}(s) \subseteq R \iff \sigma_{\text{io}}(s) \subseteq \widetilde{R} \iff \check{\sigma}_{\text{io}}(s) \subseteq \widetilde{R} \tag{31}$$

Then, using Prop. 16, the line (31) implies

$$\forall s \in T(\Sigma \cup \mathcal{X}):\ \sigma_{\text{io}}(s) \subseteq R \iff \check{\sigma}_{\text{io}}(s) \subseteq \widetilde{R} \tag{32}$$

Define a new set of variables by

$$\mathcal{X}' = \{(x, \pi) \in (\Sigma \cup \mathcal{X}) \times \mathcal{P}_B \mid \exists t \in \sigma(x):\ \pi = \pi(t)\}\,. \tag{33}$$

We also write x_π to denote the variable $(x, \pi) \in \mathcal{X}'$ and we let $\text{rk}(x_\pi) = \text{rk}(x)$. We use the second component in $x_\pi \in \mathcal{X}'$ to define a homomorphism $\gamma : \mathcal{X}' \to T_{\text{fin}}(\Sigma_{\mathcal{P}} \cup H) \setminus H$ with $\gamma(x_\pi) = t_\pi$. (Recall that t_π was defined as a finite tree such that $\pi = \pi(t_\pi)$.) Then, by Eq.(24), for every $x \in X$ the following equation holds.

$$\check{\sigma}(x) = \{\gamma(x_\pi) \mid \exists t \in \sigma(x) : \pi = \pi(t)\}\,. \tag{34}$$

Since $\sigma(x) \neq \emptyset$, the projection onto the first component $x_\pi \mapsto x$ defines a surjective homomorphism by

$$h : \mathcal{X}' \to T(\Sigma \cup \mathcal{X} \cup H) \setminus H,\ x_\pi \mapsto x(1, \ldots, \text{rk}(x)). \tag{35}$$

Note that $\text{Pos}(s') = \text{Pos}(h_{\text{io}}(s'))$ for all $s' \in T(\mathcal{X}')$. Since h is a totally defined homomorphism, we simply write henceforth $h(s')$ instead of $h_{\text{io}}(s')$. Also note that $\check{\sigma}(s) = \gamma_{\text{io}}(h^{-1}(s))$ for all $s \in T(\Sigma \cup \mathcal{X})$ by Eq.(34). Therefore,

$$\begin{aligned}
\sigma_{\text{io}}^{-1}(R) &= \check{\sigma}_{\text{io}}^{-1}(\widetilde{R}) && \text{by (32)}\\
&= \left\{s \in T(\Sigma \cup \mathcal{X}) \,\middle|\, \gamma_{\text{io}}(h^{-1}(s)) \in R\right\}\\
&= \left\{s \in T(\Sigma \cup \mathcal{X}) \,\middle|\, h^{-1}(s) \in \overset{-1}{\gamma_{\text{io}}}(R)\right\}.
\end{aligned}$$

Thus, $\sigma_{\text{io}}^{-1}(R)$ is regular if and only if the set $S = \left\{s \in T(\Sigma \cup \mathcal{X}) \,\middle|\, h^{-1}(s) \in {\gamma_{\text{io}}}^{-1}(R)\right\}$ is regular. Let $\overline{S} = T(\Sigma \cup \mathcal{X}) \setminus S$ and $\overline{R} = T(\Sigma) \setminus \widetilde{R}$. A purely set theoretical consideration shows that

$$\overline{S} = \{s \in T(\Sigma \cup \mathcal{X}) \mid h^{-1}(s) \cap \overset{-1}{\gamma_{\text{io}}}(\overline{R}) \neq \emptyset\} = h(\overset{-1}{\gamma_{\text{io}}}(\overline{R})) = h(T(\mathcal{X}') \setminus \overset{-1}{\gamma_{\text{io}}}(\widetilde{R})). \tag{36}$$

The set $T(\mathcal{X}') \setminus {\gamma_{\text{io}}}^{-1}(\widetilde{R})$ is effectively regular because ${\gamma_{\text{io}}}^{-1}(\widetilde{R})$ has this property. Thus, we can write $T(\mathcal{X}') \setminus {\gamma_{\text{io}}}^{-1}(\widetilde{R}) = L(A', p_0)$ for a parity-NTA $A' = (Q, \mathcal{X}', \delta', \chi)$. It is therefore enough to construct a parity-NTA $A = (Q, \Sigma \cup \mathcal{X}, \delta, \chi)$ such that $L(A, p_0) = h(L(A', p_0))$. The construction of A is a straightforward by using another transition relation. We define δ by the following equivalence.

$$(p, x, q_1, \ldots, q_r) \in \delta \iff \exists \pi \in \mathcal{P}_B : (p, x_\pi, q_1, \ldots, q_r) \in \delta'. \tag{37}$$

The assertion follows. ◀

In the following, $\mathcal{C}$ denotes a class of tree languages containing for every finite ranked alphabet Δ all regular tree languages in $T(\Delta)$ and such that on input $L \in \mathcal{C}$ (given in some effective way) and a regular tree language $K \subseteq T(\Delta)$ (given, say, by some parity-NTA) the problem "$L \subseteq K$?" is decidable.

In order to see that $\mathcal{C}$ can be chosen strictly larger than the class of all regular languages, we let $\mathcal{C}$ contain, in addition to the class of the regular languages in each $T(\Delta)$, all context-free languages over $T_{\text{fin}}(\Delta)$ as defined in [7, 11, 16, 25].

▶ **Theorem 30.** *Let $\mathcal{C}$ be a class of tree languages as above. Then the following decision problem is decidable.*

- *Input: Regular substitutions σ_1, σ_2 and tree languages $L \subseteq T(\Sigma \cup \mathcal{X})$, $R \subseteq T(\Sigma)$ such that R is regular and $L \in \mathcal{C}$.*
- *Question: Is there some substitution $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H) \setminus H}$ satisfying both, $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma_1 \leq \sigma \leq \sigma_2$?*

Moreover, we can effectively compute the set of maximal substitutions σ satisfying $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma_1 \leq \sigma(x) \leq \sigma_2$. It is a finite set of regular substitutions.

Proof. First, we check that $\sigma_1(x) \subseteq \sigma_2(x)$ for all $x \in \mathcal{X}$ because otherwise there is nothing to do. Moreover, without restriction we have $\sigma_1(f) = \sigma_2(f) = f(1, \ldots, [\text{rk}(f)])$ for all $f \in \Sigma \setminus \mathcal{X}$. Thus, $\sigma_1(x)$ and $\sigma_2(x)$ are defined as regular sets for all $x \in \Sigma \cup \mathcal{X}$ with $\sigma_1 \leq \sigma_2$. If there is any substitution $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H) \setminus H}$ satisfying both, $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma_1 \leq \sigma \leq \sigma_2$, then σ_1 is the unique **minimal** substitution satisfying that property. The substitution σ_1 belongs to the finite set of regular substitutions S_2 as defined in (23) in the proof of Cor. 19. Thus, if the set S_2 is empty, then we can stop. The answer to the decision problem is negative and, therefore, the set of maximal substitutions σ satisfying $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma_1 \leq \sigma(x) \leq \sigma_2$ is empty.

Thus, we may assume $S_2 \neq \emptyset$. Next, in a second phase we consider each element $\sigma \in S_2$, one after another. By Lem. 29 we know that for each $\sigma \in S_2$, the set $\sigma_{\text{io}}^{-1}(R)$ is an effectively regular set. The assertions $\sigma_{\text{io}}(L) \subseteq R$ and $L \subseteq \sigma_{\text{io}}^{-1}(R)$ are equivalent. Since $L \in \mathcal{C}$ we can check $L \subseteq \sigma_{\text{io}}^{-1}(R)$. Thus, we end up with an effectively computable subset S_2' of S_2 such that there is some substitution $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H) \setminus H}$ satisfying both, $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma_1(x) \subseteq \sigma(x) \subseteq \sigma_2(x)$ for all $x \in \mathcal{X}'$ if and only if $S_2' \neq \emptyset$. Moreover, the maximal substitutions with that property are in S_2', too[12]. We are done. ◀

▶ Remark 31. Thm. 30 constitutes a borderline of decidability in the following sense. Assume $\mathcal{C}'$ is any class of tree languages which contains all regular tree languages and there is some Σ where the decision problem "$L \subseteq K$?" becomes undecidable if (L, K) runs over all pairs where $K \subseteq T(\Sigma)$ is regular and L belongs to $\mathcal{C}'$. Suppose that $\{L \subseteq T(\Sigma) \mid L \in \mathcal{C}'\} \subseteq \{L \subseteq T(\Sigma \cup \mathcal{X}) \mid L \in \mathcal{C}'\}$. Then Thm. 30 does not hold if we replace the class $\mathcal{C}$ by the class $\mathcal{C}'$. The decision problem becomes undecidable in the special case where $L \subseteq (\Sigma \cup \mathcal{X})$ is chosen to be a subset of $T(\Sigma)$. The undecidability follows because $L = \sigma_{\text{io}}(L)$ for every σ if $L \subseteq T(\Sigma)$.

▶ **Corollary 32.** *Let $\mathcal{C}$ be a class of tree languages as above. Then the following decision problem is decidable.*

- *Input: Tree languages $L \subseteq T(\Sigma \cup \mathcal{X})$, $R \subseteq T(\Sigma)$ such that R is regular and $L \in \mathcal{C}$.*
- *Question: Is there some substitution $\sigma : \mathcal{X} \to 2^{T(\Sigma \cup H) \setminus H}$ satisfying both, $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma(x) \neq \emptyset$ for all $x \in \mathcal{X}$?*

Moreover, we can effectively compute the set of maximal substitutions σ satisfying $\sigma_{\text{io}}(L) \subseteq R$ and $\sigma(x) \neq \emptyset$ for all $x \in \mathcal{X}$. It is a finite set of regular substitutions.

Proof. Without restriction we have $R = L(B, q_0)$ for some parity-NTA B. We run the following nondeterministic decision procedure. For each $x \in \mathcal{X}$ we guess a profile $\pi \in \mathcal{P}_B$.

[12] Since S_2' is a finite set of regular substitutions, we can compute the maximal elements in S_2'.

Then we check that there is some $t \in T(\Sigma \cup [\mathrm{rk}(x)]) \setminus H$ such that $\pi = \pi(t)$. If there is no such t, then the assertion in the corollary has a negative answer for that guess. If there exists such t, then we let $\sigma_1(x) = \{t \in T(\Sigma \cup [\mathrm{rk}(x)]) \setminus H \mid \pi = \pi(t)\}$ and $\sigma_2(x) = T(\Sigma \cup [\mathrm{rk}(x)]) \setminus H$. We now apply Thm. 30. The assertion in the corollary is positive if and only if for some guess the application of Thm. 30 yields a positive answer with a set of maximal solutions. From these data we can compute the set of maximal solutions where $\sigma(x) \neq \emptyset$ for all $x \in \mathcal{X}$. ◀

7 Conclusion and open problems

The main result of the paper is Thm. 30. We included the special case Cor. 32 because the assertion in the corollary corresponds exactly to the results for regular languages over finite words due to Conway as stated in the introduction, Sec. 1. The assertions of Thm. 30 and Cor. 32 are positive decidability results, but we don't know any (matching) lower and upper complexity bounds up to a few special cases.

Various other natural problems are open, too. For example, a remaining question is whether it is possible to derive positive results for the outside-in extension σ_{oi}. Another puzzling problem is that we we don't know how to decide for regular tree languages L and R whether there exists a substitution σ such that $\sigma_{\mathrm{io}}(L) = R$. We have seen that if such a substitution σ exists, then σ is in a finite and effectively computable set of regular substitutions. Thus, the underlying problem has no existential quantifier: decide "$\sigma_{\mathrm{io}}(L) = R$?" on input L, R, and σ where L, R, and σ are regular. If the answer is *yes*: $\sigma_{\mathrm{io}}(L) = R$, then $\sigma_{\mathrm{io}}(L)$ is regular. So, one could try to solve that problem. Recall that the problem is decidable in the setting of finite trees and homomorphisms by [9, 14].

Actually, we ignore to decide the problem "$\sigma_{\mathrm{io}}(L) = T(\Sigma)$?" where $L \subseteq T(\mathcal{X})$ and $\sigma(x) = T(\Sigma \cup [\mathrm{rk}(x)])$ for all $x \in \mathcal{X}$. Both problems "$\exists\sigma : \sigma_{\mathrm{io}}(L) = R$?" and "$\exists\sigma : \sigma_{\mathrm{oi}}(L) \subseteq R$?" remain open, even if we restrict ourselves to deal with finite trees, only.

It is also open whether better results are possible if we restrict R (or L and R) to smaller classes of regular tree languages like the class of languages with Büchi acceptance.

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 Julius 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 Hubert Comon, Max Dauchet, Rémy Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc 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`, `doi:10.1016/0304-3975(78)90039-7`.

8 Bruno Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25:95–169, 1983. URL: `https://doi-org.docelec.u-bordeaux.fr/10.1016/0304-3975(83)90059-2`, `doi:10.1016/0304-3975(83)90059-2`.

9 Carles Creus, Adrià Gascón, Guillem Godoy, and Lander Ramos. The HOM problem is EXPTIME-complete. *SIAM J. Comput.*, 45:1230–1260, 2016. `doi:10.1137/140999104`.

10 Volker Diekert, Manfred Kufleitner, Gerhard Rosenberger, and Ulrich Hertrampf. *Discrete Algebraic Methods. Arithmetic, Cryptography, Automata and Groups.* Walter de Gruyter, 2016.

11 Joost Engelfriet and Erik Meineche Schmidt. IO and OI. I. *J. Comput. Syst. Sci.*, 15:328–353, 1977. `doi:10.1016/S0022-0000(77)80034-2`.

12 Joost Engelfriet and Erik Meineche Schmidt. IO and OI. II. *J. Comput. Syst. Sci.*, 16:67–99, 1978. `doi:10.1016/0022-0000(78)90051-X`.

13 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`.

14 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`.

15 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. `doi:10.1007/3-540-36387-4`.

16 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`.

17 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. `doi:10.1145/800070.802177`.

18 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.

19 Michal Kunc. The power of commuting with finite sets of words. *Theory of Computing Systems*, 40:521–551, 2007.

20 Michal Kunc and Alexander Okhotin. Language equations. In Jean-Éric Pin, editor, *Handbook of Automata*, pages 689–722. European Mathematical Society, To Appear in 2021 or later. Final version (2018) available at the homepage of Okhotin.

21 David E. Muller and Paul E. Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987. `doi:10.1016/0304-3975(87)90133-2`.

22 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. `doi:10.1016/0304-3975(94)00214-4`.

23 Christophe Prieur, Christian Choffrut, and Michel Latteux. Constructing sequential bijections. In *Structures in logic and computer science*, volume 1261 of *Lecture Notes in Computer Science*, pages 308–321. Springer, Berlin, 1997. URL: `https://doi-org.docelec.u-bordeaux.fr/10.1007/3-540-63246-8_19`, `doi:10.1007/3-540-63246-8_19`.

24 Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

25 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`, `doi:10.1016/0022-0000(85)90002-9`.

26 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.

27 Wiesław Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200:135–183, 1998. doi:10.1016/S0304-3975(98)00009-7.

8 Appendix: some additional material

As a preamble let us stress that it is not necessary to read anything in the appendix to understand the main body of the paper. C'est l'art pour l'art: ars gratia artis.

8.1 Conway's result for finite and infinite words

This section is meant for readers who are not familiar with notion of *syntactic congruences* or *recognizing morphisms* for words, but still they might be interested to understand our main results either for finite words or for both finite and infinite words. Once, the case of words is understood, the forest is hopefully not hidden behind trees. Let Σ and $\mathcal{X}$ be be finite alphabets and let Σ^∞ denote the set of finite and infinite words. The aim is to give a self-contained and elementary proof of Conway's result with respect to $\# \in \{\subseteq, =\}$ for finitely generated free monoids. Actually, let us give such a proof for "regular constraints" in the spirit of Thm. 30. (Once, we have seen the case of finite words, we explain that essentially the same approach works for infinite words, too.)

▶ **Proposition 33.** *Let σ_1, σ_2 be mappings from $\Sigma \cup \mathcal{X}$ to 2^{Σ^+} such that for $i = 1, 2$ first, $\sigma_1(x) = \{x\}$ for every $x \in \Sigma \setminus \mathcal{X}$ and second, $\sigma_i(x)$ is regular language every $x \in \mathcal{X}$. Then for regular languages $L \subseteq (\Sigma \cup \mathcal{X})^\infty$ and $R \subseteq \Sigma^\infty$ the following assertions hold for $\# \in \{\subseteq, =\}$.*

1. *The following decision problem is decidable.*

$$\text{“}\exists \sigma : \sigma(L) \# R \wedge \forall x : \sigma_1(x) \subseteq \sigma(x) \subseteq \sigma_2(x)\text{?”} \tag{38}$$

2. *Define $\sigma \le \sigma'$ by $\sigma(x) \subseteq \sigma'(x)$ for all $x \in \mathcal{X}$. Then every solution σ of (38) is upper bounded by a maximal solution; and the number of maximal solutions is finite. Moreover, if σ is maximal, then $\sigma(x)$ is regular for all $x \in \mathcal{X}$; and all maximal solutions are effectively computable.*

Proof. 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. The easiest situation is the setting of Conway in the case where $L \subseteq (\Sigma \cup \mathcal{X})^*$ and $R \subseteq \Sigma^*$. In that case let $A = (Q, \Sigma, \delta, I, F)$ be an NFA such that $\delta \subseteq Q \times \Sigma \times Q$ and $R = L(A)$. Without restriction, $Q = \{1, \dots, n\}$ with $n \ge 1$. 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 $\mu : \Sigma^* \to \mathbb{B}^{n \times n}$ (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 $\sigma : \mathcal{X} \to 2^{\Sigma^*}$ be any substitution such that $\sigma(L) \# R$. Define $\widehat{\sigma} : \mathcal{X} \to 2^{\Sigma^*}$ by $\widehat{\sigma}(x) = \mu^{-1}(\mu(\sigma(x)))$. Then $\sigma \le \widehat{\sigma}$ and $\widehat{\sigma}(L) \# R$, 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 $\sigma_1 \le \sigma_2$, we compute for each candidate σ another substitution $\widetilde{\sigma}$ by $\widetilde{\sigma}(x) = \sigma(x) \cap \sigma_2(x)$. The point is that whenever the problem in (38) has any solution $\sigma : \mathcal{X} \to 2^{\Sigma^*}$, then there is a maximal solution $\widetilde{\sigma}$ from the list of candidates which satisfies $\widetilde{\sigma}(L) \# R$. The class of regular languages is closed under substitution of letters by regular sets. This is a standard exercise in formal language theory. Hence, $\widetilde{\sigma}(L)$ is regular; and we can decide $\widetilde{\sigma}(L) \# R$.

Now, let us show that the case of infinite words can be explained 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. 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 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 $S = (\{0,1,2\}, +, \cdot)$ where $+ = \max$ and $x \cdot y = 0$ if $x = 0$ or $y = 0$ and otherwise $x \cdot y = \max\{x, y\}$. Note that 0 is a zero and 1 is neutral in $(S, \cdot)$. Let us define for $a \in \Sigma$ the matrix $M_a \in S^{n\times n}$ by

$$M_a(p,q) = \begin{cases} 0 \text{ if } (p,a,q) \notin \delta, \\ 1 \text{ if } (p,a,q) \in \delta \text{ but } \{p,q\} \cap F = \emptyset, \\ 2 \text{ otherwise: if } (p,a,q) \in \delta \text{ and } \{p,q\} \cap F \neq \emptyset. \end{cases}$$

The matrices $M_a \in S^{n\times n}$ define a homomorphism $\mu : \Sigma^* \to S^{n\times n}$. 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, $M_w(p,q) = 2$ if and only if there is path labeled by w from state p to q which visits a final state. The tricky fact is that at the end we have to decide the problem "$\widetilde{\sigma}(L) \mathrel{\#} R$?". It is here where Büchi's result comes into the play. Using Ramsey theory shows decidability of that problem. If we take [4] as a blackbox, then Conway's result for infinite words is essentially as easy as for finite words. For more details about regular languages over infinite words we refer to the textbook [10]. ◀

8.2 The outside-in extension $\sigma_{\mathrm{oi}}(s)$ for an infinite tree s

Let us define $\sigma_{\mathrm{oi}}(S)$ for a set of trees which can be finite or infinite. We content ourselves to define $\sigma_{\mathrm{oi}}(S)$ for $S \subseteq T(\mathcal{X})$ where $\Sigma \cap \mathcal{X} = \emptyset$ and $\sigma(x) \subseteq T(\Sigma \cup [\mathrm{rk}(x)]) \setminus H$ for all $x \in \mathcal{X}$. The first step reduces the problem to define $\sigma_{\mathrm{oi}}(S)$ for a set S to the case where S is a singleton simply be letting $\sigma_{\mathrm{oi}}(S) = \bigcup \{\sigma_{\mathrm{oi}}(s) \mid s \in S\}$. Thus, it is enough to define $\sigma_{\mathrm{oi}}(s)$ for an infinite tree $s \in T(\mathcal{X})$ because for a finite tree s we employ the definition in Eq.(6). The idea is to use a nondeterministic program which transforms $s \in T(\mathcal{X})$ into a tree in $T(\mathcal{X} \cup \Sigma)$. Each run of the program defines at most one output. The set of all outputs over all runs defines the set $\sigma_{\mathrm{oi}}(s)$. The program does not need to terminate, but if it runs forever, then, in the limit, it defines (nondeterministically) a tree $t \in T(\Sigma)$.

The nondeterministic program is denoted as "$\sigma_{\mathrm{oi}}^{nd}$". The input for the program is any tree $s \in T(\Sigma \cup \mathcal{X})$. We initialize a tree variable $t := s$. Then we perform the following while-loop as long as $t \notin T(\Sigma)$.

1. Choose in the breadth-first order on $\mathrm{Pos}(t)$ the first position v such that $x = s(v) \in \mathcal{X}$.
2. Denote the subtree $t|_v$ rooted at v as $t|_v = x(t_1, \ldots, t_r)$. This implies $\mathrm{rk}(x) = r$.
3. Choose nondeterministically some $t_x \in \sigma(x)$.
   If this is not possible because $\sigma(x) = \emptyset$, then EXIT without any output.
4. Replace in t the subtree $t|_v$ by $t_x[i_j \leftarrow t_i]$. Recall that this means to replace in t_x every $i_j \in \mathrm{leaf}_i(t_x)$ by the same tree t_i. Moreover, if $\mathrm{leaf}_i(t_x) \neq \emptyset$, then $1 \leq i \leq \mathrm{rk}(x)$.

Do not confuse the procedure with an inside-out extension. We visit positions in t labeled by a variable one after another and later choices of trees in $\sigma(t(v))$ are fully independent of each other. If the program terminates without using the EXIT branch, then we return the final tree t as the output $t = \sigma_{\mathrm{oi}}^{nd}(s)$ of the specific nondeterministic run. If the program runs forever, then let t_n be the value of t after performing the n-th loop. Since we always have $t_x \in T(\Sigma \cup H) \setminus H$, an easy reflection shows that it exists a unique infinite tree $\sigma_{\text{oi}}^{nd}(s) = \lim_{n\to\infty} t_n$. Moreover, $\sigma_{\text{oi}}^{nd}(s) \in T(\Sigma)$ and Eq.(6) holds.

8.3 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 $[y]$. The path can be arbitrary long and still have weight 0. Clearly, $d_\sim$ is a pseudometric satisfying

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

If for each $[x]$ there exists $x_0 \in [x]$ such that for all y we have

$$0 < g([x],[y]) = \inf\{d(x_0,y') \mid y \sim y'\}$$

then $g([x],[y]) = d_\sim([x],[y])$ is a metric.

In our situation, we have $(M,d) = (T(\Omega_\perp), d)$ and $M/{\sim}$ results by identifying all trees $s \in T(\Omega_\perp), d)$ where some position in $\mathrm{Pos}(s)$ is labeled by $\perp$. To see this, just recall our definition of $d(s,s')$ for trees in $T(\Omega_\perp)$:

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