The HOM Problem Is Decidable

GUILLEM GODOY and OMER GIMÉNEZ, Universitat Politècnica de Catalunya

We close affirmatively a question that has been open for long time: decidability of the HOM problem. The HOM problem consists in determining, given a tree homomorphism H and a regular tree language L represented by a tree automaton, whether $H(L)$ is regular. In order to decide the HOM problem, we develop new constructions and techniques that are interesting by themselves, and provide several significant intermediate results. For example, we prove that the universality problem is decidable for languages represented by tree automata with equality constraints, and that the equivalence and inclusion problems are decidable for images of regular languages through tree homomorphisms. Our contributions are based on the following new constructions. We describe a simple transformation for converting a tree automaton with equality constraints into a tree automaton with disequality constraints recognizing the complementary language. We also define a new class of tree automata with arbitrary disequality constraints and a particular kind of equality constraints. An automaton of this new class essentially recognizes the intersection of a tree automaton with disequality constraints and the image of a regular language through a tree homomorphism. We prove decidability of emptiness and finiteness for this class by a pumping mechanism. We combine the above constructions adequately to provide an algorithm deciding the HOM problem. This is the journal version of a paper presented in the 42nd ACM Symposium on Theory of Computing (STOC 2010). Here, we provide all proofs and examples. Moreover, we obtain better complexity results via the modification of some proofs and a careful complexity analysis. In particular, the obtained time complexity for the decision of HOM is a tower of three exponentials.

and inclusion?

3-EXP

Categories and Subject Descriptors: F.4.2 [**Mathematical Logic and Formal Languages**]: Grammars and Other Rewriting Systems; F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages

General Terms: Algorithms, Languages, Theory

Additional Key Words and Phrases: Tree automata, homomorphisms, regular languages, transducers

ACM Reference Format:
Godoy, G. and Giménez, O. 2013. The HOM problem is decidable. J. ACM 60, 4, Article 23 (August 2013), 44 pages.
DOI: http://dx.doi.org/10.1145/2501600

1. INTRODUCTION

1.1. Regular Tree Languages and the HOM Problem

In this article, we solve a long-standing open question in language theory, namely the HOM problem. With the aim of making the result accessible to a broad audience, we start by recalling some basic definitions on automata theory and tree languages. Later, we will describe the main results and its implications in other areas like term rewriting and XML types.

This is the journal version of Godoy et al. [2010], which appeared in STOC 2010.
The authors were supported by the Spanish Ministry of Education and Science by the FORMALISM project (TIN2007-66523).
Authors' addresses: Universitat Politènica de Catalunya, Edifici Omega, c. Jordi Girona 1-3 08034, Barcelona, Spain; email: ggodoy@lsi.upc.edu; omer.gimenez@gmail.com.

DOI: http://dx.doi.org/10.1145/2501600

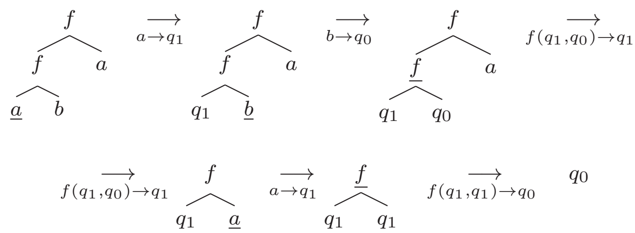

Fig. 1. Execution of the tree automaton A on the input tree $f(f(a, b), a)$.

A simple form of describing a finite automaton A is by giving a finite set of rules of the form $qa \to q'$, where q, q' are state symbols and a is an input symbol. Additionally, we must specify the initial state q_0 and the set of accepting states. A word w is accepted by A if $q_0 w$ can reach an accepting state by successive application of the rules. For example, the automaton $q_0 a \to q_1,\ q_0 b \to q_0,\ q_1 a \to q_0,\ q_1 b \to q_1$, where q_0 is the initial and the unique accepting state, recognizes the language W of words over the alphabet $\{a, b\}$ with an even number of occurrences of a. Word sets/languages recognized by automata are called regular.

Working with regular languages has several benefits. They are expressive enough in different settings, are closed under many natural operations, and there are efficient procedures for deciding most of their properties. In this article, we focus on homomorphisms. A word homomorphism is any function H relating words which commutes with word concatenation, that is, $H(xy) = H(x)H(y)$. Alternatively, any homomorphism H can be recursively defined by equations of the form $H(ax) = wH(x)$, where a is an alphabet symbol, x is a word variable and w is a word. In addition, $H(\lambda) = \lambda$ must hold for the empty word λ. For example, the image of the above language W by the homomorphism H defined by $H(ax) = aH(x),\ H(bx) = H(x),\ H(\lambda) = \lambda$ is the set of words over $\{a\}$ of even length. The advantage of homomorphic transformations on word languages is that they are regularity preserving, that is, the image of a regular language by a word homomorphism is also regular.

In some contexts, terms, that is, labeled trees, are more suited to express structured concepts than words. In a similar way to the definition of automata for words, one can define automata for trees. A (bottom-up) tree automaton (TA) is a set of rules of the form $f(q_1, \ldots, q_m) \to q$, where $q_1, \ldots, q_m, q$ are state symbols and f is an alphabet symbol that can be used for labeling some nodes of an input tree. Additionally, we must specify the set of accepting states. For example, consider the tree automaton A defined by the rules $a \to q_1,\ b \to q_0,\ f(q_0, q_0) \to q_0,\ f(q_0, q_1) \to q_1,\ f(q_1, q_0) \to q_1,\ f(q_1, q_1) \to q_0$, where q_0 is the only accepting state. It runs bottom-up on an input tree like $f(f(a, b), a)$ (see Figure 1) by successively applying the above rules. It accepts a tree if there is an execution reaching an accepting state. Our example accepts the tree language T of terms over binary f and nullaries a, b having an even number of a's.

Regular tree languages, that is, sets of terms recognized by tree automata, have almost all advantages of regular (word) languages. Nevertheless, they are not closed by homomorphisms. In the case of trees, homomorphisms can also be defined as functions commuting with a natural tree operator: application of substitution. Since this definition is not very intuitive, we prefer to just use the recursive alternative. A tree homomorphism can be defined by equations of the form $H(f(x_1, \ldots, x_n)) = t$, where t is a tree labeled by alphabet symbols and $H(x_1), \ldots, H(x_n)$, which may only appear at the leaves. For example, the image of the previous language T by the tree homomorphism H defined by $H(a) = a,\ H(b) = a,\ H(f(x_1, x_2)) = g(H(x_1), H(x_1))$ is the set of complete trees over binary g and nullary a (see Figure 2). It is not difficult to prove that this

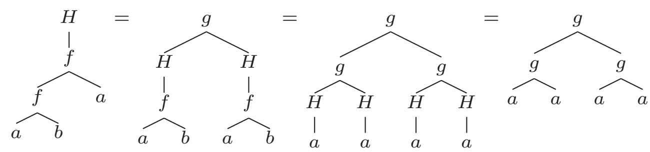

Fig. 2. Recursive application of the tree homomorphism H to compute $H(f(f(a,b),a))$.

language is not regular. In general, non-linear tree homomorphisms like H (due to the duplication of the variable x_1) may produce nonregular images. The HOM problem questions for a given regular tree language L described by a tree automaton and a tree homomorphism H, whether $H(L)$ is regular.

In order to increase expressiveness of the represented languages, several extensions of tree automata have been considered in literature. A well-studied case consists in allowing rules of the form $f(q_1,\ldots,q_m) \xrightarrow{c} q$, where c imposes an additional constraint for the rule application. Typically, c imposes equalities or disequalities between some subterms pending at certain positions relative to the rule application place. As an example of tree automaton with equality constraints consider the following: $a \to q,\ g(q,q) \xrightarrow{1=2} q$. The constraint $1 = 2$ imposes that the first and second subtrees must coincide for the application of the rule. Thus, this automaton recognizes the set of complete trees over binary g and nullary a. We denote by $\mathrm{TA}_{\neq,=}$ the class of tree automata with equality and disequality constraints, and by $\mathrm{TA}_{=}$ and $\mathrm{TA}_{\neq}$ the respective subclasses allowing only equalities or only disequalities.

1.2. Approach for Proving HOM

In order to solve HOM, our first contribution consists in a construction for converting a $\mathrm{TA}_{=}$ into a $\mathrm{TA}_{\neq}$ recognizing the complement language of the first one. This construction is rather easy: the $\mathrm{TA}_{\neq}$ just computes sets of states that are not reachable by the original $\mathrm{TA}_{=}$. For example, a state q might not be reachable because a rule having q as right-hand side is not applicable due to the fact that an equality constraint is not satisfied, and the new $\mathrm{TA}_{\neq}$ realizes that with a disequality constraint. To the best of our knowledge, this construction has not been stated before, although similar arguments are used by Comon and Jacquemard [1994] to construct a $\mathrm{TA}_{\neq}$ recognizing the set of normal forms of a term rewrite system.

The complement construction has significant consequences, like decidability of the universality problem for tree automata with equality constraints, and more generally, decidability of the inclusion of a regular language into the language represented by a $\mathrm{TA}_{=}$. Moreover, it gives a simple proof of undecidability of regularity test for $\mathrm{TA}_{\neq}$, and hence, for the so-called class of reduction tree automata (this question was left open in Bogaert et al. [1999]): this class allow for arbitrary disequalities and a limited amount of equalities (roughly speaking, a fixed number of equalities are permitted at each path of a run).

In a second step, we define a new class of automata with constraints called tree automata with disequality and HOM equality constraints, $\mathrm{TA}_{\neq,hom}$ for short. Essentially, they recognize the intersection language between a $\mathrm{TA}_{\neq}$ and the image of a regular language through a tree homomorphism. We also define the particular subclass of TA_{hom} that recognizes images of regular languages through tree homomorphisms.

The class $\mathrm{TA}_{\neq,hom}$ is interesting by itself. It is essentially a particular subclass of $\mathrm{TA}_{\neq,=}$, subsumes the class of tree automata with disequality and equality constraints between ~~brothers~~ siblings, and is independent from the class of reduction automata: on the one side, reduction automata can test equality of subterms at positions reaching different states while $\mathrm{TA}_{\neq,hom}$ cannot, but in contrast to reduction automata, $\mathrm{TA}_{\neq,hom}$ permit an unbounded number of equality tests at each path of a run. Moreover, emptiness and finiteness are decidable for $\mathrm{TA}_{\neq,hom}$. The proofs of decidability are based on the concept of pumping. A pumping alters a run producing an alternative run, smaller or bigger, depending on the necessity. For proving decidability of emptiness, we prove that a "too big" accepting run can be transformed into a smaller accepting run. This way we obtain an upper bound for the smallest accepting run. For proving decidability of finiteness, we prove that a "big" run can be transformed into an even bigger one. Thus, infinitely many accepting runs can be obtained from a "big" accepting run. This way we obtain an upper bound for the biggest run under the assumption that the recognized language is ~~not~~ ~~in~~finite.

These constructions and results allow to derive new significant consequences. Since two TA_{hom} A and B represent the images of regular languages through tree homomorphisms, by complementing A and intersecting with B we obtain a $\mathrm{TA}_{\neq,hom}$ whose emptiness is equivalent to the inclusion $\mathcal{L}(A) \supseteq \mathcal{L}(B)$. Therefore, we are able to prove decidability of the inclusion and equivalence problems for images of regular languages through tree homomorphisms.

Our decision algorithm for the HOM problem has a very simple description. First, it generates a TA_{hom} A recognizing the language $H(L)$, for the given regular tree language L and the given tree homomorphism H. Second, it linearizes A into a TA B by removing all equality constraints and replacing the involved positions in the constraints by all possible valid terms up to a certain height. Third, it checks $L = \mathcal{L}(B)$, concludes "regularity" in the affirmative case, and concludes "nonregularity" in the negative case. The proof that this algorithm decides HOM is based on the complement construction as well as on pumpings. In Section 7, we provide more intuition about the underlying ideas of this argumentation.

1.3. Summary of Results and Applications

We summarize our contributions as follows.

(1) Given a $\mathrm{TA}_{=}$ A, a $\mathrm{TA}_{\neq}$ $\bar{A}$ can be constructed recognizing the complementary language of the first one. The worst-case size of $\bar{A}$ is exponential with respect to the size of A, and is constructed in exponential time.
(2) The converse construction of a $\mathrm{TA}_{=}$ from a $\mathrm{TA}_{\neq}$ can also be done analogously.
(3) As a consequence of item (1), the inclusion problem $\mathcal{L}(B) \subseteq \mathcal{L}(A)$ for a TA B and a $\mathrm{TA}_{=}$ A is shown decidable. The obtained time complexity is exponential.
(4) As a consequence of item (3) the universality problem for $\mathrm{TA}_{=}$, that is, whether a given $\mathrm{TA}_{=}$ A accepts all terms, is decidable. The obtained time complexity is exponential.
(5) As a consequence of item (3), the inclusion problem $L_1 \subseteq H(L_2)$, for a given tree homomorphism H and regular tree languages L_1, L_2, is decidable. The obtained time complexity is exponential.
(6) As a consequence of item (1), and the fact that regularity and emptiness are known undecidable for $\mathrm{TA}_{=}$, regularity and universality of the language represented by a given $\mathrm{TA}_{\neq}$ are undecidable.
(7) The emptiness of the language recognized by a given $\mathrm{TA}_{\neq,hom}$ is shown decidable. The obtained time complexity is a tower of three exponentials.
(8) As a consequence of items (1) and (7), the inclusion problem $H_1(L_1) \subseteq H_2(L_2)$, for given tree homomorphisms H_1, H_2 and regular tree languages L_1, L_2 is decidable. The obtained time complexity is a tower of three exponentials.

(9) As a consequence of item (8), the equivalence problem $H_1(L_1) = H_2(L_2)$, for given tree homomorphisms H_1, H_2 and regular tree languages L_1, L_2, is decidable. The obtained time complexity is a tower of three exponentials.
(10) The finiteness of the language recognized by a given $\mathrm{TA}_{\neq,hom}$ is decidable. The obtained time complexity is a tower of three exponentials.
(11) As a consequence of items (1) and (10), finiteness of the set $H_1(L_1) - H_2(L_2)$, for given tree homomorphisms H_1, H_2 and regular tree languages L_1, L_2, is decidable. The obtained time complexity is a tower of three exponentials.
(12) The HOM problem is decidable. The obtained time complexity is a tower of three exponentials.

Our results have also implications in the context of term rewriting. The set of normal forms by a term rewrite system, that is, the set of terms for which no rule can be applied, can be described as the complement of the image of a tree homomorphism. Thus, we can compare by inclusion and equality such sets of reducible terms by two different term rewrite systems. In Gilleron and Tison [1995], the question $Rel(L_1) \subseteq L_2$ is shown decidable for given regular tree languages L_1, L_2 and where the relation Rel is defined in several ways according to a given term rewrite system R. For instance, decidability holds when $Rel(t)$ is the set of reachable terms from term t by applying just one rewrite step. By using our results, we are able to extend some of the results in Gilleron and Tison [1995] to the question $Rel_1(L_1) \subseteq Rel_2(L_2)$.

The decidability of HOM has also direct implications in the context of XML. Extended DTD (document type descriptions) is a formalism expressively equivalent to tree automata. Thus, in this setting, a tree automaton is a type, that is, it defines a set of valid descriptions, and the HOM problem is equivalent to the following problem: given a type τ and a homomorphic transformation T, is $T(\tau)$ a type? We conclude that this question is decidable, and moreover, we are able to generate this type since our decision procedure of HOM is constructive. The typechecking problem [Schwentick 2007], that is, "does a transformation T converts every input document of a type τ_1 into a document of a type τ_2?", is well known to be decidable by backward typechecking for the case where the transformation T is a tree homomorphism: one can test the inclusion $\tau_1 \subseteq T^{-1}(\tau_2)$. Our result of item (8) gives an alternative decision method for this problem by forward typechecking, since we are able to directly test $T(\tau_1) \subseteq \tau_2$: it suffices to define L_1 as τ_1, L_2 as τ_2, H_1 as T and H_2 as the identity transformation, which is homomorphic.

The decidability result of item (8) has more implications: images of tree homomorphisms could be used to describe types instead of just regular tree languages. This would give much finer approximations of real types, for which we would have closure under union and application of tree homomorphism, but not under intersection. Item (8) shows that type inclusion is decidable in this setting. Suciu [2002, Example 3.1] shows that type inference using regular tree languages is not complete by means of a transformation that copies a tree. We do not solve Suciu's example, because such a transformation not only copies, but also transforms in different ways the two resulting copies. Nevertheless, the approach using tree homomorphisms is promising in this setting, and perhaps other variants of tree automata allowing different semantics for the equality constraints may solve such example.

1.4. Related Work

The study of preservation of regularity by tree homomorphisms is introduced in Thatcher [1969]. In this article, tree automata and tree homomorphisms are defined for the first time, and it is proved that the application of linear tree homomorphisms preserves regularity. Tree homomorphisms are also introduced in Engelfriet [1975], as a particular case of tree stateless transducers. The oldest reference we know where the decidability of HOM is explicitly stated as a difficult open question is Gilleron [1991]. In the same paper, the particular case of regularity of the reducible terms by a term rewrite system is also left as an open problem. The latter case is posted as the 7th problem in the RTA list of open problems in 1991 by Comon and Dauchet [RTA-LOOP 1991]. It is solved independently in Hofbauer and Huber [1992], Vágvölgyi and Gilleron [1992], and Kucherov and Tajine [1995]. The HOM problem appears also in Fülöp [1994], where the more general case of regularity of the range of a top-down tree transducer is shown undecidable. In more recent years, some other particularizations of the HOM problem have been taken into consideration. Tree automata with constraints are used to prove decidability of the HOM problem restricted to shallow homomorphisms (the ones where the height of any term bounds the height of its image) in Bogaert et al. [1999]. For the same particular case, it is shown in Dauchet et al. [2002] that tree homomorphisms preserving regularity can be assumed to be linear. A semidecision procedure for detecting some cases of regularity of images of regular languages through tree homomorphisms is proposed in Vágvölgyi [2009]. The HOM problem restricted to monadic signatures or to top-copying homomorphisms is proved decidable in Godoy et al. [2008]. This particular case is proved to be exptime-complete in Giménez et al. [2011].

Emptiness of tree automata with disequality constraints is proved decidable in Comon and Jacquemard [1994]. Exptime-completeness of this problem is proved in Comon and Jacquemard [1997]. This result is developed in order to prove exptime-completeness of the ground reducibility problem. The decidability result in Comon and Jacquemard [1994] is extended in Dauchet et al. [1995] to the class of reduction automata. This result is used to prove decidability of the existential fragment of the first order theory of reduction. Emptiness and finiteness of tree automata with disequality and equality constraints between brothers (direct subtrees) is proved decidable in Bogaert and Tison [1992]. Regularity for this class is proved decidable in Bogaert et al. [1999]. As a consequence, the HOM problem is proved decidable for the particular case of shallow homomorphisms, as we have mentioned and described previously. But also, these results are a key for proving decidability of preservation of regularity for shallow term rewrite systems and innermost rewriting in Godoy and Huntingford [2007], Gascón et al. [2009]. Several other variants of tree automata with constraints have been developed, providing decidability results in logic and term rewriting, with application in protocol verification.

1.5. Organization of the Article

In Section 2, we introduce basic concepts on terms and tree automata. In Section 3, we present the construction transforming a $TA_{=}$ into a $TA_{\neq}$. In Section 4, we define $TA_{\neq,hom}$ and TA_{hom} and their runs, show that the image of a regular language through a tree homomorphism can be recognized by a TA_{hom}, define the intersection of a TA_{hom} and a $TA_{\neq}$, the intersection of two runs, and the respective projections to recover the original runs from a run of the intersection. In Section 5, we define the concept of pumping of a run of a $TA_{\neq,hom}$. Also we show that for a big enough run there exists a un- pumping providing a smaller run, thus concluding decidability of the emptiness problem for $TA_{\neq,hom}$. Moreover, we show that for a big enough run there exist pumpings providing infinitely many bigger runs, thus concluding decidability of the finiteness problem for $TA_{\neq,hom}$. In Section 6, we show all the significant intermediate consequences of our constructions. In Section 7, we use all the developed techniques to prove decidability of the HOM problem.

2. PRELIMINARIES

2.1. Terms

The size of a set S is denoted by $|S|$, and the powerset of S is denoted by 2^S. We assume that the reader is familiarized with terms, positions, substitutions, and replacements. For more detailed explanations see Baader and Nipkow [1998].

A *signature* consists of an alphabet Σ, that is, a finite set of symbols, together with a mapping that assigns to each symbol in Σ a natural number, its *arity*. We write $\Sigma^{(m)}$ to denote the subset of symbols in Σ that are of arity m, and we write $f^{(m)}$ to denote that f is a symbol of arity m. The *set of all terms over* Σ is denoted $\mathcal{T}(\Sigma)$ and is inductively defined as the smallest set T such that for every $f \in \Sigma^{(m)}$, $m \geq 0$, and $t_1, \dots, t_m \in T$, the term $f(t_1, \dots, t_m)$ is in T. For a term of the form $a()$, we simply write a. For instance, if $\Sigma = \{f^{(2)}, a^{(0)}\}$, then $\mathcal{T}(\Sigma)$, is the set of all terms that represent binary trees with internal nodes labeled f and leaves labeled a. We fix the set $\mathcal{X} = \{x_1, x_2, \dots\}$ of variables, that is, any set V of variables is always assumed to be a subset of $\mathcal{X}$. The set of terms over Σ with variables in $\mathcal{X}$, denoted $T(\Sigma \cup \mathcal{X})$, is the set of terms over $\Sigma \cup \mathcal{X}$ where every symbol in $\mathcal{X}$ has arity zero. By $|t|$, we denote the size of t, defined recursively as $|f(t_1, \dots, t_m)| = 1 + |t_1| + \cdots + |t_m|$ for each $f \in \Sigma^{(m)}$, $m \geq 0$ and $t_1, \dots, t_m \in \mathcal{T}(\Sigma \cup \mathcal{X})$, and $|x| = 1$ for each x in $\mathcal{X}$. By $height(t)$ we denote the height of t, defined recursively as $height(f(t_1, \dots, t_m)) = 1 + max(height(t_1), \dots, height(t_m))$ for each $f \in \Sigma^{(m)}$, $m \geq 1$ and $t_1, \dots, t_m \in \mathcal{T}(\Sigma)$, $height(a) = 0$ for each $a \in \Sigma^{(0)}$, and $height(x) = 0$ for each $x \in \mathcal{X}$. Positions in terms are sequences of natural numbers. Given a term $f(t_1, \dots, t_m) \in \mathcal{T}(\Sigma \cup \mathcal{X})$, its set of positions $Pos(t)$ is defined recursively as $\{\lambda\} \cup_{1 \leq i \leq m} \{i.p \mid p \in Pos(t_i)\}$. Here, λ denotes the empty sequence (position of the root node), and . denotes concatenation. The subterm of t at position p is denoted by $t|_p$, and is formally defined as $t|_\lambda = t$, and $f(t_1, \dots, t_m)|_{i.p} = t_i|_p$. $root(f(t_1, \dots, t_m))$ is f for any symbol f. Thus, the symbol of t occurring at position p is denoted by $root(t|_p)$, and we say that t at position p is labeled by $root(t|_p)$. For instance, for $s = g(f(a, b), c)$, $s|_1$ equals $f(a, b)$ and $root(s|_{1.2})$ is b. For a set Γ, we use $Pos_\Gamma(t)$ to denote the set of positions of t that are labeled by symbols in Γ. When a position p is of the form $p_1.p_2$, we say that p_1 is a prefix of p, denoted $p_1 \leq p$, and p_2 is a suffix of p. If, in addition, p_2 is not λ, then we say that p_1 is a proper prefix of p, denoted $p_1 < p$. Moreover, with $p - p_1$, we denote p_2. We say that two positions p_1 and p_2 are parallel, denoted $p_1 \parallel p_2$, if neither $p_1 \leq p_2$ nor $p_2 \leq p_1$ hold. For terms s, t and $p \in Pos(s)$, we denote by $s[t]_p$ the result of replacing the subterm at position p in s by the term t. More formally, $s[t]_\lambda$ is t, and $f(s_1, \dots, s_m)[t]_{i.p}$ is $f(s_1, \dots, s_{i-1}, s_i[t]_p, s_{i+1}, \dots, s_m)$. For instance, $f(f(a, a), a)[a]_1$ is $f(a, a)$. A *substitution* σ is a mapping from variables to terms. It can be homomorphically extended to a function from terms to terms: $\sigma(t)$ denotes the result of simultaneously replacing in t every $x \in Dom(\sigma)$ by $\sigma(x)$. For example, if σ is $\{x \mapsto f(b, y), y \mapsto a\}$, then $\sigma(g(x, y))$ is $g(f(b, y), a)$. A rewrite rule is a pair of terms $l \to r$. Application of a rewrite rule $l \to r$ to a term $s[\sigma(l)]_p$ at position p produces the term $s[\sigma(r)]_p$. If R is a set of rules, application of a rule of R to a term s resulting into a term t is denoted by $s \to_R t$, and the reflexive-transitive closure of this relation is denoted by $\to_R^*$.

In this article, unless the opposite is stated, by $t|_{p_1} = t|_{p_2}$, we mean that p_1 and p_2 are positions in $Pos(t)$ and the subterms of t at positions p_1 and p_2 coincide. On the other hand, by $t|_{p_1} \neq t|_{p_2}$, we mean that either p_1 or p_2 is not in $Pos(t)$, or that the subterms $t|_{p_1}$ and $t|_{p_2}$ are different. Note that, with this semantics, $t|_{p_1} \neq t|_{p_2}$ is the negation of $t|_{p_1} = t|_{p_2}$.

2.2. Tree Automata with Constraints

Tree automata and regular languages are well-known concepts of theoretical computer science [Gécseg and Steinby 1984, 1997; Comon et al. 2007]. We assume that the reader knows the Boolean closure properties and the decidability results on regular tree languages. Here, we only recall the notion of tree automata with constraints.

Definition 2.1. A *tree automaton with disequality and equality constraints*, $\mathrm{TA}_{\neq,=}$ for short, is a tuple $A = \langle Q, \Sigma, F, \Delta\rangle$, where Q is a set of states, Σ is a signature, $F \subseteq Q$ is the subset of final states, and Δ is a set of rules of the form $f(q_1, \ldots, q_m) \xrightarrow{c} q$, where $q_1, \ldots, q_m, q$ are in Q, f is in $\Sigma^{(m)}$ and c, called the constraint of the rule, is a conjunction/set of atoms of the form $p_1 \neq p_2$ and $p_1 = p_2$ for arbitrary positions p_1, p_2. When all constraints in Δ contain only disequalities (respectively, equalities) we say that A is a $\mathrm{TA}_{\neq}$ (respectively, a $\mathrm{TA}_{=}$). When all the constraints are empty, we say that A is a TA.

In literature, a run r of a tree automaton A on a term t is usually defined as a function from the positions of t to rules of A. Intuitivelly, the rules are applied bottom-up on t, and $r(p)$ is the rule applied at position p. The state reached at p is the right-hand side of $r(p)$. For r being correct, if $r(p)$ is a rule of the form $f(q_1, \ldots, q_m) \xrightarrow{c} q$, then $root(t|_p)$ is f and the states reached at positions $p.1, \ldots, p.m$ are $q_1, \ldots, q_m$. Moreover, for each equality constraint $p_1 = p_2$ in c, the subterms $t|_{p.p_1}$ and $t|_{p.p_2}$ must be identical, and for each disequality constraint $p_1 \neq p_2$ in c, the subterms $t|_{p.p_1}$ and $t|_{p.p_2}$ must be different.

In this article, we use a different, but equivalent, notation for runs. A run r is a term on the alphabet of rules. The symbol labeling position p is the rule applied at p. Note that r implicitly defines the term t on which r applies: if the symbol/rule labeling p is $f(q_1, \ldots, q_m) \xrightarrow{c} q$, then the symbol labeling t at position p is f. Again, for r being correct, if the symbol/rule labeling position p is of the form $f(q_1, \ldots, q_m) \xrightarrow{c} q$, then, the right-hand sides of the symbols/rules labeling positions $p.1, \ldots, p.m$ are $q_1, \ldots, q_m$, respectively. Moreover, for each equality constraint $p_1 = p_2$ in c, the subterms $t|_{p.p_1}$ and $t|_{p.p_2}$ must be identical, and for each disequality constraint $p_1 \neq p_2$ in c, the subterms $t|_{p.p_1}$ and $t|_{p.p_2}$ must be different. The reason for using this alternative representation is because, very often in this article, we have to make replacements on runs producing terms over the alphabet of rules, which are not necessarily runs, and reason about them. Moreover, in further sections, new tree automata with constraints are defined, and in such definitions, an equality constraint $p_1 = p_2$ requires not only that the corresponding subterms at the relative positions p_1 and p_2 coincide, but also that the corresponding runs coincide. Altogether, it is easier to manage using the formalism of terms for runs.

Definition 2.2. Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $\mathrm{TA}_{\neq,=}$. The *alphabet of a run* of A is Δ, where each symbol of the form $(f(q_1, \ldots, q_m) \xrightarrow{c} q)$ has arity m, that is, the same as f. The *resulting state* of a term r in $\mathcal{T}(\Delta)$ is q if r is of the form $(f(q_1, \ldots, q_m) \xrightarrow{c} q)$ $(t_1, \ldots, t_m)$. The *projection* $\pi_\Sigma : \mathcal{T}(\Delta) \to \mathcal{T}(\Sigma)$ is recursively defined as $(\pi_\Sigma(f(q_1, \ldots, q_m) \xrightarrow{c} q)(t_1, \ldots, t_m)) = f(\pi_\Sigma(t_1), \ldots, \pi_\Sigma(t_m))$.

Definition 2.3. Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $\mathrm{TA}_{\neq,=}$. We define the concept of *run* of A as a term in $\mathcal{T}(\Delta)$ satisfying certain conditions recursively as follows. Let $(f(q_1, \ldots, q_m) \xrightarrow{c} q)$ be a rule of Δ. Let $r_1, \ldots, r_m$ be runs of A with resulting states $q_1, \ldots, q_m$, respectively. Let t be $f(\pi_\Sigma(r_1), \ldots, \pi_\Sigma(r_m))$. Suppose $t|_{p_1} \neq t|_{p_2}$ holds for each $(p_1 \neq p_2) \in c$, and that $t|_{p_1} = t|_{p_2}$ holds for each $(p_1 = p_2) \in c$. Then, $(f(q_1, \ldots, q_m) \xrightarrow{c} q)(r_1, \ldots, r_m)$ is a run of A on the term t.

By $\mathcal{L}(A, q)$, we denote the set of terms t for which there exists a run r of A with resulting state q such that $\pi_\Sigma(r) = t$. The language accepted by A, denoted $\mathcal{L}(A)$, is $\bigcup_{q\in F} \mathcal{L}(A, q)$. A language L is *regular* if there exists a TA A such that $\mathcal{L}(A) = L$ holds.

A run r on t is *deterministic* if, for each two positions p_1, p_2 satisfying $t|_{p_1} = t|_{p_2}$, $r|_{p_1} = r|_{p_2}$ also holds, that is, deterministic runs have identical subruns for identical subterms. We say that a $\mathrm{TA}_{\neq,=}$ A *admits deterministic accepting runs* if for each $t \in \mathcal{L}(A)$, there is a deterministic accepting run of A on t.

For the complexity analysis of the presented algorithms in this article, we define some notions related to the size of the input.

Definition 2.4. The size of a position, denoted $|p|$, is its length. The size of an atom $p_1 \neq p_2$ or $p_1 = p_2$ is $|p_1| + |p_2|$. The size of a constraint c is the sum of sizes of its atoms. The size of a rule $f(q_1, \ldots, q_m) \xrightarrow{c} q$, denoted $|f(q_1, \ldots, q_m) \xrightarrow{c} q|$, is $|f(q_1, \ldots, q_m)| + |c| + 1$. The size of the set of rules Δ, denoted $\|\Delta\|$ for avoiding ambiguity with the number of rules $|\Delta|$, is $\sum_{(f(q_1,\ldots,q_m)\xrightarrow{c}q)\in\Delta} |f(q_1, \ldots, q_m) \xrightarrow{c} q|$. Given a $\mathrm{TA}_{\neq,=}$ $A = \langle Q, \Sigma, F, \Delta\rangle$, its size, denoted $|A|$, is $|Q| + \|\Delta\|$.

By $N(A)$, we denote the number of distinct equality and disequality atoms in the rules of A. By $h(A)$, we denote the maximum between 1 and the lengths of the positions ocurring in equality and disequality atoms. We just write N, h when A is clear from the context. Note that these values are bounded by the size of A.

Definition 2.5. Let Σ_1, Σ_2 be two signatures. A *tree homomorphism* is a function $H : \mathcal{T}(\Sigma_1) \to \mathcal{T}(\Sigma_2)$, which can be defined as follows.

Let X_m represent the set of variables $\{x_1, \ldots, x_m\}$ for each natural number m, which is assumed to be disjoint with Σ_1 and Σ_2. The definition of a tree homomorphism $H : \mathcal{T}(\Sigma_1) \to \mathcal{T}(\Sigma_2)$ requires to define $H(f(x_1, \ldots, x_m))$ for each function symbol $f \in \Sigma_1$ of arity m as a term t_f in $\mathcal{T}(\Sigma_2 \cup X_m)$. After that, $H(f(t_1, \ldots, t_m))$ is defined, for each term $f(t_1, \ldots, t_m) \in \mathcal{T}(\Sigma_1)$, as $\{x_1 \mapsto H(t_1), \ldots, x_m \mapsto H(t_m)\}(t_f)$.

Alternatively, tree homomorphisms can be defined in the following way as a function $H : \mathcal{T}(\Sigma_1) \to \mathcal{T}(\Sigma_2)$ satisfying the following condition. For any arbitrary set of variables $\mathcal{X}$ disjoint with Σ_1 and Σ_2, there exists an extension $\overline{H} : \mathcal{T}(\Sigma_1 \cup \mathcal{X}) \to \mathcal{T}(\Sigma_2 \cup \mathcal{X})$ of H such that, $\overline{H}(x) = x$ for each x in $\mathcal{X}$, and for each term $t \in \mathcal{T}(\Sigma_1 \cup \mathcal{X})$ and each substitution $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma_1 \cup \mathcal{X})$, $\overline{H}(\sigma(t))$ is $(\overline{H}(\sigma))(\overline{H}(t))$, where $(\overline{H}(\sigma))(x)$ is interpreted in the natural way as $\overline{H}(\sigma(x))$.

Definition 2.6. The HOM problem is defined as follows:
Input: A TA A and a tree homomorphism H.
Question: Is $H(\mathcal{L}(A))$ regular?

For the complexity analysis of the presented algorithms in this article, we define the following notions related to the size of the input.

Definition 2.7. The size of a tree homomorphism $H : \mathcal{T}(\Sigma_1) \to \mathcal{T}(\Sigma_2)$, denoted $|H|$, is the sum of sizes of the images of terms of the definition of H, that is, $\sum_{f\in\Sigma_1^{(m)},m\geq 0} |H(f(x_1, \ldots, x_m))|$. We denote by $N_{Pos}(H)$ the number of distinct positions in terms of such images, that is, $|\{p \mid \exists m \geq 0,\ f \in \Sigma_1^{(m)} : (p \in Pos(H(f(x_1, \ldots, x_m))))\}|$. We denote by $N(H)$ the number of distinct pairs of positions in which, intuitively, the definition of H forces an equality on the image by means of a duplicated variable. More precisely, $N(H)$ is $|\{(p_1, p_2) \mid p_1 \neq p_2 \wedge \exists m \geq 0,\ f \in \Sigma_1^{(m)} : (H(f(x_1, \ldots, x_m))|_{p_1} = H(f(x_1, \ldots, x_m))|_{p_2} \in \mathcal{X})\}|$. We denote by $h(H)$ the maximum height of a term in the definition of H, that is, $max(\{|p| \mid \exists m \geq 0,\ f \in \Sigma_1^{(m)} : (p \in Pos(H(f(x_1, \ldots, x_m))))\})$

3. THE COMPLEMENT OF A TA$_=$

For a given TA$_=$ $A = \langle Q, \Sigma, F, \Delta\rangle$, we want to construct a TA$_{\neq}$ B recognizing the complement of $\mathcal{L}(A)$. To this end, we consider 2^Q as the set of states of B, that is, the states of B are sets of states of A. The rules of B are defined so that there exists a run with B on a term t with resulting state $S \subseteq Q$ if and only if, for each q in S, there is no run with A on t with resulting state q. In other words, a set of states S reachable using B satisfies that no q in S is reachable using A. Moreover, B is able to compute any S satisfying this condition and, in particular, a maximal one. Thus, if one of the possible S's contains F, we can assure that t is not in $\mathcal{L}(A)$. For this reason, the set of accepting states of B is defined as $\{F' \mid F \subseteq F' \subseteq Q\}$. The rules $f(S_1, \ldots, S_m) \xrightarrow{D} S$ of B are constructed to ensure that, for those states q in S, no rule of A with right-hand side q can be applied.

Definition 3.1. Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a TA$_=$. Then, the *complement* TA$_{\neq}$ B of A is defined as the tuple $\langle 2^Q, \Sigma, \{F' \mid F \subseteq F' \subseteq Q\}, \Delta'\rangle$ where Δ' is the set of rules $f(S_1, \ldots, S_m) \xrightarrow{D} S$ satisfying the following.

—$S_1, \ldots, S_m, S \subseteq Q$.
—D is a conjunction of disequalities $p \neq q$ such that $p = q$ occurs in the constraint of some rule in Δ.
—For each q in S and each rule of the form $f(q_1, \ldots, q_m) \xrightarrow{c} q$ in Δ either there exists some i in $\{1, \ldots, m\}$ satisfying $q_i \in S_i$, or there exist positions p_1, p_2 such that $p_1 = p_2$ occurs in c and $p_1 \neq p_2$ occurs in D.

The last item of the previous definition is the one assuring that none of the states q in S is reachable using A. A rule $f(q_1, \ldots, q_m) \xrightarrow{c} q$ cannot be applied to a position of a term rooted with f if either for some relative position i the corresponding q_i is not reached, or some equality $p_1 = p_2$ in c does not hold. The first condition is inductively ensured by having q_i inside the corresponding S_i. The second condition is ensured by having a constraint $p_1 \neq p_2$ in D.

Example 3.2. Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be the TA$_=$ with $Q = \{q\}$, $\Sigma = \{f^{(2)}, a^{(0)}\}$, $F = \{q\}$, and where Δ contains the two rules $a \to q$ and $f(q, q) \xrightarrow{1=2} q$. This TA$_=$ recognizes the language of the complete trees over Σ. We can construct the complement TA$_{\neq}$ B as follows. The set of states is $2^Q = \{\emptyset, \{q\}\}$, the signature is the same Σ, the set of accepting states is $F' = \{\{q\}\}$ and some of the rules in Δ' are:

—$a \to \emptyset$
—$a \xrightarrow{1\neq 2} \emptyset$
—$f(\emptyset, \emptyset) \to \emptyset$
—$f(\emptyset, \emptyset) \xrightarrow{1\neq 2} \emptyset$
—$f(\emptyset, \emptyset) \xrightarrow{1\neq 2} \{q\}$
—$f(\{q\}, \emptyset) \to \{q\}$
—$f(\{q\}, \emptyset) \xrightarrow{1\neq 2} \{q\}$
—$f(\emptyset, \{q\}) \to \{q\}$
—$f(\emptyset, \{q\}) \xrightarrow{1\neq 2} \{q\}$
—$f(\{q\}, \{q\}) \to \{q\}$
—$f(\{q\}, \{q\}) \xrightarrow{1\neq 2} \{q\}$.

Note that several of the rules in B having the constraint $1 \neq 2$ are subsumed by other analogous rules without the constraint. This is not the case for the rule $f(\emptyset, \emptyset) \xrightarrow{1\neq 2} \{q\}$, since there is not an analogous rule $f(\emptyset, \emptyset) \to \{q\}$ in B. This is because for the original rule $f(q, q) \xrightarrow{1=2} q$ of A, the state q is not in $\emptyset$, and the constraint $1 = 2$ is not contradicted by the constraint of $f(\emptyset, \emptyset) \to \{q\}$, since it is empty.

This automaton clearly recognizes the language of the noncomplete trees over Σ (because at each position we can nondeterministically check that either at least one child is noncomplete (state $\{q\}$), or both children are not equal (constraint $1 \neq 2$)).

The following lemma establishes one of the directions of the statement mentioned previously: whenever a state S is the result of a run of B on a term t, no state q in S can be the result of a run of A on t.

LEMMA 3.3. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a TA$_=$, and $B = \langle 2^Q, \Sigma, \{F' \mid F \subseteq F' \subseteq Q\}, \Delta'\rangle$ be its complement TA$_{\neq}$. Let q be a state in Q and let S be a state of B containing q. Let t be a term in $\mathcal{L}(B, S)$.*

Then, t is not in $\mathcal{L}(A, q)$.

PROOF. We prove it by contradiction. Let t be a minimum term in size contradicting the statement, that is, there exists $S \subseteq Q$, a run r' of B satisfying $\pi_\Sigma(r') = t$ with resulting state S, a state q in S and a run r of A satisfying $\pi_\Sigma(r) = t$ with resulting state q, and no other term smaller than t accomplishes this statement.

We write t more explicitly as $f(t_1, \ldots, t_m)$, and these runs r' and r as $(f(S_1, \ldots, S_m) \xrightarrow{D} S)(r'_1, \ldots, r'_m)$ and $(f(q_1, \ldots, q_m) \xrightarrow{c} q)(r_1, \ldots, r_m)$, respectively. Note that $S_1, \ldots, S_m$ are the resulting states of the runs $r'_1, \ldots, r'_m$ of B on $t_1, \ldots, t_m$, respectively, and $q_1, \ldots, q_m$ are the resulting states of the runs $r_1, \ldots, r_m$ of A on $t_1, \ldots, t_m$, respectively.

By the definition of Δ', since q belongs to S, for the rule $f(q_1, \ldots, q_m) \xrightarrow{c} q$ it holds that either (i) there exists some i in $\{1, \ldots, m\}$ satisfying $q_i \in S_i$, or (ii) there exist positions p_1, p_2 such that $p_1 = p_2$ occurs in c and $p_1 \neq p_2$ occurs in D.

In case (i), t_i, S_i, r'_i, q_i, and r_i satisfy the assumed conditions for t, S, r', q and r, but also $|t_i| < |t|$ holds. This is in contradiction with the minimality of t.

In case (ii), by the definition of run applied on r, it holds $t|_{p_1} = t|_{p_2}$. But, similarly, by the definition of run applied on r', it holds $t|_{p_1} \neq t|_{p_2}$, which is a contradiction again.

The following lemma establishes the other direction of the previous statement, but for maximal S's, that is, given a term t, there exists a run r with B on t whose resulting state S is just the set of states q which cannot be the result of a run with A on t. Moreover, this is also the case for each subrun (subterm) of r.

LEMMA 3.4. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a TA$_=$, and $B = \langle 2^Q, \Sigma, \{F' \mid F \subseteq F' \subseteq Q\}, \Delta'\rangle$ be its complement TA$_{\neq}$. Let t be a term.*

Then, there exists a run r of B satisfying $\pi_\Sigma(r) = t$ and such that, for each $p \in Pos(r)$ it holds that $r|_p$ is a run with resulting state $\{q \in Q \mid t|_p \notin \mathcal{L}(A, q)\}$.

PROOF. We prove it by induction on $|t|$. We write t more explicitly as $f(t_1, \ldots, t_m)$. By induction hypothesis, for each t_i, there exists a run r_i of B satisfying $\pi_\Sigma(r_i) = t_i$ and such that, for each $p \in Pos(r_i)$ it holds that $r_i|_p$ is a run with resulting state $\{q \in Q \mid t_i|_p \notin \mathcal{L}(A, q)\}$. In particular, the resulting state S_i of r_i is $\{q \in Q \mid t_i \notin \mathcal{L}(A, q)\}$.

Let D be the constraint defined as the conjunction of disequalities $p_1 \neq p_2$ such that $p_1 = p_2$ occurs in the constraint of some rule of A and $t|_{p_1} \neq t|_{p_2}$ holds (recall that, by $t|_{p_1} \neq t|_{p_2}$, we understand that either p_1 or p_2 is not in $Pos(t)$, or that the subterms $t|_{p_1}$ and $t|_{p_2}$ are different). In order to conclude, it suffices to prove that $f(S_1, \ldots, S_m) \xrightarrow{D} S$

for $S = \{q \in Q \mid t \notin \mathcal{L}(A, q)\}$ is a rule of B. To this end, we must show, for each q in S and each rule of the form $f(q_1, \ldots, q_m) \xrightarrow{c} q$ in Δ, that either some i in $\{1, \ldots, m\}$ satisfies $q_i \in S_i$, or there exist positions p_1, p_2 such that $p_1 = p_2$ occurs in c and $p_1 \neq p_2$ occurs in D. Thus, consider any of such q and $f(q_1, \ldots, q_m) \xrightarrow{c} q$ and suppose that all i in $\{1, \ldots, m\}$ satisfy $q_i \notin S_i$. Then, by the definition of each of such S_i, there exists a run r'_i of A satisfying $\pi_\Sigma(r'_i) = t_i$ and with resulting state q_i. Since q is in S, by the definition of S it holds that there is no run r' of A with resulting state q such that $\pi_\Sigma(r') = t$. Hence, $(f(q_1, \ldots, q_m) \xrightarrow{c} q)(r'_1, \ldots, r'_m)$ is not a run. Thus, some $p_1 = p_2$ occurring in the constraint c must be unsatisfied on t, that is, $t|_{p_1} \neq t|_{p_2}$ for some $p_1 = p_2$ in c. By the selection of D, $p_1 \neq p_2$ occurs in D. Thus, the existence of such p_1, p_2 concludes the proof.

$B = \bar{A}$

COROLLARY 3.5. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_=$, and $B = \langle 2^Q, \Sigma, \{F' \mid F \subseteq F' \subseteq Q\}, \Delta'\rangle$ be its complement $TA_{\neq}$. Then, B admits deterministic accepting runs.*

PROOF. We must prove that, for each term t in $\mathcal{L}(B)$, there is a deterministic accepting run of B on t. To this end, consider a term t in $\mathcal{L}(B)$. Thus, t is in $\mathcal{L}(B, F')$, for some F' holding $F \subseteq F' \subseteq Q$. By Lemma 3.3, each state q in F' satisfies $t \notin \mathcal{L}(A, q)$. By Lemma 3.4, there exists a run r' of B on t such that, for each $p \in Pos(r')$ it holds that $r'|_p$ is a run with resulting state $\{q \in Q \mid t|_p \notin \mathcal{L}(A, q)\}$. Note that r' is a deterministic run. Note also that $F' \subseteq \{q \in Q \mid t \notin \mathcal{L}(A, q)\}$. Thus, $F \subseteq \{q \in Q \mid t \notin \mathcal{L}(A, q)\} \subseteq Q$ holds, and hence, r' is a deterministic accepting run of B on t.

THEOREM 3.6. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_=$, and $B = \langle 2^Q, \Sigma, \{F' \mid F \subseteq F' \subseteq Q\}, \Delta'\rangle$ be its complement $TA_{\neq}$. Then, $\overline{\mathcal{L}(A)} = \mathcal{L}(B)$.*

PROOF.

—We first prove $\overline{\mathcal{L}(A)} \supseteq \mathcal{L}(B)$. Let t be a term in $\mathcal{L}(B)$. Then, there exists a run r' of B with resulting state S such that $\pi_\Sigma(r') = t$ and $F \subseteq S \subseteq Q$ hold. Hence, for each q in F, by Lemma 3.3, it follows that t is not in $\mathcal{L}(A, q)$. Thus, t is not in $\mathcal{L}(A)$.

—Now, we prove $\overline{\mathcal{L}(A)} \subseteq \mathcal{L}(B)$. Let t be a term that is not accepted by A. Then, for each state q of F, $t \notin \mathcal{L}(A, q)$ holds. By Lemma 3.4, t is in $\mathcal{L}(B, S)$ where S is $\{q \in Q \mid t \notin \mathcal{L}(A, q)\}$, and since $F \subseteq S \subseteq Q$ holds, it follows that t is in $\mathcal{L}(B)$.

Proceeding analogously we could transform a $TA_{\neq}$ into a $TA_=$ recognizing the complement of the first. We do not develop this transformation since it is very similar and we do not use it for the proof of the HOM problem, but state the analogous consequence as follows.

THEOREM 3.7. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq}$. Then, the $TA_=$ $B = \langle 2^Q, \Sigma, \{F' \mid F \subseteq F' \subseteq Q\}, \Delta'\rangle$, called the complement $TA_=$ of A, can be computed, and it satisfies $\overline{\mathcal{L}(A)} = \mathcal{L}(B)$.*

3.1. Complexity Analysis

The number of states of the new $TA_{\neq}$ B is $2^{|Q|}$. Nevertheless, the number of distinct disequality atoms which may appear in the constraints of B, that is, $N(B)$, coincides with $N(A)$. Thus, the amount of different constraints in the rules of B is bounded by $2^{N(A)}$. The number of rules in B is bounded by $2^{(Maxarity(\Sigma)+1)\cdot|Q|} \cdot |\Sigma| \cdot 2^{N(A)}$. In order to simplify arguments, we will simply consider the weaker expression $|B| \leq 2^{2\cdot Maxarity(\Sigma)\cdot|\Sigma|\cdot|A|}$. The time and size complexity of the complementation construction is $\mathcal{O}(2^{2\cdot Maxarity(\Sigma)\cdot|\Sigma|\cdot|A|})$. For further analysis, we also note that $h(B)$ coincides with $h(A)$.

4. TREE AUTOMATA WITH DISEQUALITY AND HOM EQUALITY CONSTRAINTS

Our aim is to define a new class of automata, with a certain kind of equality constraints, recognizing images of tree homomorphisms applied to regular languages. These are the tree automata with HOM constraints, denoted TA_{hom}, and they recognize the range of a bottom-up tree transducer. But we define a more general class including also arbitrary disequality constraints, denoted $\mathrm{TA}_{\neq,hom}$. The reason is that we will need to argue about the intersection of the languages represented by a TA_{hom} and a $\mathrm{TA}_{\neq}$.

lookahead

The definition of $\mathrm{TA}_{\neq,hom}$ has differences from the definition of $\mathrm{TA}_{\neq,=}$. The left-hand side of rules are not necessarily flat. Thus, they directly use information of states computed at relative positions deeper than 1. The disequality constraints are arbitrary, but the equality constraints are restricted. They always refer to positions with identical computed states. The rules are also labeled. The labels are not relevant at all for further definitions of run, pumping, etc. We will use them later, when intersecting two automata, for keeping the necessary information to recover the runs of the two original automata from a run of their intersection automaton.

Definition 4.1. A *tree automaton with disequality and HOM equality constraints*, $\mathrm{TA}_{\neq,hom}$ for short, is a tuple $A = \langle Q, \Sigma, F, \Delta\rangle$ satisfying that Q is a set of states, Σ is a signature, $F \subseteq Q$ is the subset of final states, and Δ is a set of labeled rules of the form $(I : s \xrightarrow{c} q)$, where I is the label, s is a term over $\mathcal{T}(\Sigma \cup Q) - Q$, interpreting the states of Q as 0-ary symbols, and c is a conjunction/set of atoms of the form $p_1 \neq p_2$ for arbitrary positions p_1, p_2, and atoms of the form $\hat{p}_1 = \hat{p}_2$, where $\hat{p}_1$ and $\hat{p}_2$ are different positions satisfying $s|_{\hat{p}_1} = s|_{\hat{p}_2} \in Q$. Moreover, for all positions $\hat{p}_1, \hat{p}_2, \hat{p}_3$, if $(\hat{p}_1 = \hat{p}_2)$ and $(\hat{p}_2 = \hat{p}_3)$ occur in c, then $(\hat{p}_1 = \hat{p}_3)$ also occurs in c. When no atom of the form $p_1 \neq p_2$ occurs in the rules of A, we say that A is a TA_{hom}.

The notions of size of constraints, rules, and automata are defined identically for $\mathrm{TA}_{\neq,hom}$ as for $\mathrm{TA}_{\neq,=}$ (see Definition 2.4). Labels of rules are not considered for the size.

By $N(A)$, we denote the number of distinct equality and disequality atoms in the rules of A. By $h(A)$, we denote the maximum among the heights of left-hand sides of rules in Δ and the lengths of positions occurring in the constraints of Δ, and write h when A is clear from the context. By $\mathit{Suff}_{\neq}$ we denote the set of suffixes of positions occurring in the disequality constraints of Δ. By $N_{Pos}(A)$, we denote the number of distinct positions in left-hand sides of rules, that is, $|\{p \mid \exists(I : s \xrightarrow{c} q) \in \Delta : (p \in Pos(s))\}|$.

Example 4.2. As an example of TA_{hom} consider $A_c = \langle Q, \Sigma_c, F, \Delta\rangle$, where $Q = \{q, q'\}$, $F = \{q'\}$, $\Sigma_c = \{f^{(2)}, g^{(2)}, a^{(0)}, b^{(0)}\}$, and Δ is the following set of rules.

—$\rho_1 = I_1 : a \to q$.
—$\rho_2 = I_2 : f(q, q) \xrightarrow{1=2} q$.
—$\rho_3 = I_3 : b \to q'$.
—$\rho_4 = I_4 : g(q, g(q, q')) \xrightarrow{1=2.1} q'$.

The set of terms $\mathcal{L}(A_c, q)$ is the language of the complete trees over $\{f^{(2)}, a^{(0)}\}$, as in Example 3.2. Terms in $\mathcal{L}(A_c)$ are of the form $g(t_1, g(t_1, g(t_2, g(t_2, \cdots g(t_n, g(t_n, b))\cdots))))$, where $t_1, t_2, \ldots, t_n \in \mathcal{L}(A_c, q)$. That is, we recognize sequences of pairs of identical complete trees.

If we replace ρ_4 by the rule:

$$—\rho_4' = I_4 : g(q, g(q, q')) \xrightarrow{\substack{1=2.1\\ 2.1\neq 2.2.1}} q',$$

then we get a $\mathrm{TA}_{\neq,hom}$. In this case, the accepted terms are of the form $g(t_1, g(t_1, g(t_2, g(t_2, \cdots g(t_n, g(t_n, b))\cdots))))$, where $t_1, t_2, \ldots, t_n \in \mathcal{L}(A_c, q)$ and $t_1 \neq t_2,\ t_2 \neq$

$t_3, \ldots, t_{n-1} \neq t_n$. Note that disequality constraints are unrestricted, in the sense that they are allowed to refer to arbitrary positions, even if such positions are not in the set of positions of the left-hand side of the rule. In contrast, equality constraints always refer to positions labeled with the same state in the left-hand side of the rule. For instance, a rule like

$$-\rho_4'' = I_4 : g(q, q') \xrightarrow{1=2.1} q',$$

which would allow us to recognize terms of the form $g(t, g(t, \cdots g(t, b) \cdots))$, is not a valid rule for a TA_{hom} since 2.1 is not a position of the term $g(q, q')$.

As in the case of tree automata with constraints, in order to define the concept of run of a $\mathrm{TA}_{\neq,hom}$ we define the alphabet for describing runs on terms, which are just terms with additional labels indicating which rule has been applied at each node. The difference with respect to the case of plain tree automata with constraints is that now we may also use function symbols in Σ for defining runs, but not only rules. Roughly speaking, this is because the rules are not applied at all the positions of a term, since the left-hand sides of rules are not necessarily flat. The projection π_Σ is overloaded to this alphabet.

Definition 4.3. Let $A = \langle Q, \Sigma, F, \Delta \rangle$ be a $\mathrm{TA}_{\neq,hom}$. The *alphabet of a run* of A is $\Sigma \cup \Delta$, where each rule $(I : s \xrightarrow{c} q)$ has the same arity as the one of the symbol $root(s)$. The *resulting state* of a term r in $\mathcal{T}(\Sigma \cup \Delta)$ of the form $(I : s \xrightarrow{c} q)(t_1, \ldots, t_m)$ is q. If r is not of such form, then the resulting state is undefined. The *projection* $\pi_\Sigma : \mathcal{T}(\Sigma \cup \Delta) \to \mathcal{T}(\Sigma)$ is recursively defined as $\pi_\Sigma(f(t_1, \ldots, t_m)) = f(\pi_\Sigma(t_1), \ldots, \pi_\Sigma(t_m))$ and as $\pi_\Sigma((I : s \xrightarrow{c} q)(t_1, \ldots, t_m)) = (root(s))(\pi_\Sigma(t_1), \ldots, \pi_\Sigma(t_m))$ for each rule $(I : s \xrightarrow{c} q)$ in Δ. For a term t in $\mathcal{T}(\Sigma \cup \Delta)$ and a position p in $Pos(t)$, we say that p is a Δ*-position* (of t) if $root(t|_p)$ is in Δ. If, in addition, p is not λ and the only proper prefix of p being a Δ position is λ, we say that p is a *first* Δ *position* (of t). We will usually denote with a hat ($\hat{p}$) the first Δ positions.

Runs of $\mathrm{TA}_{\neq,hom}$ are defined similarly to runs of $\mathrm{TA}_{\neq,=}$. One of the differences is that, for a given equality $(p_1 = p_2)$, while a $\mathrm{TA}_{\neq,=}$ checks for equality of the projected terms at the relative positions p_1 and p_2, a $\mathrm{TA}_{\neq,hom}$ checks for equality of the subruns at the relative positions p_1 and p_2, before projecting. This difference is just chosen for presentation purposes but is not relevant at all, since by interpreting equalities in the usual way we would have the same expressiveness. This is proved in Lemma 4.8 just for TA_{hom}. The extension of this result to $\mathrm{TA}_{\neq,hom}$ is trivial since disequality constraints directly refer to projected terms at relative positions, as already happens with $\mathrm{TA}_{\neq,=}$.

Definition 4.4. Let $A = \langle Q, \Sigma, F, \Delta \rangle$ be a $\mathrm{TA}_{\neq,hom}$. We define the concept of a *run* of A as a term in $\mathcal{T}(\Sigma \cup \Delta)$ satisfying certain conditions recursively as follows. Let $(I : s \xrightarrow{c} q)$ be a rule of Δ, where s is of the form $f(s_1, \ldots, s_m)$. Let $P = \{\hat{p}_1, \ldots, \hat{p}_n\}$ be the set of positions of s ~~such that~~ $\{\hat{p} : s|_{\hat{p}} \in Q)$ ~~holds~~. Let $q_1, \ldots, q_n$ be $s|_{\hat{p}_1}, \ldots, s|_{\hat{p}_n}$, respectively. Let $r_1, \ldots, r_n$ be runs of A with resulting states $q_1, \ldots, q_n$, respectively (recall that the resulting state has been defined in Definition 4.3). Let s' be $\pi_\Sigma(s[r_1]_{\hat{p}_1} \cdots [r_n]_{\hat{p}_n})$. Suppose $r_i = r_j$ for each $(\hat{p}_i = \hat{p}_j) \in c$, and $s'|_{p_1} \neq s'|_{p_2}$ for each $(p_1 \neq p_2) \in c$. Then, $(I : s \xrightarrow{c} q)(s_1, \ldots, s_m)[r_1]_{\hat{p}_1} \cdots [r_n]_{\hat{p}_n}$ is a *run* of A. Note that P is the set of first Δ positions of r.

By $\mathcal{L}(A, q)$, we denote the set of terms t for which there exists a run r of A with resulting state q such that $\pi_\Sigma(r) = t$. The language accepted by A, denoted $\mathcal{L}(A)$, is $\bigcup_{q \in F} \mathcal{L}(A, q)$.

Example 4.5 (*Following Example 4.2*). The terms $r_1 = \rho_3$, $r_2 = \rho_2(\rho_1, \rho_1)$, $r_3 = \rho_4(\rho_2(\rho_1, \rho_1), g(\rho_2(\rho_1, \rho_1), \rho_3)$ are runs of A_c. Their respective projections are $\pi_{\Sigma_c}(r_1) = b$, $\pi_{\Sigma_c}(r_2) = f(a, a)$ and $\pi_{\Sigma_c}(r_3) = g(f(a, a), g(f(a, a), b))$. Note that $r_1, r_3 \in \mathcal{L}(A_c)$ but $r_2 \notin \mathcal{L}(A_c)$.

The following lemma establishes that TA_{hom} can be used to represent images of regular languages through tree homomorphisms.

PROPOSITION 4.6. *Let $B = \langle Q, \Sigma_1, F, \Delta\rangle$ be a TA. Let $H : \mathcal{T}(\Sigma_1) \to \mathcal{T}(\Sigma_2)$ be a tree homomorphism. Then, a TA_{hom} A can be computed, satisfying $\mathcal{L}(A) = H(\mathcal{L}(B))$.*

PROOF. We define A as $\langle Q, \Sigma_2, F, \Delta'\rangle$ where Δ' is defined as follows. Let Δ'' be the set of rules of the form $\sigma(t_f) \xrightarrow{c} q$, for substitutions σ and terms t_f satisfying the following conditions.

—There exists a rule $f(q_1, \dots, q_m) \to q$ in Δ for a function symbol f of arity m such that $H(f(x_1, \dots, x_m)) = t_f$.
—Moreover, c is the conjunction of equalities $(\hat{p}_1 = \hat{p}_2)$ such that $\hat{p}_1$ and $\hat{p}_2$ are different, and $t_f|_{\hat{p}_1}$ and $t_f|_{\hat{p}_2}$ are the same variable.
—The substitution σ is $\{x_1 \mapsto q_1, \dots, x_m \mapsto q_m\}$.

Now, we define Δ' as the set obtained by closing Δ'' by the fixpoint computation $\Delta'' := \Delta'' \cup \{s \xrightarrow{c} q \mid \exists (s \xrightarrow{c} q'), (q' \to q) \in \Delta'' : s \notin Q\}$, and afterward by removing from Δ'' all rules of the form $q' \to q$. It is straightforward to see that $\mathcal{L}(A)$ equals $H(\mathcal{L}(B))$ by induction of the size of the involved terms.

Example 4.7. Let $A_r = \langle Q, \Sigma_r, F, \Delta\rangle$ be the TA satisfying $Q = \{q, q'\}$, $F = \{q'\}$, $\Sigma_r = \{a^{(0)}, b^{(0)}, f^{(1)}, g^{(2)}\}$ and $\Delta = \{a \to q,\ f(q) \to q, b \to q', g(q, q') \to q'\}$. That is, a TA recognizing the set of all trees of the form $g(t_1, g(t_2, \cdots g(t_n, b) \cdots))$, where $t_1, \dots, t_n$ are sequences of the form $f(f(\cdots f(a) \cdots))$. Let Σ_c be the signature of the TA_{hom} A_c of Example 4.2. Let $H : \mathcal{T}(\Sigma_r) \to \mathcal{T}(\Sigma_c)$ be the tree homomorphism defined as follows.

—$H(a) = a$.
—$H(b) = b$.
—$H(f(x_1)) = f(x_1, x_1)$.
—$H(g(x_1, x_2) = g(x_1, g(x_1, x_2))$.

The construction of Proposition 4.6 produces the TA_{hom} A_c, and it is easy to prove that $H(\mathcal{L}(A_r)) = \mathcal{L}(A_c)$ holds.

The following lemma establishes that a TA_{hom} is essentially a particular case of a $\text{TA}_=$, that is, for each TA_{hom} we can construct a $\text{TA}_=$ recognizing the same language. It will be useful to obtain a $\text{TA}_{\neq}$ recognizing the complement of a TA_{hom}. The only nontrivial point to prove this inclusion is the fact that equalities of a TA_{hom} ask for identical runs, while equalities of a $\text{TA}_=$ ask just for identically projected runs. But this can be solved by transforming a run of the constructed $\text{TA}_=$ in order to make those identically projected runs also identical as runs. Also: TA_{hom} rules are deep, while $TA_=$ rules are shallow

LEMMA 4.8. *Given a TA_{hom} $A_{hom} = \langle Q, \Sigma, F, \Delta\rangle$, a $\text{TA}_=$ A can be computed satisfying $\mathcal{L}(A) = \mathcal{L}(A_{hom})$. Moreover, the set of states Q' of A includes Q, and for each q in Q, $\mathcal{L}(A, q) = \mathcal{L}(A_{hom}, q)$.*

PROOF. For making the proof easier to read, the Q' we will define does not include Q, but it includes a set $\{\mathbf{q}_q \mid q \in Q\}$, and by renaming each $\mathbf{q}_q$ to q the result holds.

We define A as $\langle Q', \Sigma, F', \Delta'\rangle$, where Q' is $\{\mathbf{q}_t \mid t$ is a proper subterm of a left-hand side of a rule in$\Delta\} \cup \{\mathbf{q}_q \mid q \in Q\}$, F' is $\{\mathbf{q}_q \mid q \in F\}$ and Δ' is $\{f(\mathbf{q}_{t_1}, \dots, \mathbf{q}_{t_m}) \to \mathbf{q}_{f(t_1,\dots,t_m)} \mid$

$\mathbf{q}_{f(t_1,\ldots,t_m)} \in Q'\} \cup \{f(\mathbf{q}_{t_1},\ldots,\mathbf{q}_{t_m}) \xrightarrow{c} \mathbf{q}_q \mid (f(t_1,\ldots,t_m) \xrightarrow{c} q) \in \Delta\}$. It remains to prove $\mathcal{L}(A) = \mathcal{L}(A_{hom})$, and, more in general, $\mathcal{L}(A, \mathbf{q}_q) = \mathcal{L}(A_{hom}, q)$ for each q in Q.

Since the inclusion $\mathcal{L}(A, \mathbf{q}_q) \supseteq \mathcal{L}(A_{hom}, q)$ is completely straightforward, we just prove $\mathcal{L}(A, \mathbf{q}_q) \subseteq \mathcal{L}(A_{hom}, q)$, and do it by induction on the size of the involved terms. Let t be a term in $\mathcal{L}(A, \mathbf{q}_q)$. Let r be a run of A with resulting state $\mathbf{q}_q$ such that $\pi_\Sigma(r) = t$. Let $root(r)$ be of the form $f(\mathbf{q}_{s_1},\ldots,\mathbf{q}_{s_m}) \xrightarrow{c} \mathbf{q}_q$. By the definition of A, the existence of this rule in Δ' implies the existence of the rule $f(s_1,\ldots,s_m) \xrightarrow{c} q$ in Δ. Let $\hat{p}_1,\ldots,\hat{p}_n$ be the positions $\hat{p}_i$ satisfying $(f(s_1,\ldots,s_m)|_{\hat{p}_i} \in Q)$. Again, by the definition of A, r is necessarily of the form $(f(\mathbf{q}_{s_1},\ldots,\mathbf{q}_{s_m}) \xrightarrow{c} \mathbf{q}_q)(s'_1,\ldots,s'_m)[r_1]_{\hat{p}_1}\cdots[r_n]_{\hat{p}_n}$, where each s'_i is the term satisfying $Pos(s'_i) = Pos(s_i)$, and for each p in $Pos(s_i)$, either $s_i|_p$ is in Q and $root(s'_i|_p)$ is $\mathbf{q}_{s_i|_p}$, or $root(s_i|_p)$ is a symbol g in Σ of a certain arity k and $root(s'_i|_p)$ is $g(\mathbf{q}_{s_i|_{p.1}},\ldots,\mathbf{q}_{s_i|_{p.k}}) \rightarrow \mathbf{q}_{s_i|_p}$. Moreover, $r_1,\ldots,r_n$ are runs of A with resulting states $\mathbf{q}_{s|_{\hat{p}_1}},\ldots,\mathbf{q}_{s|_{\hat{p}_n}}$, respectively, where s is $f(s_1,\ldots,s_m)$, and such that $\pi_\Sigma(r_1) = t|_{\hat{p}_1},\ldots,\pi_\Sigma(r_n) = t|_{\hat{p}_n}$ hold. By induction hypothesis, there exist runs $r'_1,\ldots,r'_n$ of A_{hom} with resulting states $s|_{\hat{p}_1},\ldots,s|_{\hat{p}_n}$, respectively, such that $\pi_\Sigma(r'_1) = t|_{\hat{p}_1},\ldots,\pi_\Sigma(r'_n) = t|_{\hat{p}_n}$ hold. Note that, it could be the case that, for some $1 \le i < j \le n$, a constraint $(\hat{p}_i = \hat{p}_j)$ occurs in c, thus $t|_{\hat{p}_i} = t|_{\hat{p}_j}$ holds, but the runs r'_i and r'_j are different. For this reason, $(f(s_1,\ldots,s_m) \xrightarrow{c} q)(s_1,\ldots,s_m)[r'_1]_{\hat{p}_1}\cdots[r'_n]_{\hat{p}_n}$ is not necessarily a run. Let $r''_1,\ldots,r''_n$ be defined inductively as follows for each j in $\{1,\ldots,n\}$. If, for some $i < j$, a constraint $(\hat{p}_i = \hat{p}_j)$ occurs in c, then r''_j is defined as r''_i. Otherwise, r''_j is defined as r'_j. It is straightforward to check that $r'' = (f(s_1,\ldots,s_m) \xrightarrow{c} q)(s_1,\ldots,s_m)[r''_1]_{\hat{p}_1}\cdots[r''_n]_{\hat{p}_n}$ is a run of A_{hom} with resulting state q and such that $\pi_\Sigma(r'')$ is t.

From Lemmas 4.8, 3.3, and 3.4, the following corollary follows.

COROLLARY 4.9. *Given a TA_{hom} $A_{hom} = \langle Q, \Sigma, F, \Delta\rangle$, a $TA_{\neq}$ A can be computed satisfying the following conditions.*

—*The set of states of A is $2^{Q'}$, where Q' is a set including Q.*
—*For each state q in Q, and each state S in $2^{Q'}$ containing q and each term t in $\mathcal{L}(A, S)$, t is not in $\mathcal{L}(A_{hom}, q)$.*
—*For each term t in $\mathcal{T}(\Sigma)$, there exists a run r of A satisfying $\pi_\Sigma(r) = t$ and such that, for each $p \in Pos(r)$, $r|_p$ is a run of A with a resulting state S satisfying $S \cap Q = \{q \in Q \mid t|_p \notin \mathcal{L}(A_{hom}, q)\}$. Moreover, the run r is deterministic, that is, if two positions $p_1, p_2 \in Pos(t)$ satisfy $t|_{p_1} = t|_{p_2}$, then $r|_{p_1} = r|_{p_2}$.*

Definition 4.10. Given a TA_{hom} A_{hom}, by $\overline{A_{hom}}$ we define the $TA_{\neq}$ provided by Corollary 4.9.

Now we provide the necessary definitions and lemmas to intersect a $TA_{\neq}$ that admits deterministic accepting runs and a TA_{hom}, thus producing a $TA_{\neq,hom}$, and to intersect the corresponding runs. Here is when the labels play an important role, keeping the necessary information of the original runs in order to recover them by projection from a run of the produced $TA_{\neq,hom}$. We do not give the proofs of these lemmas, since they are rather straightforward. Note that the labels of the initial TA_{hom} are removed.

Definition 4.11. Let $A = \langle Q_A, \Sigma, F_A, \Delta_A\rangle$ be a TA_{hom}. Let $B = \langle Q_B, \Sigma, F_B, \Delta_B\rangle$ be a $TA_{\neq}$ that admits deterministic accepting runs. We define the $TA_{\neq,hom}$ $A \cap B$ as $\langle Q_A \times Q_B, \Sigma, F_A \times F_B, \Delta\rangle$, where Δ is the set of rules $(I : s \xrightarrow{c} \langle q_A, q_B\rangle)$ satisfying the following conditions.

not necessary for the construction (but necessary for correctness)

—The term s is in $\mathcal{T}(\Sigma \cup (Q_A \times Q_B))$.
—There exists a rule $(s' \xrightarrow{c'} q_A)$ in Δ_A satisfying $Pos_\Sigma(s') = Pos_\Sigma(s)$ and for each $p \in Pos_\Sigma(s')$, $root(s'|_p) = root(s|_p)$ holds.
—For each $\hat{p} \in Pos(s)$ such that $s|_{\hat{p}}$ is of the form $\langle q'_A, q'_B\rangle$, $s'|_{\hat{p}}$ is q'_A.
—The label $I : Pos_\Sigma(s) \to \Delta_B$ is a mapping satisfying the following conditions for each p in $Pos_\Sigma(s)$: If $I(p)$ is of the form $(f(q_{1B}, \ldots, q_{mB}) \xrightarrow{c''} q'_B)$, then $root(s|_p)$ is f, and for each i in $\{1, \ldots, m\}$, either $s|_{p.i}$ is a state of the form $\langle \_, q_{iB}\rangle$, or $I(p.i)$ is defined as a rule with right-hand side q_{iB}.
—The constraint c is the set $c' \cup \{p.p_1 \neq p.p_2 \mid p \in Pos_\Sigma(s) \wedge \exists t, c'', q'_B : (I(p) = (t \xrightarrow{c''} q'_B) \wedge (p_1 \neq p_2) \in c'')\}$. Moreover, for each $\hat{p}_1 = \hat{p}_2$ in c', $s|_{\hat{p}_1} = s|_{\hat{p}_2}$ holds.

Example 4.12. Let $A_{ex} = \langle Q, \Sigma, F, \Delta\rangle$ be a TA$_{hom}$ with $Q = F = \{q\}$, $\Sigma = \{f^{(3)}, h^{(1)}, a^{(0)}\}$, and Δ the set of rules

—$\rho_1 = I_1 : a \to q$
—$\rho_2 = I_2 : h(a) \to q$
—$\rho_3 = I_3 : h(f(q, q, q)) \xrightarrow{1.1=1.2} q$.

Let $A_d = \langle Q', \Sigma, F', \Delta'\rangle$ be a TA$_{\neq}$ with $Q' = F' = \{q'\}$, the same Σ as A_{ex}, and where Δ' contains the rules

—$\bar{\rho}_1 = a \to q'$,
—$\bar{\rho}_2 = f(q', q', q') \xrightarrow{2\neq 3} q'$,
—$\bar{\rho}_3 = h(q') \xrightarrow{1.1\neq 1.3} q'$,
—$\bar{\rho}_4 = h(q') \xrightarrow{1.1\neq 1.2} q'$.

Note that A_d admits deterministic accepting runs since it has only one state. We can compute the TA$_{\neq,hom}$ $\hat{A} = A_{ex} \cap A_d$ obtaining $\langle Q \times Q', \Sigma, F \times F', \Delta\rangle$, where Δ contains the following rules.

—$\hat{\rho}_1 = I_1 : a \to \langle q, q'\rangle$, with label $I_1(\lambda) = \bar{\rho}_1$.
—$\hat{\rho}_2 = I_2 : h(a) \xrightarrow{1.1\neq 1.3} \langle q, q'\rangle$, with label $I_2(\lambda) = \bar{\rho}_3$, $I_2(1) = \bar{\rho}_1$. (Note that the constraint is always satisfied.)
—$\hat{\rho}_3 = I_3 : h(a) \xrightarrow{1.1\neq 1.2} \langle q, q'\rangle$, with label $I_3(\lambda) = \bar{\rho}_4$, $I_3(1) = \bar{\rho}_1$. (Note that the constraint is always satisfied.)
—$\hat{\rho}_4 = I_4 : h(f(\langle q, q'\rangle, \langle q, q'\rangle, \langle q, q'\rangle)) \xrightarrow{\substack{1.1=1.2\\ 1.2\neq 1.3\\ 1.1\neq 1.3}} \langle q, q'\rangle$, with label $I_4(\lambda) = \bar{\rho}_3$, $I_4(1) = \bar{\rho}_2$.
—$\hat{\rho}_5 = I_5 : h(f(\langle q, q'\rangle, \langle q, q'\rangle, \langle q, q'\rangle)) \xrightarrow{\substack{1.1=1.2\\ 1.2\neq 1.3\\ 1.1\neq 1.2}} \langle q, q'\rangle$, with label $I_5(\lambda) = \bar{\rho}_4$, $I_5(1) = \bar{\rho}_2$, which is useless because the constraint is unsatisfiable.

PROPOSITION 4.13. *Let A be a TA$_{hom}$. Let B be a TA$_{\neq}$ that admits deterministic accepting runs. Then, $\mathcal{L}(A \cap B) = \mathcal{L}(A) \cap \mathcal{L}(B)$.*

In the proof of decidability of HOM, the intersection $A \cap B$ is used to combine a run r_A of a TA$_{hom}$ A with a run r_B of a TA$_{\neq}$ B. Next, the resulting run $r_{A\cap B}$ is modified to a new run $r'_{A\cap B}$. From this new run we need to retrieve runs r'_A of A and r'_B of B. The following definition shows how to retrieve these runs.

Definition 4.14. Let $A = \langle Q_A, \Sigma, F_A, \Delta_A\rangle$ be a TA$_{hom}$. Let $B = \langle Q_B, \Sigma, F_B, \Delta_B\rangle$ be a TA$_{\neq}$ that admits deterministic accepting runs. Let $r = (I : s \xrightarrow{c} \langle q, q'\rangle)(s_1, \ldots, s_m)$

$[r_1]_{\hat{p}_1}\cdots[r_n]_{\hat{p}_n}$ be a run of $A\cap B$, where $c=c_=\cup c_{\neq}$ for sets of equality and disequality constraints $c_=$ and $c_{\neq}$, respectively.

The *projected run* $\pi_{hom}(r)$ is defined recursively as $(s\xrightarrow{c_=}q)(s_1,\ldots,s_m)[\pi_{hom}(r_1)]_{\hat{p}_1}\cdots[\pi_{hom}(r_n)]_{\hat{p}_n}$.

The *projected run* $\pi_{\neq}(r)$ is defined recursively as follows. Let $r'_1,\ldots,r'_n$ be $\pi_{\neq}(r_1),\ldots,\pi_{\neq}(r_n)$, respectively. Let s' be any term satisfying $Pos(s')=Pos(s)$, and for each $p\in Pos_\Sigma(s)$ it holds that $root(s'|_p)$ is $I(p)$. Then, $\pi_{\neq}(r)$ is defined as $s'[r'_1]_{\hat{p}_1}\cdots[r'_n]_{\hat{p}_n}$.

The following proposition states that the retrieved runs are on the same term.

PROPOSITION 4.15. *Let $A=\langle Q_A,\Sigma,F_A,\Delta_A\rangle$ be a TA_{hom}. Let $B=\langle Q_B,\Sigma,F_B,\Delta_B\rangle$ be a $TA_{\neq}$ that admits deterministic accepting runs. Let r be a run of $A\cap B$. Then, $\pi_{hom}(r)$ is a run of A, and $\pi_{\neq}(r)$ is a run of B. Moreover, $\pi_\Sigma(r)=\pi_\Sigma(\pi_{hom}(r))=\pi_\Sigma(\pi_{\neq}(r))$.*

The following definition shows how r_A and r_B are combined.

Definition 4.16. Let $A=\langle Q_A,\Sigma,F_A,\Delta_A\rangle$ be a TA_{hom}. Let $B=\langle Q_B,\Sigma,F_B,\Delta_B\rangle$ be a $TA_{\neq}$ that admits deterministic accepting runs. Let t be a term in $\mathcal{T}(\Sigma)$. Let $r_A=(s\xrightarrow{c}q)(s_1,\ldots,s_m)[r_1]_{\hat{p}_1}\cdots[r_n]_{\hat{p}_n}$ be a run of A on t, where $s=f(s_1,\ldots,s_m)$. Let r_B be a deterministic run of B on t. The *intersected run* $r=r_A\cap r_B$ is defined recursively to satisfy the following conditions.

—$Pos(r)=Pos(r_A)=Pos(r_B)$.
—For each i in $\{1,\ldots,n\}$, $r|_{\hat{p}_i}=(r_i\cap(r_B|_{\hat{p}_i}))$ holds.
—For each p in $Pos_\Sigma(s)-\{\lambda\}$, $root(r|_p)=root(r_A|_p)$ holds.
—$root(r)$ is $(I:s'\xrightarrow{c'}\langle q,q'\rangle)$, where q' is the resulting state of r_B, $Pos(s)=Pos(s')$ holds, for each p in $Pos_\Sigma(s)$, the equalities $root(s'|_p)=root(s|_p)$ and $I(p)=root(r_B|_p)$ hold, and for each $\hat{p}$ in $Pos_Q(s)$, the term $s'|_{\hat{p}}$ is $\langle s|_{\hat{p}},q'_B\rangle$, where q'_B is the resulting state of the corresponding $r_B|_{\hat{p}}$. Moreover, c' is the set $c\cup\{p.p_1\neq p.p_2\mid p\in Pos_\Sigma(s)\wedge\ \exists t,c'',q'_B:(I(p)=(t\xrightarrow{c''}q'_B)\wedge(p_1\neq p_2)\in c'')\}$. Furthermore, for each $\hat{p}_1=\hat{p}_2$ in c', $s'|_{\hat{p}_1}=s'|_{\hat{p}_2}$ holds.

Example 4.17 (*Continues Example 4.12*). The following run $\hat{r}$ is a run of $\hat{A}$: $\hat{r}=\hat{\rho}_4(f(\hat{\rho}_1,\hat{\rho}_1,\hat{\rho}_2(a)))$, with projection $\pi_\Sigma(\hat{r})=h(f(a,a,h(a)))$, and it is the intersection of the runs $\pi_{hom}(\hat{r})=\rho_3(f(\rho_1,\rho_1,\rho_2(a)))$ (but without labels) and $\pi_{\neq}(\hat{r})=\bar{\rho}_3(\bar{\rho}_2(\bar{\rho}_1,\bar{\rho}_1,\bar{\rho}_3(\bar{\rho}_1)))$.

The following proposition states that projecting intersected runs retrieve the original runs.

PROPOSITION 4.18. *Let A be a TA_{hom}. Let B be a $TA_{\neq}$. Let r_A,r_B be runs of A and B, respectively, on the same term, and where r_B is deterministic. Then, $\pi_{\neq}(r_A\cap r_B)=r_B$ and $\pi_{hom}(r_A\cap r_B)=r_A$.*

Consider that we obtain $r_{A\cap B}$ from r_A and r_B and next transform $r_{A\cap B}$ into $r'_{A\cap B}$ in a way such that the labels at some positions in $r_{A\cap B}$ and $r'_{A\cap B}$ coincide. In the proof of decidability of HOM, under such assumptions, it will be used that then, the projected r'_A and r'_B also coincide with r_A and r_B, respectively, at such positions.

LEMMA 4.19. *Let $A=\langle Q_A,\Sigma,F_A,\Delta_A\rangle$ be a TA_{hom}. Let $B=\langle Q_B,\Sigma,F_B,\Delta_B\rangle$ be a $TA_{\neq}$ that admits deterministic accepting runs. Let r_1,r_2 be runs of $A\cap B$. Let p be a position in $Pos(r_1)\cap Pos(r_2)$ such that $root(r_1|_p)=root(r_2|_p)$.*

Then, $root((\pi_{hom}(r_1))|_p)=root((\pi_{hom}(r_2))|_p)$. Moreover, if for each prefix p' of p, $root(r_1|_{p'})=root(r_2|_{p'})$, then $root((\pi_{\neq}(r_1))|_p)=root((\pi_{\neq}(r_2))|_p)$.

4.1. Complexity Analysis

We first analyze the complexity of the construction of Proposition 4.6. There, we obtain a TA_{hom} A from a TA B and a tree homomorphism H. The set of states of A is the same as the one of B. Also note that $N_{Pos}(A)$ coincides with $N_{Pos}(H)$, $N(A)$ coincides with $N(H)$, $h(A)$ coincides with the maximum between 1 and $h(H)$. The new Δ' has a rule of the form $\sigma(t_f) \xrightarrow{c} q$ for each rule $f(q_1, \ldots, q_m) \to q$ in Δ, where t_f is $H(f(x_1, \ldots, x_m))$. Note that each of such $|\sigma(t_f)|$ coincides with the corresponding $|t_f|$, which is bounded by $N_{Pos}(H)$. Thus, $|\Delta'| = |\Delta|$ and the size of each rule is bounded by $N_{Pos}(H) + 2 \cdot N(H) \cdot h(H) + 1$. But the closure applied to rules of the form $q_1 \to q_2$ produced an additional quadratic increase. Hence, $|A|$ can be bounded by $(|B| \cdot 3 \cdot |H|^2)^2$ (by an adequate representation of A this bound could be smaller). The time complexity for the construction of A is $\mathcal{O}(|B|^2 \cdot |H|^4)$.

Now we analyze the complexity of the construction of Lemma 4.8. There, a $\mathrm{TA}_=$ A is constructed recognizing the same language as a given TA_{hom} A_{hom}. It holds that $N(A)$ coincides with $N(A_{hom})$ and $h(A)$ coincides with $h(A_{hom})$. The new set of states Q' has a state for each state of Q and a state for each subterm of a left-hand side of a rule of A_{hom}. Thus, $|Q'| \leq |Q| + \|\Delta\|$ holds. For each state in Q' and not in Q, there is just one new rule whose left-hand side has height 1 and whose constraint is empty. Such a rule precisely comes from a subterm occurring in a left-hand side of a rule of Δ. In fact, each rule of Δ has also a version in Δ' with a smaller size. Note that the sum of sizes of left-hand sides of rules in Δ' is bounded by twice the sum of sizes of left-hand sides of rules in Δ. Also, the sum of sizes of right-hand sides of rules in Δ' is bounded by $\|\Delta\|$, and Δ' contains the same constraints as Δ. Thus, $\|\Delta'\| \leq 3 \cdot \|\Delta\|$. In summary, $|A| = |Q'| + \|\Delta'\| \leq |Q| + \|\Delta\| + 3 \cdot \|\Delta\| = |Q| + 4 \cdot \|\Delta\| \leq 4 \cdot |A_{hom}|$. The construction takes time $\mathcal{O}(|A_{hom}|)$.

From the previous analysis, it follows that the composition of the constructions from Proposition 4.6 and Lemma 4.8 allows to obtain a $\mathrm{TA}_=$ A from a TA B and a tree homomorphism H such that $\mathcal{L}(A) = H(\mathcal{L}(B))$ and $|A| \leq 36 \cdot |B|^2 \cdot |H|^4$. The time complexity is $\mathcal{O}(|B|^2 \cdot |H|^4)$. Further, if we compose also the construction of complementation of Definition 3.1, and according to the discussion of Section 3.1, a $\mathrm{TA}_{\neq}$ A can be obtained such that $\mathcal{L}(A) = \overline{H(\mathcal{L}(B))}$ and $|A| \leq 2^{72 \cdot Maxarity(\Sigma) \cdot |\Sigma| \cdot |B|^2 \cdot |H|^4}$. The time and space complexity of the construction is $\mathcal{O}(2^{72 \cdot Maxarity(\Sigma) \cdot |\Sigma| \cdot |B|^2 \cdot |H|^4})$. Also, $N(A)$ coincides with $N(H)$ and $h(A)$ coincides with $h(H)$.

Now we analyze the construction of Definition 4.11. There, a $\mathrm{TA}_{\neq,hom}$ $A \cap B$ is obtained from a TA_{hom} A and a $\mathrm{TA}_{\neq}$ B. The number of states of $A \cap B$ is $|Q_A| \cdot |Q_B|$. The value $N_{Pos}(A \cap B)$ coincides with $N_{Pos}(A)$. Any equality occurring in the constraints of $A \cap B$ occurs in the constraints of A. Any disequality occurring in the constraints of $A \cap B$ is of the form $p.p_1 \neq p.p_2$, where p is a position of a left-hand side of a rule of A, and $p_1 \neq p_2$ is a disequality occurring in a constraint of B. Thus, $N(A \cap B) \leq N(A) + N_{Pos}(A) \cdot N(B)$ and $h(A \cap B) \leq h(A) + h(B)$. The number of different possible conjunctions of constraints in $A \cap B$ is bounded by $2^{N(A \cap B)}$, which in turn is bounded by $2^{N(A) + N_{Pos}(A) \cdot N(B)}$. The number of rules in $A \cap B$ is bounded by $(|\Sigma| + |Q_A| \cdot |Q_B|)^{1 + N_{Pos}(A)} \cdot 2^{N(A) + N_{Pos}(A) \cdot N(B)}$. Note that the $|\Sigma|$ in the last expression can be assumed as bounded by $|A|$, since all alphabet symbols occurring in the new automata were already occurring in A. In order to work with simpler formulas, we just consider the weaker relation $|A \cap B| \leq (|A| \cdot |B|)^{1 + N_{Pos}(A) + N(A) + N_{Pos}(A) \cdot N(B)}$. The time and space complexity of the construction of $A \cap B$ is $\mathcal{O}((|A| \cdot |B|)^{1 + N_{Pos}(A) + N(A) + N_{Pos}(A) \cdot N(B)})$.

5. PUMPINGS

In this section, we prove that emptiness and finiteness are decidable for $\mathrm{TA}_{\neq,hom}$. Section 5.1 is devoted to emptiness, while Section 5.2 is devoted to finiteness. We deal with both problems in the same section because the respective proofs use similar techniques and have common intermediate results, which are developed here, before such subsections. For the emptiness problem, we proceed by assuming that the language of a given $\text{TA}_{\neq,hom}$ A is not empty, and then prove that an accepting run of A minimum with respect to a well founded ordering $\gg$ (essentially the size ordering plus a notion for comparing runs with equal size) has a size computationally bounded by A. For the finiteness problem, we directly give a computational bound and prove that, if an accepting run is higher than this bound, then there is another accepting run even much higher than the starting one. This is done by contradiction, assuming that the new higher run does not exist. Thus, in both cases, the starting point is an accepting run r, minimum in size or higher than a computational bound, depending on the case. In both cases, we modify r in several ways getting new terms smaller in size or bigger in height, depending on the case. By the assumed conditions, the new terms cannot be accepting runs. This allows us to infer properties about r in order to bound its size or its height.

The modifications of r are based on the notion of pumping. Pumping is a traditional concept in automata theory, and in particular, they are very useful to reason about tree automata. The basic idea is to convert a given run r into another run by replacing a subterm at a certain position p in r by a run r', thus obtaining a run $r[r']_p$. For plain tree automata, the necessary and sufficient condition to ensure that $r[r']_p$ is a run is that the resulting states of $r|_p$ and r' coincide, since the correct application of a rule at a certain position depends only on the resulting states of the subruns of the direct children. When the tree automaton has equality and disequality constraints, the constraints may be falsified when replacing a subrun by a new run. For $\text{TA}_{\neq,hom}$, we define a notion of pumping ensuring that the equality constraints are satisfied, while nothing is guaranteed for the disequality constraints.

Definition 5.1. Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $\text{TA}_{\neq,hom}$. Let r, r' be runs of A, and let $\bar{p}$ be a position such that $r|_{\bar{p}}$ is a run of A and $root(r|_{\bar{p}}) = root(r')$ holds. The *pumping* of r' into r at position $\bar{p}$, denoted $r[[r']]_{\bar{p}}$, is a term in $\mathcal{T}(\Sigma \cup \Delta)$ defined recursively as follows. Let r be of the form $(I : s \xrightarrow{c} q)(s_1, \ldots, s_m)[r_1]_{\hat{p}_1} \cdots [r_n]_{\hat{p}_n}$.

Suppose first that $\bar{p}$ is λ. Then, $r[[r']]_{\bar{p}}$ is r'.

Otherwise, suppose that $\bar{p}$ is of the form $\hat{p}_i.\tilde{p}$ for some i in $\{1, \ldots, n\}$, and let r'_i be $r_i[[r']]_{\tilde{p}}$. Then, $r[[r']]_{\bar{p}}$ is $r[r'_1]_{\hat{p}_1} \cdots [r'_n]_{\hat{p}_n}$, where each r'_j is defined as r'_i in the case where j is i or $(\hat{p}_i = \hat{p}_j)$ occurs in c, and r'_j is defined as r_j otherwise.

The case where $\bar{p}$ is not λ and no $\hat{p}_i$ is a prefix of $\bar{p}$ is not possible, by the condition that $r|_{\bar{p}}$ is a run.

Note that $r[[r']]_{\bar{p}}$ is just a new term in $\mathcal{T}(\Sigma \cup \Delta)$. Nevertheless, by abuse of notation, when we write $r[[r']]_{\bar{p}}$ we sometimes consider it as the action of constructing a pumping by assuming that r, r', and $\bar{p}$ are still explicit.

The following lemma argues that a pumping is just a simultaneous replacement of occurrences of one term by another new term.

LEMMA 5.2. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $\text{TA}_{\neq,hom}$. Let r be a run of A, and let $\bar{p}$ be a position such that $r|_{\bar{p}}$ is a run of A. Then, there exist parallel positions $\bar{p}_1, \ldots, \bar{p}_k$ including $\bar{p}$ such that, for all run r' satisfying $root(r') = root(r|_{\bar{p}})$, then $r|_{\bar{p}_1} = \cdots = r|_{\bar{p}_k}$ and $r[[r']]_{\bar{p}} = r[r']_{\bar{p}_1} \cdots [r']_{\bar{p}_k}$. Moreover, if a prefix p of $\bar{p}$ satisfies that $r|_p$ is a run and for some i in $\{1, \ldots, k\}$, p is a prefix of $\bar{p}_i$ and $|\bar{p}_i - p| \leq h(A)$, then, $|\{p' \mid p \leq p' < \bar{p} \wedge r|_{p'}$ is a run$\}|$ is bounded by $h(A)$.*

PROOF. It is easy to verify that the set of positions $reppos(r, \bar{p})$, defined recursively as follows for a given run r and a position $\bar{p}$ under the previous assumptions, satisfies the statement. $reppos(r, \lambda)$ is defined as $\{\lambda\}$, and $reppos((I : s \xrightarrow{c} q)(s_1, \ldots, s_m)[r_1]_{\hat{p}_1} \cdots [r_n]_{\hat{p}_n}, \hat{p}_i.\tilde{p})$ is defined as $\{\hat{p}_j.\tilde{p}' \mid (j = i \vee (\hat{p}_i = \hat{p}_j) \in c) \wedge \tilde{p}' \in reppos(r_i, \tilde{p})\}$

While the definition of pumping preserves satisfaction of equalities in the constraints, nothing is guaranteed for disequalities. Thus, the new term might not be a run because some disequality might be falsified after the replacement. The following lemma shows that one of the falsified disequalities is in the constraint of a rule applied at a position which is a prefix of the pumping position $\bar{p}$.

LEMMA 5.3. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq,hom}$. Let r, r' be runs of A, and let $\bar{p}$ be a position such that $r|_{\bar{p}}$ is a run of A and $root(r|_{\bar{p}}) = root(r')$. Suppose that $r[[r']]_{\bar{p}}$ is not a run. Then, there exists a position p such that $root((r[[r']]_{\bar{p}})|_p)$ is of the form $(I : s \xrightarrow{c} q)$, and there exists $(p_1 \neq p_2)$ in c satisfying $\pi_\Sigma(r[[r']]_{\bar{p}})|_{p.p_1} = \pi_\Sigma(r[[r']]_{\bar{p}})|_{p.p_2}$. Moreover, p can be chosen to be a prefix of $\bar{p}$.*

PROOF. We prove it by induction on $height(r)$. If $\bar{p}$ is λ, then, $r[[r']]_{\bar{p}}$ is r', which is a run, thus contradicting the statement. Hence, without loss of generality, assume that r is of the form $(I : s \xrightarrow{c} q)(s_1, \ldots, s_m)[r_1]_{\hat{p}_1} \cdots [r_n]_{\hat{p}_n}$ and that $\bar{p}$ is of the form $\hat{p}_1.\tilde{p}$.

If $r_1[[r']]_{\tilde{p}}$ is not a run, then, by induction hypothesis, there exists a prefix p' of $\tilde{p}$ such that $root((r_1[[r']]_{\tilde{p}})|_{p'})$ is of the form $(I' : s' \xrightarrow{c'} q')$, and there exists $(p_1 \neq p_2)$ in c' satisfying $\pi_\Sigma(r_1[[r']]_{\tilde{p}})|_{p'.p_1} = \pi_\Sigma(r_1[[r']]_{\tilde{p}})|_{p'.p_2}$. By defining p as $\hat{p}_1.p'$ the lemma follows.

Otherwise, assume that $r_1[[r']]_{\tilde{p}}$ is a run. Since $r[[r']]_{\bar{p}}$ is not a run, this can only be due to the existence of a disequality $(p_1 \neq p_2)$ in c satisfying $\pi_\Sigma(r[[r']]_{\bar{p}})|_{p_1} = \pi_\Sigma(r[[r']]_{\bar{p}})|_{p_2}$. By defining p as λ, the lemma follows.

A pumping applied on an accepting run minimum in size, and producing a smaller term in size, cannot give us an accepting correct run. Similarly, a pumping applied on an accepting run maximum in height and producing a bigger term in height, cannot give us an accepting correct run. In both cases, a disequality must be falsified. This will allow us to bound the size or height of the original run. To this end, first we need to study how a disequality might be falsified.

Consider the case where a pumping $r[[r']]_{\bar{p}} = r[r']_{\bar{p}_1} \cdots [r']_{\bar{p}_n}$ falsifies a disequality $(p_1 \neq p_2)$ occurring in a rule applied at a certain position p. This means that the terms $\pi_\Sigma(r)|_{p.p_1}, \pi_\Sigma(r)|_{p.p_2}$ are different, but the terms $\pi_\Sigma(r[[r']]_{\bar{p}})|_{p.p_1}, \pi_\Sigma(r[[r']]_{\bar{p}})|_{p.p_2}$ are identical. It might be the case that some position $\bar{p}_i$ is a proper prefix of either $p.p_1$ or $p.p_2$. In this case, we say that the pumping close-falsifies the disequality. Otherwise, we say that the pumping far-falsifies the disequality. Both situations are treated differently, and we define them in detail as follows.

Definition 5.4. Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $\mathrm{TA}_{\neq,hom}$. Let r be a run of A, and let $\bar{p}$ be a position such that $r|_{\bar{p}}$ is a run of A.

According to Lemma 5.2, there exist parallel positions $\bar{p}_1 = \bar{p}, \ldots, \bar{p}_k$ such that $r[[r']]_{\bar{p}} = r[r']_{\bar{p}_1} \cdots [r']_{\bar{p}_k}$ for any run r' satisfying $root(r|_{\bar{p}}) = root(r')$. These positions are called the *replaced positions* of $r[[r']]_{\bar{p}}$. We will usually denote them with a bar ($\bar{p}_i$), or with a tilde ($\tilde{p}_i$).

Let r' be a run such that $root(r|_{\bar{p}}) = root(r')$ holds but $r[[r']]_{\bar{p}}$ is not a run. According to Lemma 5.3, there exists a position p such that $root((r[[r']]_{\bar{p}})|_p)$ is of the form $(I : s \xrightarrow{c} q)$, and there exists $(p_1 \neq p_2)$ in c satisfying $\pi_\Sigma(r[[r']]_{\bar{p}})|_{p.p_1} = \pi_\Sigma(r[[r']]_{\bar{p}})|_{p.p_2}$. In such a case, we say that the pumping $r[[r']]_{\bar{p}}$ *falsifies* $(p_1 \neq p_2)$ at p. Moreover, if some $\bar{p}_i$ is a proper prefix of $p.p_1$ ($p.p_2$), we say that the pumping $r[[r']]_{\bar{p}}$ *close-falsifies* $(p_1 \neq p_2)$ at $\langle p, \bar{p}_i, p.p_1\rangle$ (at $\langle p, \bar{p}_i, p.p_2\rangle$). Note that it may happen that $r[[r']]_{\bar{p}}$ close-falsifies $(p_1 \neq p_2)$ at $\langle p, \bar{p}_i, p.p_1\rangle$ and at $\langle p, \bar{p}_i, p.p_2\rangle$. In the case where $r[[r']]_{\bar{p}}$ falsifies $(p_1 \neq p_2)$

at p but no $\bar{p}_i$ is a proper prefix of $p.p_1$ or of $p.p_2$, we say that $r[[r']]_{\bar{p}}$ *far-falsifies* $(p_1 \neq p_2)$ at p.

Example 5.5 (*Continues Example 4.17*). Let r_p be a run of $\hat{A}$, defined as $r_p = \hat{\rho}_2(a)$. Then, the pumping $\hat{r}[[r_p]]_{1.1}$ is the run $\hat{r} = \hat{\rho}_4(f(\hat{\rho}_2(a), \hat{\rho}_2(a), \hat{\rho}_2(a)))$, which far-falsifies the disequalities $1.1 \neq 1.3$ and $1.2 \neq 1.3$ at position λ.

In order to give an example of a close-falsified disequality, let $A_{ex2} = \langle Q, \Sigma, F, \Delta\rangle$ be a $\mathrm{TA}_{\neq,hom}$ where $Q = F = \{q\}$ holds, $\Sigma = \{f^{(2)}, h^{(1)}, a^{(0)}\}$ holds, and Δ is the conjunction of the following rules.

—$\rho_1 = a \rightarrow q$.
—$\rho_2 = h(q) \xrightarrow{1.1\neq 1.2} q$.
—$\rho_3 = f(q, q) \xrightarrow{1=2} q$.

Let r_{ex2} be the following run of A_{ex2}: $\rho_3(\rho_2(\rho_2(\rho_1)), \rho_2(\rho_2(\rho_1)))$, with projection $\pi_\Sigma(r_{ex2}) = f(h(h(a)), h(h(a)))$, and let r_p be $\rho_3(\rho_1, \rho_1)$, whose projection is $\pi_\Sigma(r_p) = f(a, a)$. Then, the pumping $r_{ex2}[[r_p]]_{1.1}$ is: $\rho_3(\rho_2(\rho_3(\rho_1, \rho_1)), \rho_2(\rho_3(\rho_1, \rho_1)))$ which is not a run, because it close-falsifies $(1.1 \neq 1.2)$ at $\langle 1, 1.1, 1.1.1\rangle$ (or at $\langle 1, 1.1, 1.1.2\rangle$), and it close-falsifies $(1.1 \neq 1.2)$ at $\langle 2, 2.1, 2.1.1\rangle$ (or at $\langle 2, 2.1, 2.1.2\rangle$).

For the case of emptiness, recall that the starting point is an accepting run r minimum in size. Consider a position $\bar{p}$ of r where $r|_{\bar{p}}$ is a run, and two other subruns r'_1, r'_2 of r smaller than $r|_p$ in size. By the minimality of r, the terms $r[[r'_1]]_{\bar{p}}, r[[r'_2]]_{\bar{p}}$ are not runs. This means that both replacements are falsifying some disequality. Now, consider the case where both replacements are just falsifying the same disequality. In this case, it is possible to prove that the terms $\pi_\Sigma(r'_1), \pi_\Sigma(r'_2)$ are not completelly independent because they share some subterm. This information can be used to bound the number of different subterms in the starting r.

The case of finiteness proceeds similarly. The starting point is an accepting run r that cannot be transformed into a higher one by means of a pumping. Consider a position $\bar{p}$ in the path of maximum length of r and where $r|_{\bar{p}}$ is a run, and consider also two other subruns r'_1, r'_2 pending in the same path, but at positions “much shorter” than $\bar{p}$. In particular, the heights of r'_1, r'_2 are “much bigger” than the one of $r|_p$. By the properties of r, the terms $r[[r'_1]]_{\bar{p}}, r[[r'_2]]_{\bar{p}}$ are not runs. As before, this means that both replacements are falsifying some disequality, and if both replacements are just falsifying the same disequality, it is possible to prove that the terms $\pi_\Sigma(r'_1), \pi_\Sigma(r'_2)$ share some subterm. Analogously to the case of emptiness, this information can be used to bound the height of r.

We analyse why two replacements falsifying the same disequality imply the existence of a common subterm, and which one is this subterm. The following basic lemma will be useful to deal with the case of far-falsified disequalities. It intuitively says that, when a concrete disequality of terms is falsified as a consequence of a simultaneous replacement of occurrences of one term by another new term, the new term is uniquely determined.

LEMMA 5.6. *Let s and t be terms in $\mathcal{T}(\Sigma)$ such that $s \neq t$. Let $P = \{p_1, \ldots, p_n\}$ be a set of positions in $Pos(s)$. Let $P' = \{p'_1, \ldots, p'_k\}$ be a set of positions in $Pos(t)$. Suppose that $s|_{p_1} = \cdots = s|_{p_n} = t|_{p'_1} = \cdots = t|_{p'_k}$. Then, there exists at most one term u satisfying $s[u]_{p_1} \cdots [u]_{p_n} = t[u]_{p'_1} \cdots [u]_{p'_k}$*

PROOF. We prove it by induction on $|s| + |t|$, and distinguishing cases depending on whether some p_i or p'_i is λ or not.

If some p_i or p'_i is λ, say p_1, then n is 1 and since $s \neq t$ and $s|_{p_1}$ coincides with all $t|_{p'_j}$, it follows that no p'_j is λ. Therefore, either k is 0 and hence $t[u]_{p'_1} \cdots [u]_{p'_k}$ is t for any term u, or k is not 0 and hence u is a proper subterm of $t[u]_{p'_1} \cdots [u]_{p'_k}$ for any term u. In the first case, only $u = t$ makes the equality $s[u]_{p_1} \cdots [u]_{p_n} = u = t = t[u]_{p'_1} \cdots [u]_{p'_k}$ true, and we are done. In the second case, no u satisfies the equality $s[u]_{p_1} \cdots [u]_{p_n} = t[u]_{p'_1} \cdots [u]_{p'_k}$, and we are done.

Otherwise, suppose that none of the p_i's and p'_i's is λ. Since s and t are different, either $root(s) \neq root(t)$, or s and t are of the form $f(s_1, \ldots, s_m)$, $f(t_1, \ldots, t_m)$, respectively, and there exists some i in $\{1, \ldots, m\}$ such that s_i and t_i are different. In the first case, it is obvious that no u satisfies $s[u]_{p_1} \cdots [u]_{p_n} = t[u]_{p'_1} \cdots [u]_{p'_k}$, and we are done. Thus, assume that the second case holds for a certain i, and let P_i and P'_i be $\{p \mid i.p \in P\}$ and $\{p \mid i.p \in P'\}$, respectively. The terms s_i, t_i, and sets of positions P_i and P'_i satisfy the conditions of the lemma, and $|s_i| + |t_i| < |s| + |t|$. Hence, induction hypothesis apply for them, and it follows also for s and t that there exists at most one term u such that $s[u]_{p_1} \cdots [u]_{p_n} = t[u]_{p'_1} \cdots [u]_{p'_k}$ holds.

For the case of close-falsified disequalities, we will need an analogous lemma to 5.3.

LEMMA 5.7. *Let $A = \langle Q, \Sigma, F, \Delta \rangle$ be a $TA_{\neq,hom}$. Let r, r' be runs of A, and let $\bar{p}$ be a position such that $r|_{\bar{p}}$ is a run of A and $root(r|_{\bar{p}}) = root(r')$.*

If $r[[r']]_{\bar{p}}$ close-falsifies some disequality, then $r[[r']]_{\bar{p}}$ close-falsifies a disequality $(p_1 \neq p_2)$ at a tuple $\langle p, \bar{p}_i, p.p_1 \rangle$ where p is a prefix of $\bar{p}$.

PROOF. We prove it by induction on $height(r)$. If $\bar{p}$ is λ, then, $r[[r']]_{\bar{p}}$ is r', which is a run, thus contradicting the statement. Hence, without loss of generality, assume that r is of the form $(I : s \xrightarrow{c} q)(s_1, \ldots, s_m)[r_1]_{\hat{p}_1} \cdots [r_n]_{\hat{p}_n}$ and that $\bar{p}$ is of the form $\hat{p}_1.\tilde{p}$.

If $r_1[[r']]_{\tilde{p}}$ close-falsifies some disequality, then, by induction hypothesis, $r_1[[r']]_{\tilde{p}}$ close-falsifies a disequality $(p_1 \neq p_2)$ at a tuple $\langle p', \tilde{p}_i, p'.p_1 \rangle$ where p' is a prefix of $\tilde{p}$. By defining p as $\hat{p}_1.p'$ and $\bar{p}_i$ as $\hat{p}_1.\tilde{p}_i$, the lemma follows.

Otherwise, assume that $r_1[[r']]_{\tilde{p}}$ does not close-falsify any disequality. Since $r[[r']]_{\bar{p}}$ close-falsifies some disequality, it must be a disequality $(p_1 \neq p_2)$ close-falsified at a tuple of the form $\langle \lambda, \bar{p}_i, \lambda.p_1 \rangle$. By defining p as λ, the lemma follows.

Consider the case where pumpings $r[[r'_1]]_{\bar{p}} = r[r'_1]_{\bar{p}_1} \cdots [r'_1]_{\bar{p}_n}$ and $r[[r'_2]]_{\bar{p}} = r[r'_2]_{\bar{p}_1} \cdots [r'_2]_{\bar{p}_n}$ close-falsify the same disequality $(p_1 \neq p_2)$ occurring in a rule applied at a certain position p. Also, consider the particular case where a position $\bar{p}_i$ is a proper prefix of $p.p_1$, and a position $\bar{p}_j$ is a proper prefix of $p.p_2$. We have that the terms $\pi_\Sigma(r)|_{p.p_1}, \pi_\Sigma(r)|_{p.p_2}$ are different, but the terms $\pi_\Sigma(r[[r'_1]]_{\bar{p}})|_{p.p_1}, \pi_\Sigma(r[[r'_1]]_{\bar{p}})|_{p.p_2}$ are identical, and the terms $\pi_\Sigma(r[[r'_2]]_{\bar{p}})|_{p.p_1}, \pi_\Sigma(r[[r'_2]]_{\bar{p}})|_{p.p_2}$ are identical. This is the same as saying that the terms $\pi_\Sigma(r|_{\bar{p}})|_{p.p_1-\bar{p}_i}, \pi_\Sigma(r|_{\bar{p}})|_{p.p_2-\bar{p}_j}$ are different, but the terms $\pi_\Sigma(r'_1|_{p.p_1-\bar{p}_i}), \pi_\Sigma(r'_1|_{p.p_2-\bar{p}_j})$ are identical, and the terms $\pi_\Sigma(r'_2|_{p.p_1-\bar{p}_i}), \pi_\Sigma(r'_2|_{p.p_2-\bar{p}_j})$ are identical. Note that the positions $p.p_1 - \bar{p}_i$ and $p.p_2 - \bar{p}_j$ are positions in $Suff_{\neq}(A)$. Thus, we are in a situation where a subrun $r|_{\bar{p}}$ is replaced by subruns r'_1, r'_2 having different (from $r|_{\bar{p}}$) equality and disequality relations among the subterms at positions in $Suff_{\neq}(A)$. In this case, we are not able to infer that r'_1 and r'_2 share some subterm. For this reason, we will avoid these situations by just applying pumpings $r[[r']]_{\bar{p}}$ where $r|_p$ and r' satisfy the same equality and disequality relations among the positions in $Suff_{\neq}(A)$.

Definition 5.8. Let $A = \langle Q, \Sigma, F, \Delta \rangle$ be a $\mathrm{TA}_{\neq,hom}$. Let s and t be terms in $\mathcal{T}(\Sigma)$. By $s \sim^{Suff_{\neq}(A)} t$, or just $s \sim t$ when A is clear from the context, we denote that for each two positions $p_1, p_2 \in Suff_{\neq}(A)$, $(s|_{p_1} = s|_{p_2} \Leftrightarrow t|_{p_1} = t|_{p_2})$ holds.

Note that $\sim$ is an equivalence relation with a finite number of classes bounded by $(2Nh)^{2Nh}$ (an upper bound on the number of partitions of the at most $2Nh$ elements of $Suff_{\neq}(A)$).

The following lemma establishes which subterms are identical when two pumpings falsify the same disequality. Note that equivalence with respect to $\sim$ is one of the necessary assumptions in the case of close-falsified disequalities.

LEMMA 5.9. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq,hom}$. Let r, r_1, r_2 be runs of A, and let $\bar{p}$ be a position such that $r|_{\bar{p}}$ is a run of A and $root(r|_{\bar{p}}) = root(r_1) = root(r_2)$. Suppose that $r[[r_1]]_{\bar{p}}$ and $r[[r_2]]_{\bar{p}}$ are not runs. Let $\bar{p}_1, \ldots, \bar{p}_k$ be the replaced positions of both $r[[r_1]]_{\bar{p}}$ and $r[[r_2]]_{\bar{p}}$.*

— *If both $r[[r_1]]_{\bar{p}}$ and $r[[r_2]]_{\bar{p}}$ far-falsify the same $(p_1 \neq p_2)$ at the same p, then, $\pi_\Sigma(r_1) = \pi_\Sigma(r_2)$.*
— *If $\pi_\Sigma(r|_{\bar{p}}) \sim \pi_\Sigma(r_1) \sim \pi_\Sigma(r_2)$ and both $r[[r_1]]_{\bar{p}}$ and $r[[r_2]]_{\bar{p}}$ close-falsify the same $(p_1 \neq p_2)$ at the same $\langle p, \bar{p}_i, p.p_1\rangle$, then $\pi_\Sigma(r)|_{\bar{p}.((p.p_1)-\bar{p}_i)} \neq \pi_\Sigma(r_1)|_{(p.p_1)-\bar{p}_i} = \pi_\Sigma(r_2)|_{(p.p_1)-\bar{p}_i}$.*

PROOF. The first item is a direct consequence of previous definitions and Lemma 5.6.

For the second item, we first observe that $\pi_\Sigma(r)|_{\bar{p}.((p.p_1)-\bar{p}_i)} = \pi_\Sigma(r|_{\bar{p}})|_{(p.p_1)-\bar{p}_i} = \pi_\Sigma(r|_{\bar{p}_i})|_{(p.p_1)-\bar{p}_i} = \pi_\Sigma(r)|_{\bar{p}_i.((p.p_1)-\bar{p}_i)} = \pi_\Sigma(r)|_{p.p_1} \neq \pi_\Sigma(r)|_{p.p_2}$ holds.

Second, we prove that no position in $\{\bar{p}_1, \ldots, \bar{p}_k\}$ can be a proper prefix of $p.p_2$ by contradiction. Thus, assume that, for some position in $\{\bar{p}_1, \ldots, \bar{p}_k\}$, say $\bar{p}_j$, $\bar{p}_j$ is a proper prefix of $p.p_2$. Note that $r|_{\bar{p}} = r|_{\bar{p}_i} = r|_{\bar{p}_j}$ holds. Thus, $\pi_\Sigma(r)|_{p.p_2} = \pi_\Sigma(r)|_{\bar{p}_j.((p.p_2)-\bar{p}_j)} = \pi_\Sigma(r|_{\bar{p}_j})|_{((p.p_2)-\bar{p}_j)} = \pi_\Sigma(r|_{\bar{p}})|_{((p.p_2)-\bar{p}_j)} = \pi_\Sigma(r)|_{\bar{p}.((p.p_2)-\bar{p}_j)}$ holds, and hence, $\pi_\Sigma(r|_{\bar{p}})|_{(p.p_1)-\bar{p}_i} \neq \pi_\Sigma(r|_{\bar{p}})|_{((p.p_2)-\bar{p}_j)}$ follows from our first observation. Since $\pi_\Sigma(r|_{\bar{p}}) \sim \pi_\Sigma(r_1)$ holds, then $\pi_\Sigma(r_1)|_{(p.p_1)-\bar{p}_i} \neq \pi_\Sigma(r_1)|_{(p.p_2)-\bar{p}_j}$ holds. But this is in contradiction with the fact that $r[[r_1]]_{\bar{p}}$ close-falsifies $(p_1 \neq p_2)$ at $\langle p, \bar{p}_i, p.p_1\rangle$.

Third, we show that $p.p_2$ is not a prefix of any position in $\{\bar{p}_1, \ldots, \bar{p}_k\}$, again by contradiction. Thus, assume that, for some position in $\{\bar{p}_1, \ldots, \bar{p}_k\}$, say $\bar{p}_j$, $p.p_2$ is a prefix of $\bar{p}_j$. Then, r_1 is a subterm of $r[[r_1]]_{\bar{p}}|_{p.p_2}$. On the other side, since $\bar{p}_i$ is a proper prefix of $p.p_1$, it holds that $r[[r_1]]_{\bar{p}}|_{p.p_1}$ is a proper subterm of r_1. Thus, $r[[r_1]]_{\bar{p}}|_{p.p_1} = r[[r_1]]_{\bar{p}}|_{p.p_2}$ is not possible, contradicting the fact that $r[[r_1]]_{\bar{p}}$ close-falsifies $(p_1 \neq p_2)$ at $\langle p, \bar{p}_i, p.p_1\rangle$.

From the two previous facts, we conclude that $p.p_2$ is parallel with all replaced positions $\bar{p}_1, \ldots, \bar{p}_n$. Thus, $r|_{p.p_2} = r[[r_1]]_{\bar{p}}|_{p.p_2} = r[[r_2]]_{\bar{p}}|_{p.p_2}$ holds. Moreover, recall from the first observation that $\pi_\Sigma(r)|_{\bar{p}.((p.p_1)-\bar{p}_i)} \neq \pi_\Sigma(r)|_{p.p_2} = \pi_\Sigma(r|_{p.p_2})$ holds. Hence, $\pi_\Sigma(r)|_{\bar{p}.((p.p_1)-\bar{p}_i)} \neq \pi_\Sigma(r[[r_1]]_{\bar{p}}|_{p.p_2}) = \pi_\Sigma(r[[r_2]]_{\bar{p}}|_{p.p_2})$. Note also that $\pi_\Sigma(r_1)|_{(p.p_1)-\bar{p}_i} = \pi_\Sigma(r[[r_1]]_{\bar{p}})|_{p.p_1} = \pi_\Sigma(r[[r_1]]_{\bar{p}})|_{p.p_2} = \pi_\Sigma(r[[r_1]]_{\bar{p}}|_{p.p_2})$, and $\pi_\Sigma(r_2)|_{(p.p_1)-\bar{p}_i} = \pi_\Sigma(r[[r_2]]_{\bar{p}})|_{p.p_1} = \pi_\Sigma(r[[r_2]]_{\bar{p}})|_{p.p_2} = \pi_\Sigma(r[[r_2]]_{\bar{p}}|_{p.p_2})$ hold. Therefore, $\pi_\Sigma(r)|_{\bar{p}.((p.p_1)-\bar{p}_i)} \neq \pi_\Sigma(r_1)|_{(p.p_1)-\bar{p}_i} = \pi_\Sigma(r_2)|_{(p.p_1)-\bar{p}_i}$ also hold, and we are done.

5.1. Decreasing Pumpings

When an accepting run is big enough, it can be argued the existence of a pumping decreasing the size and producing a correct run. This allows us to prove decidability of emptiness. The argumentation of this fact follows the ideas presented in Dauchet et al. [1995] for proving decidability of emptiness for reduction automata. In the following proof, we assume a given well-founded ordering $\gg$, total on terms constructed over the alphabet of runs of A, fulfilling the strict size relation (if $|r| > |r'|$, then $r \gg r'$) and monotonic ($r_1 \gg r_2$ implies $r[r_1] \gg r[r_2]$ for any r).

The proof schema has been outlined at the beginning of this section. We consider an accepting run r minimal with respect to $\gg$. As a first step, we note that in this minimal run, two subruns with the same rule applied at their roots and whose corresponding subterms are identical, are also identical as runs. Next, by applying several pumpings replacing subruns by other smaller subruns, we infer the coincidence of several subterms. The conclusion is a bound for the number of different subruns in r. The proof is constructive. We keep pairs $\langle E_i, P_i\rangle$, where E_i is a set of positions of r identifying different runs, and P_i is a set of relative positions of such subruns with the following property: whenever a position p belongs to P_i, all the terms of the runs identified by E_i must have the same subterm at position p. Thus, the more elements has P_i, the more similar are the terms of the runs identified by E_i. In the constructive process, the sets P_i grow up until forcing the corresponding E_i to have runs of the same term. Also, this allows us to conclude that each E_i has just one run.

LEMMA 5.10. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq,hom}$. Let r be an accepting run of A, which is minimal with respect to $\gg$ among all the accepting runs of A.*
Then, the number of distinct subterms of r, that is, $|\{r|_p \mid p \in Pos(r)\}|$, is bounded by $N_{Pos}(N+1)^{|\Delta|\cdot(2Nh)^{(4Nh)}}$.

PROOF. We start by observing that two subruns $r|_{p_1}, r|_{p_2}$ of r with the same rule at the root and projected to the same term by π_Σ, that is, $\pi_\Sigma(r|_{p_1}) = \pi_\Sigma(r|_{p_2})$ and $root(r|_{p_1}) = root(r|_{p_2})$, must be identical: otherwise, both pumpings $r[[r|_{p_1}]]_{p_2}, r[[r|_{p_2}]]_{p_1}$ are runs, because they do not falsify any disequality, both are accepting, and one of them is smaller than r with respect to $\gg$ (this depends on which one of $r|_{p_1} \gg r|_{p_2}$ or $r|_{p_2} \gg r|_{p_1}$ hold).

Now, let P be a maximal set of positions identifying different subruns of r, that is, P is a set of positions included in $\{p \mid r|_p \text{ is a run}\}$ such that $\{r|_p \mid r|_p \text{ is a run}\} = \{r|_p \mid p \in P\}$ and $(p_1, p_2 \in P \wedge p_1 \neq p_2 \Rightarrow r|_{p_1} \neq r|_{p_2})$ hold. In order to conclude, it suffices to bound $|P|$ by $(N+1)^{|\Delta|\cdot(2Nh)^{(4Nh)}}$. We will apply a conceptual process dealing with a data structure of the form $\langle S, \{\langle E_1, P_1\rangle, \ldots, \langle E_k, P_k\rangle\}\rangle$, where all $S, E_1, P_1, \ldots, E_k, P_k$ are sets of positions satisfying the following invariants.

—$S \cup E_1 \cup \cdots \cup E_k$ equals P.
—The sets $S, E_1, \ldots, E_k$ are pairwise disjoint.
—For each i in $\{1, \ldots, k\}$ and each two positions $\bar{p}, p \in E_i$, $s|_{\bar{p}} \sim s|_p$ and $root(r|_{\bar{p}}) = root(r|_p)$ hold.
—$P_1, \ldots, P_k$ contain positions that are suffixes of positions occurring in disequality atoms of constraints of rules in Δ.
—For each i in $1, \ldots, k$, and each $\bar{p}, p \in E_i$, and each $\check{p}$ in P_i, $s|_{\bar{p}.\check{p}} = s|_{p.\check{p}}$ holds.

By our first observation in this proof, different subruns with the same rule applied at their roots have different corresponding subterms. Thus, for any two different positions p_1, p_2 in a concrete E_i, the terms $\pi_\Sigma(r|_{p_1})$ and $\pi_\Sigma(r|_{p_2})$ are different. Hence, according to the first item of Lemma 5.9, they cannot far-falsify the same disequality when they are used for a pumping at the same position in r. Moreover, if a concrete P_i coincides with $Suff_{\neq}(A)$, then, the corresponding E_i contains at most one element: suppose the opposite, that is, that there are two different positions p_1, p_2 in E_i. Then, since $Suff_{\neq}(A)$ contains λ, $\pi_\Sigma(r|_{p_1}) = \pi_\Sigma(r|_{p_2})$ holds, and hence $r|_{p_1} = r|_{p_2}$ holds, which is a contradiction with the fact that $p_1, p_2 \in P$.

We describe the process as follows.

—*Starting the Process.* The first tuple of our process will be $\langle \emptyset, \{\langle E_1, \emptyset\rangle, \ldots, \langle E_k, \emptyset\rangle\}\rangle$, where $\{E_1, \ldots, E_k\}$ is the partition satisfying that $E_1 \cup \cdots \cup E_k$ equals P, and two positions $\bar{p}, p$ are in the same E_i if and only if $(s|_{\bar{p}} \sim s|_p \wedge root(r|_{\bar{p}}) = root(r|_p))$ holds. It is clear that the first tuple satisfies the invariants.

—*A Step of the Process.* Now, at each step of the process with a current tuple $\langle S, \{\langle E_1, P_1\rangle, \dots, \langle E_k, P_k\rangle\}\rangle$, it is chosen the position $\bar{p}$ of $E_1 \cup \dots \cup E_k$ such that $r|_{\bar{p}}$ is maximum with respect to $\gg$. Note that such a position is uniquely determined by the totality of $\gg$. Without loss of generality, suppose that $\bar{p}$ is in E_1. Consider any p in $E_1 - \{\bar{p}\}$. By the selection of $\bar{p}$ and the monotonicity of $\gg$, $r[[r|_p]]_{\bar{p}}$ is smaller than r. Thus, by the minimality of r, such a term $r[[r|_p]]_{\bar{p}}$ is not a run. Hence, by Lemma 5.3, $r[[r|_p]]_{\bar{p}}$ falsifies some disequality $(p_1 \neq p_2)$ at some p' which is a prefix of $\bar{p}$.

Now, the process considers the set S' of all positions p in $E_1 - \{\bar{p}\}$ such that $r[[r|_p]]_{\bar{p}}$ far-falsifies some disequality at some prefix of $\bar{p}$. Recall that, for two positions $p_1, p_2 \in E_1$, the pumpings $r[[r|_{p_1}]]_{\bar{p}}, r[[r|_{p_2}]]_{\bar{p}}$ cannot far-falsify the same disequality. Therefore, $|S'|$ is bounded by $N|\bar{p}|$. Moreover, each prefix p' of $\bar{p}$ satisfies $r|_{p'} \gg r|_{\bar{p}}$, and all such $r|_{p'}$'s are different. Thus, for each of such p', there is a position in S from which the same run pends, and it follows that $|p|$ is bounded by $|S|$. Hence, $|S'| \leq N|S|$ holds.

Now, note that, for each p in $E_1 - (\{\bar{p}\} \cup S')$, $r[[r|_p]]_{\bar{p}}$ close-falsifies some disequality $(p_1 \neq p_2)$ at some $\langle p', \bar{p}', p'.p_1\rangle$, where p' is a prefix of $\bar{p}$. Note also that, by the last part of the statement in Lemma 5.2, such a p' can be chosen only among $h(A)$ possibilities. The process constructs a partition $\{E'_1, \dots, E'_n\}$ of $E_1 - (\{\bar{p}\} \cup S')$ and satisfying the following condition: for each i in $\{1, \dots, n\}$, there exists a position $\breve{p}_i$, a disequality $(p_1 \neq p_2)$, and a tuple $\langle p', \bar{p}', p'.p_1\rangle$, such that, for each position p in E'_i, the pumping $r[[r|_p]]_{\bar{p}}$ close-falsifies $(p_1 \neq p_2)$ at $\langle p', \bar{p}', p'.p_1\rangle$, p' is a prefix of $\bar{p}$ and $p'.p_1 - \bar{p}'$ equals $\breve{p}_i$ (note that there could be several different selections for the partition $\{E'_1, \dots, E'_n\}$ if some $r[[r|_p]]_{\bar{p}}$ close-falsifies several disequalities). For each of such i's, disequalities $(p_1 \neq p_2)$, and tuples $\langle p', \bar{p}', p'.p_1\rangle$, by the second item of Lemma 5.9, $\pi_\Sigma(r)|_{\bar{p}.((p'.p_1)-\bar{p}')}$ is different from $\pi_\Sigma(r|_p)|_{(p'.p_1)-\bar{p}'}$ for each p in E'_i, and all of such $\pi_\Sigma(r|_p)|_{(p'.p_1)-\bar{p}'}$ are identical. It follows $\pi_\Sigma(r)|_{\bar{p}.\breve{p}_i} \neq \pi_\Sigma(r)|_{p.\breve{p}_i}$ for each p in E'_i, and all $\pi_\Sigma(r)|_{p.\breve{p}_i}$ are identical for all p in E'_i. Thus, by the invariants of the process, $\breve{p}_i$ is not in P_1 (note that this consequence is valid for $\breve{p}_1, \dots, \breve{p}_n$). The tuple constructed for the next step is $\langle S \cup \{\bar{p}\} \cup S', \{\langle E'_1, P_1 \cup \{\breve{p}_1\}\rangle, \dots, \langle E'_n, P_1 \cup \{\breve{p}_n\}\rangle, \langle E_2, P_2\rangle, \dots, \langle E_k, P_k\rangle\}\rangle$. From these observations, it follows that the new tuple satisfies the invariants, too.

Recall that, at each step, $\langle E_1, P_1\rangle$ is removed and new $\langle E'_1, P_1 \cup \{\breve{p}_1\}\rangle, \dots, \langle E'_n, P_1 \cup \{\breve{p}_n\}\rangle$ are added, where each of such $P_1 \cup \{\breve{p}_i\}$ satisfies that $\breve{p}_i$ is not in P_1. Also recall that, by the invariants, the sets P_j contain suffixes of positions occurring at disequalities in the constraints of rules in Δ. There are at most $2Nh$ different suffixes of this kind. Moreover, by the last part of the statement in Lemma 5.2, the triples $\langle p', \bar{p}', p'.p_1\rangle$ are constructed by choosing p' among h possibilities, and p_1 among $2N$ possibilities, from which $\bar{p}'$ is uniquely determined. Hence, there are at most $2Nh$ possible triples. Thus, when $\langle E_1, P_1\rangle$ is replaced by $\langle E'_1, P_1 \cup \{\breve{p}_1\}\rangle, \dots, \langle E'_n, P_1 \cup \{\breve{p}_n\}\rangle$, such an n is bounded by $2Nh$. Finally, recall that the number of equivalent classes of $\sim$ is bounded by $(2Nh)^{(2Nh)}$. Therefore, the number of execution steps of the process is bounded by $M = |\Delta| \cdot (2Nh)^{(4Nh)}$. It follows that the process terminates, and S contains P when it halts. Let S_i represent the set S at the ith execution step. By these remarks $|S_{i+1}| \leq |S_i| + N|S_i| + 1$. Since $|S_0| = 0$, we can bound $|S_i|$ by $(N+1)^i$, and hence, $|P|$ is bounded by $|S_M| \leq (N+1)^M$, and this concludes the proof.

COROLLARY 5.11. *The emptiness problem is decidable for $\mathrm{TA}_{\neq,hom}$.*

5.2. Increasing Pumpings

In this section, we prove that finiteness is decidable for $\mathrm{TA}_{\neq,hom}$. To this end, we define a computational bound $\check{h}$ ~~on~~ from the size of a given $\mathrm{TA}_{\neq,hom}$ A, and prove that any accepting run of A with height greater than $\check{h}$ can be tranformed into a higher accepting run by means of a pumping. As a consequence, the language recognized by A is infinite if and only if there exists an accepting run with height greater than $\check{h}$. The particular constraints of the pumping we define will be useful also for the decision prodedure of the HOM problem in Section 7.

Definition 5.12. Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $\mathrm{TA}_{\neq,hom}$. We define $\check{h}(A)$ as $|\Delta| \cdot (h^2 + h) \cdot ((2Nh)^{4Nh})$. We write $\check{h}$ when A is clear from the context.

As a first step, in Lemma 5.13 we consider a run with height greater than $\check{h}$, and prove that there exists a pumping $r[[r|_p]]_{\bar{p}}$ which does not close-falsify any disequality. Moreover, we prove that the two involved positions $p, \bar{p}$ can be chosen to satisfy some additional constraints: p is a prefix of $\bar{p}$ and $|\bar{p}| - |p| > h^2$. These additional restrictions will be used in further lemmas for proving that, in fact, the pumping does not far-falsify any disequality, and that the obtained $r[[r|_p]]_{\bar{p}}$ is higher than r. This will allow us to conclude that the obtained $r[[r|_p]]_{\bar{p}}$ is a higher run than r.

The proof of Lemma 5.13 is somehow similar to the proof of Lemma 5.10. We start by assuming that any of such pumpings close-falsify some disequality, and proceed to reach a contradiction. We keep a pair $\langle E, P\rangle$, where E keeps positions of runs in the longest path of r, and P is a set of positions in $\mathit{Suff}_{\neq}(A)$ saying how similar are the runs referred by E: if a position p belongs to P and the positions p_1, p_2 belong to E, then the terms $\pi_\Sigma(r|_{p_1.p}), \pi_\Sigma(r|_{p_2.p})$ are identical. The set P grows up step by step while the set E decreases by a constant factor. The number of steps is bounded because $|P|$ is always bounded by $|\mathit{Suff}_{\neq}(A)|$. Thus, the size of the initial E can be also bounded. We reach a contradiction by concluding that the size of r was bounded by $\check{h}$.

LEMMA 5.13. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq,hom}$. Let r be a run of A satisfying $height(r) > \check{h}(A)$. Let $\check{p}$ be a position in $Pos(r)$ satisfying $|\check{p}| = height(r)$. Then, there are two positions $p, \bar{p}$ of $Pos(r)$ satisfying the following conditions.*

—*$p < \bar{p} \leq \check{p}$ and $|p| + h^2 < |\bar{p}|$.*
—*$r|_p$ and $r|_{\bar{p}}$ are runs satisfying $root(r|_p) = root(r|_{\bar{p}})$.*
—*$r[[r|_p]]_{\bar{p}}$ does not close-falsify any disequality.*

PROOF. We proceed by contradiction by assuming that such two positions do not exist. Thus, for any two positions $p, \bar{p}$ satisfying $p < \bar{p} \leq \check{p}$, $|p| + h^2 < |\bar{p}|$, $r|_p$ and $r|_{\bar{p}}$ are runs, and $root(r|_p) = root(r|_{\bar{p}})$, it holds that the pumping $r[[r|_p]]_{\bar{p}}$ close-falsifies some disequality.

Since $|\check{p}| = height(r)$ holds, in particular, $|\check{p}| > \check{h}$ holds. From the prefixes of $\check{p}$, we can choose a set E with $|E| \geq \check{h}/((2Nh)^{2Nh} \cdot |\Delta| \cdot (h^2 + h)) = (2Nh)^{(2Nh)}$ and satisfying the following conditions.

—For each position p in E, $r|_p$ is a run.
—For each two positions $p, \bar{p}$ in E, $\pi_\Sigma(r|_p) \sim \pi_\Sigma(r|_{\bar{p}})$ and $root(r|_p) = root(r|_{\bar{p}})$ hold.
—For each two positions $p, \bar{p}$ in E, satisfying $p < \bar{p}$, $|p| + h^2 < |\bar{p}|$ holds.

By our assumptions, for any two positions $p < \bar{p}$ in E, the pumping $r[[r|_p]]_{\bar{p}}$ close-falsifies some disequality.

We will proceed by modifying E and a set of positions P as follows by preserving these conditions, and also by preserving the following invariant: P contains suffixes of positions occurring at disequalities in the constraints of rules in Δ, and for each position $\check{p}$ in P and each two positions $p < \bar{p}$ in E, $\pi_\Sigma(r|_{p.\check{p}}) = \pi_\Sigma(r|_{\bar{p}.\check{p}})$ holds.

Initially, P is $\emptyset$ and E is defined as described here. At each step, we consider the maximum position $\bar{p}$ of E in size. Note that, for each position $p < \bar{p}$ in E, the pumping $r[[r|_p]]_{\bar{p}}$ close-falsifies some disequality $(p_1 \neq p_2)$ at some $\langle p', \bar{p}', p'.p_1\rangle$. Moreover, by

Lemma 5.7, for each of such p we can assume that the corresponding p' is a prefix of $\bar{p}$. Note also that, by the last part of the statement in Lemma 5.2, such a p' can be chosen only among h possibilities.

Let $\{E_1, \ldots, E_n\}$ be a partition of $E - \{\bar{p}\}$ satisfying the following condition: for each i in $\{1, \ldots, n\}$, there exists a position $\check{p}_i$, a disequality $(p_1 \neq p_2)$, and a tuple $\langle p', \bar{p}', p'.p_1\rangle$, such that, for each position p in E_i, the pumping $r[[r|_p]]_{\bar{p}}$ close-falsifies $(p_1 \neq p_2)$ at $\langle p', \bar{p}', p'.p_1\rangle$, p' is a prefix of $\bar{p}$ and $p'.p_1 - \bar{p}'$ equals $\check{p}_i$ (note that there could be several different selections for the partition $\{E_1, \ldots, E_n\}$ if some $r[[r|_p]]_{\bar{p}}$ close-falsifies several disequalities). For each of such i's, disequalities $(p_1 \neq p_2)$, and tuples $\langle p', \bar{p}', p'.p_1\rangle$, by the second item of Lemma 5.9, $\pi_\Sigma(r)|_{\bar{p}.((p'.p_1)-\bar{p}')}$ is different from $\pi_\Sigma(r|_p)|_{(p'.p_1)-\bar{p}'}$ for each p in E_i, and all of such $\pi_\Sigma(r|_p)|_{(p'.p_1)-\bar{p}'}$ are identical. It follows $\pi_\Sigma(r)|_{\bar{p}.\check{p}_i} \neq \pi_\Sigma(r)|_{p.\check{p}_i}$ for each p in E_i, and all $\pi_\Sigma(r)|_{p.\check{p}_i}$ are identical for all p in E_i. Thus, by the invariant, $\check{p}_i$ is not in P (note that this consequence is valid for $\check{p}_1, \ldots, \check{p}_n$). For the next step, we choose E as the E_i with maximum cardinality $|E_i|$, and choose P as $P \cup \{\check{p}_i\}$. From these observations, it follows that the new E, P satisfy the invariants, too.

Note that, at each step, P increases its cardinality by 1. Also recall that the set P contains suffixes of positions occurring at disequalities in the constraints of rules in Δ. There are at most $2Nh$ different suffixes of this kind. It follows that the number of execution steps is bounded by $2Nh$. Moreover, by the last part of the statement in Lemma 5.2, the triples $\langle p', \bar{p}', p'.p_1\rangle$ are constructed by choosing p' among h possibilities, and p_1 among $2N$ possibilities, from which $\bar{p}'$ is uniquely determined. Hence, there are at most $2Nh$ possible triples. Thus, the partition $E_1, \ldots, E_n$ has at most $2Nh$ parts. Hence, the cardinal of E is subtracted by 1 and then divided by at most $2Nh$ at each execution step, that is, $|E_{j+1}| \geq \lceil (|E_j| - 1)/(2Nh)\rceil$. Note that, according to the assumptions, E must be empty at the last execution step. Thus, the starting E satisfies $|E| < ((2Nh)^{(2Nh)})$, and this is in contradiction with $|E| \geq (2Nh)^{(2Nh)}$.

We prove in Lemma 5.17 that the pumping given by Lemma 5.13 does not far-falsify any disequality, either. Thus, it produces a run. We anticipate here the schema of that proof in order to give meaning to the following intermediate lemmas. The proof proceeds by contradiction, by assuming that the pumping far-falsifies a disequality $(p_1 \neq p_2)$ at a position p'. Thus, $\pi_\Sigma(r|_{p'.p_1}) \neq \pi_\Sigma(r|_{p'.p_2})$ and $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1}) = \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2})$ hold. The positions $p'.p_1$ and $p'.p_2$ can be enlarged in some ways to positions $p'.p_1.p''$, $p'.p_2.p''$ still satisfying $\pi_\Sigma(r|_{p'.p_1.p''}) \neq \pi_\Sigma(r|_{p'.p_2.p''})$. It suffices to follow one of the paths that make both terms different. We follow this path until we find a rule application at one of the positions. This way, the position p'' ensures that the number of rule applications in r at positions between p' and $p'.p_1.p''$ is the same as at positions between p' and $p'.p_1$, and thus bounded by h. Similarly, the number of rule applications in r at positions beween p' and $p'.p_2.p''$ is also bounded by h. In order to see why this is useful, suppose that $p'.p_1.p''$ is a prefix of a replaced position. It follows that the height of $r|_{p'.p_1.p''}$ is at most h^2 units smaller than the height of $r|_{p'}$. Since $|p| + h^2 < |\bar{p}|$ holds, we conclude that the height of $r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}$ is bigger than the height of $r|_{p'}$ (this is proved with precision by Lemma 5.15 and Corollary 5.16). This bound is important in the case where $p'.p_1.p''$ is a prefix of some replaced position, and $p'.p_2.p''$ is not. In this case, $r|_{p'.p_2.p''}$ coincides with $r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''}$, and its height is bounded by the height of $r|_{p'}$. Thus, the height of $r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''}$ is smaller than the height of $r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}$, and this is in contradiction with $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1}) = \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2})$. Another difficult case is when both $p'.p_1.p''$ and $p'.p_2.p''$ are prefixes of replaced positions. Lemma 5.14 deals with this case by proving that then one of $r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}$ or $r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''}$ is a proper subterm of the other, thus contradicting $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1}) = \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2})$ again.

LEMMA 5.14. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq,hom}$. Let r, r' be runs of A, and let $\bar{p}$ be a position such that $root(r|_{\bar{p}}) = root(r')$. Let p_1, p_2 be two positions such that $r|_{p_1}$ is a* *run, $r|_{p_1} \neq r|_{p_2}$, and both p_1, p_2 are prefixes of replaced positions in $r[[r']]_{\bar{p}}$. Then, either $r[[r']]_{\bar{p}}|_{p_1}$ is a proper subterm of $r[[r']]_{\bar{p}}|_{p_2}$, or $r[[r']]_{\bar{p}}|_{p_2}$ is a proper subterm of $r[[r']]_{\bar{p}}|_{p_1}$.*

PROOF. We prove it by induction on $|\bar{p}|$. We write r more explicitly as $(I : s \xrightarrow{c} q)(s_1, \ldots, s_m)[r_1]_{\hat{p}_1} \cdots [r_n]_{\hat{p}_n}$. If p_1 or p_2 is λ, the result follows trivially. Thus, assume that none of them is λ. If p_2 is a position in $Pos_\Sigma(s)$, then, since both p_1 and p_2 are prefixes of replaced positions and $r|_{p_1}$ is a run, it follows that p_1 is of the form $\hat{p}_i.p'_1$, and p_2 is a proper prefix of a certain $\hat{p}_j$ satisfying that $(\hat{p}_i = \hat{p}_j)$ occurs in c. The result follows trivially by the definition of pumping. Thus, assume that p_2 is not in $Pos_\Sigma(s)$. In this case, p_1 is of the form $\hat{p}_i.p'_1$, and p_2 is of the form $\hat{p}_j.p'_2$ where $(\hat{p}_i = \hat{p}_j)$ occurs in c. Let $\hat{p}_k$ be the prefix of $\bar{p}$ among the positions $\hat{p}_1, \ldots, \hat{p}_n$. Note that either $\hat{p}_i$ is $\hat{p}_k$ or $(\hat{p}_i = \hat{p}_k)$ occurs in c, and either $\hat{p}_j$ is $\hat{p}_k$ or $(\hat{p}_j = \hat{p}_k)$ occurs in c. Now, observe that the runs $r|_{\hat{p}_k}, r'$, the position $\bar{p} - \hat{p}_k$, and the two positions p'_1 and p'_2 satisfy the assumptions of the lemma and $|\bar{p} - \hat{p}_k| < |\bar{p}|$ holds. Therefore, by induction hypothesis, either $(r|_{\hat{p}_k})[[r']]_{\bar{p}-\hat{p}_k}|_{p'_1}$ is a proper subterm of $(r|_{\hat{p}_k})[[r']]_{\bar{p}-\hat{p}_k}|_{p'_2}$, or $(r|_{\hat{p}_k})[[r']]_{\bar{p}-\hat{p}_k}|_{p'_2}$ is a proper subterm of $(r|_{\hat{p}_k})[[r']]_{\bar{p}-\hat{p}_k}|_{p'_1}$. Finally, since either $\hat{p}_i$ is $\hat{p}_k$ or $(\hat{p}_i = \hat{p}_k)$ occurs in c, and either $\hat{p}_j$ is $\hat{p}_k$ or $(\hat{p}_j = \hat{p}_k)$ occurs in c, then $(r|_{\hat{p}_k})[[r']]_{\bar{p}-\hat{p}_k}|_{p'_1} = r[[r']]_{\bar{p}}|_{p_1}$ and $(r|_{\hat{p}_k})[[r']]_{\bar{p}-\hat{p}_k}|_{p'_2} = r[[r']]_{\bar{p}}|_{p_2}$ hold, and hence, the result follows.

LEMMA 5.15. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq,hom}$. Let r, r' be runs of A, and let $\bar{p}$ be a position such that $root(r|_{\bar{p}}) = root(r')$. Let p be a prefix of $\bar{p}$ such that $r|_p$ is a run and $height(r|_p) = |(\bar{p} - p)| + height(r|_{\bar{p}})$. Let p_1 be a position such that $p.p_1$ is a prefix of a replaced position in $r[[r']]_{\bar{p}}$. Let k be $|\{p' \mid p \leq p' < p.p_1 \wedge r|_{p'}$ is a run$\}|$.*
Then, $height(r[[r']]_{\bar{p}}|_{p.p_1}) \geq |\bar{p} - p| + height(r') - k \cdot h(A)$.

PROOF. We can assume that p is λ: otherwise, by replacing r by $r|_p$, $\bar{p}$ by $\bar{p} - p$ and p by λ we force this assumption. We prove the lemma by induction on k. If $r|_{p_1}$ is not a run, then, since p_1 is a prefix of a replaced position, p_1 can be enlarged to satisfy that $r|_{p_1}$ is a run, by preserving the rest of assumptions and the value for k. Thus, assume that $r|_{p_1}$ is a run. Assume that p_1 is λ. Then k is 0, $height(r[[r']]_{\bar{p}}|_{p_1}) = height(r[[r']]_{\bar{p}}) \geq |\bar{p}| + height(r')$ holds, and hence, $height(r[[r']]_{\bar{p}}|_{p_1}) \geq |\bar{p}| + height(r') - k \cdot h$ follows. Otherwise, assume that p_1 is not λ. In such a case, since p_1 is a prefix of a replaced position, $\bar{p}$ cannot be λ. We write r more explicitly as $(I : s \xrightarrow{c} q)(s_1, \ldots, s_m)[r_1]_{\hat{p}_1} \cdots [r_n]_{\hat{p}_n}$. Let $\hat{p}_j$ be the prefix of $\bar{p}$ among the positions $\hat{p}_1, \ldots, \hat{p}_n$. Since $r|_{p_1}$ is a run, p_1 is of the form $\hat{p}_i.p'_1$ for some $\hat{p}_i$ such that either $\hat{p}_i$ is $\hat{p}_j$, or $(\hat{p}_i = \hat{p}_j)$ occurs in c. Now, note that the runs $r|_{\hat{p}_j}, r'$, the position $\bar{p} - \hat{p}_j$, and the positions λ and p'_1 satisfy the assumptions of the lemma for $k-1$. Thus, by induction hypothesis, $height(r|_{\hat{p}_j}[[r']]_{\bar{p}-\hat{p}_j}|_{p'_1}) \geq |\bar{p} - \hat{p}_j| + height(r') - (k-1) \cdot h$ holds. Since either $\hat{p}_i$ is $\hat{p}_j$ or $(\hat{p}_i = \hat{p}_j)$ occurs in c, it follows that $r[[r']]_{\bar{p}}|_{p_1}$ coincides with $r[[r']]_{\bar{p}}|_{\hat{p}_j.p'_1}$ and with $r|_{\hat{p}_j}[[r']]_{\bar{p}-\hat{p}_j}|_{p'_1}$. Since $|\bar{p}| \leq |\bar{p} - \hat{p}_j| + h$ holds, it follows $height(r[[r']]_{\bar{p}}|_{p_1}) \geq |\bar{p}| + height(r') - k \cdot h$ and we are done.

COROLLARY 5.16. *Suppose the hypothesis of the previous lemma and $height(r') > height(r|_{\bar{p}}) + k \cdot h(A)$, Then, $height(r[[r']]_{\bar{p}}|_{p.p_1}) > height(r|_p)$.*

LEMMA 5.17. *Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq,hom}$. Let r be a run of A satisfying $height(r) > \check{h}(A)$. Let $\check{p}$ be a position in $Pos(r)$ satisfying $|\check{p}| = height(r)$. Then, there exist two positions $p, \bar{p}$ of $Pos(r)$ satisfying the following conditions.*

—$p < \bar{p} \leq \check{p}$ *and* $|p| + h(A)^2 < |\bar{p}|$.
—$r|_p$ *and* $r|_{\bar{p}}$ *are runs such that* $root(r|_p) = root(r|_{\bar{p}})$.
—$r[[r|_p]]_{\bar{p}}$ *is a run.*

PROOF. We consider $p, \bar{p}$ to be the two positions given by Lemma 5.13. In order to conclude, it suffices to prove that $r[[r|_p]]_{\bar{p}}$ does not far-falsify any disequality. We proceed by contradiction by assuming that it far-falsifies a disequality $(p_1 \neq p_2)$ at a position p'. Thus, $\pi_\Sigma(r|_{p'.p_1}) \neq \pi_\Sigma(r|_{p'.p_2})$ and $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1}) = \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2})$ hold. By Lemma 5.3, p' can be assumed to be a prefix of $\bar{p}$. Note that, by the definition of far-falsification, no replaced position in $r[[r|_p]]_{\bar{p}}$ is a proper prefix of $p'.p_1$ nor $p'.p_2$. Since $\pi_\Sigma(r|_{p'.p_1}) \neq \pi_\Sigma(r|_{p'.p_2})$ holds, there exists a position p'' satisfying the following conditions.

—$|p''| \leq h$.
—No replaced position in $r[[r|_p]]_{\bar{p}}$ is a proper prefix of $p'.p_1.p''$ nor $p'.p_2.p''$.
—$\pi_\Sigma(r|_{p'.p_1.p''}) \neq \pi_\Sigma(r|_{p'.p_2.p''})$.
—Either (i) the roots of $\pi_\Sigma(r|_{p'.p_1.p''})$ and $\pi_\Sigma(r|_{p'.p_2.p''})$ differ, or (ii) some of $r|_{p'.p_1.p''}$ or $r|_{p'.p_2.p''}$ is a run.

We choose p'' to be minimal in size satisfying these conditions. Note that, since $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1}) = \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2})$ holds, then $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}) = \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''})$ also holds.

In case (i), it follows $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}) \neq \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''})$, a contradiction. Thus, assume that case (ii) holds. At this point, we distinguish the following cases.

—Assume that both $p'.p_1.p''$ and $p'.p_2.p''$ are prefixes of replaced positions in $r[[r|_p]]_{\bar{p}}$. Since either $r|_{p'.p_1.p''}$ or $r|_{p'.p_2.p''}$ is a run, by Lemma 5.14, either $r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}$ is a proper subterm of $r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''}$ or $r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''}$ is a proper subterm of $r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}$. In any case, $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}) \neq \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''})$ follows, a contradiction.
—Assume that none of $p'.p_1.p''$ and $p'.p_2.p''$ is a prefix of a replaced position in $r[[r|_p]]_{\bar{p}}$. In this case, $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''})$ coincides with $\pi_\Sigma(r|_{p'.p_1.p''})$, and $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''})$ coincides with $\pi_\Sigma(r|_{p'.p_2.p''})$. This is in contradiction with $\pi_\Sigma(r|_{p'.p_1.p''}) \neq \pi_\Sigma(r|_{p'.p_2.p''})$ and $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}) = \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''})$.
—Finally, assume that only one of $p'.p_1.p''$ and $p'.p_2.p''$ is a prefix of a replaced position in $r[[r|_p]]_{\bar{p}}$. We assume that it is $p'.p_1.p''$ (the other case reaches a contradiction analogously). Thus, $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''})$ coincides with $\pi_\Sigma(r|_{p'.p_2.p''})$. Note that $height(r|_{p'.p_2.p''}) < height(r|_{p'})$ holds. By the minimality selection for p'', it holds that $k = |\{p''' \mid p' \leq p''' < p'.p_1.p'' \wedge r|_{p'''} \text{ is a run}\}|$ is smaller than or equal to h. Recall that p' is a prefix of $\bar{p}$, $p'.p_1.p''$ is a prefix of a replaced position in $r[[r|_p]]_{\bar{p}}$, and since $\bar{p}$ is a prefix of $\check{p}$, then $height(r|_{p'}) = |(\bar{p} - p')| + height(r|_{\bar{p}})$ holds. Recall also that $|\bar{p} - p| > h^2$ holds, and hence, $height(r|_p) > height(r|_{\bar{p}}) + h^2$ holds. Thus, by Corollary 5.16, $height(r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}) > height(r|_{p'})$ holds. It follows $height(r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}) > height(r|_{p'.p_2.p''}) = height(\pi_\Sigma(r|_{p'.p_2.p''})) = height(\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''}))$. Therefore, $\pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_1.p''}) \neq \pi_\Sigma(r[[r|_p]]_{\bar{p}}|_{p'.p_2.p''}))$ holds, a contradiction.

Lemma 5.17 is enough to conclude the existence of infinite accepted terms when there is one with height greather than $\check{h}$. Nevertheless, we describe a concrete form of generating such infinite new terms that will be useful also for Section 7. Basically, we want the involved pumpings to take place at the same position, which in addition is bigger than h^2.

COROLLARY 5.18. *Let $A = \langle Q, \Sigma, F, \Delta \rangle$ be a $TA_{\neq,hom}$. Let r be a run of A satisfying $height(r) > \check{h}(A)$. Let $\check{p}$ be a position in $Pos(r)$ satisfying $|\check{p}| = height(r)$. Let $\bar{p}$ be the minimum prefix of $\check{p}$ in size satisfying that $r|_{\bar{p}}$ is a run and $|\bar{p}| > h(A)^2$.*

Then, there exists a run r' satisfying that $r[[r']]_{\bar{p}}$ is a run and $height(r') > height(r|_{\bar{p}})$.

PROOF. Let $p, \tilde{p}$ be the positions given by Lemma 5.17. Thus, they satisfy the following conditions.

—$p < \tilde{p} \leq \check{p}$ and $|p| + h^2 < |\tilde{p}|$.
—$r|_p$ and $r|_{\tilde{p}}$ are runs such that $root(r|_{\tilde{p}}) = root(r|_p)$.
—$r[[r|_p]]_{\tilde{p}}$ is a run.

In order to conclude, it suffices to observe that the pumping $r[[r|_p]]_{\tilde{p}}$ at $\tilde{p}$ can be seen as a pumping at $\bar{p}$, since $r[[r|_p]]_{\tilde{p}}$ equals $r[[\ ((r|_{\bar{p}})[[r|_p]]_{\tilde{p}-\bar{p}})\ ]]_{\bar{p}}$.

LEMMA 5.19. *Let A be a $TA_{\neq,hom}$. Let r be a run of A such that $height(r) > \check{h}(A)$. Then, there exists a position $\bar{p}$ in r and infinitely many different runs $r_1, r_2, \ldots$ of A such that:*

—*$|\bar{p}| > h(A)^2$ and $r|_{\bar{p}}$ is a run;*
—*all $root(r|_{\bar{p}}), root(r_1), root(r_2), \ldots$ coincide;*
—*all pumpings $r[[r_1]]_{\bar{p}}, r[[r_2]]_{\bar{p}}, \ldots$ are runs.*

PROOF. Let $\check{p}$ be any position in $Pos(r)$ satisfying $|\check{p}| = height(r)$. We choose $\bar{p}$ as the minimum prefix of $\check{p}$ in size satisfying that $r|_{\bar{p}}$ is a run and $|\bar{p}| > h^2$. By Corollary 5.18, there exists a run r_1 satisfying that $r_1' := r[[r_1]]_{\bar{p}}$ is a run and $height(r_1) > height(r|_{\bar{p}})$. Note that $\bar{p}$ is also the minimum position in size satisfying that $r_1'|_{\bar{p}}$ is a run, $|\bar{p}| > h^2$, and being a prefix of a position in $Pos(r_1')$ with length $height(r_1')$. Thus, Corollary 5.18 can be applied again, concluding the existence of a run r_2 satisfying that $r_2' := r_1'[[r_2]]_{\bar{p}} = r[[r_2]]_{\bar{p}}$ is a run and $height(r_2) > height(r_1'|_{\bar{p}}) = height(r_1)$. We conclude by noting that this inference can be iterated again and again.

COROLLARY 5.20. *The finiteness problem is decidable for $TA_{\neq,hom}$.*

PROOF. By Lemma 5.19, in order to decide this problem for a given a $TA_{\neq,hom}$ $A = \langle Q, \Sigma, F, \Delta\rangle$, it suffices to check whether there is a run r of A with an accepting resulting state and satisfying $height(r) > \check{h}(A)$. This question can be easily reduced to the emptiness problem, which is decidable, according to Corollary 5.11. To this end, it suffices to straightforwardly construct a new $TA_{\neq,hom}$ A' accepting the same language as A minus the terms with height smaller than or equal to $\check{h}(A)$, and then decide emptiness of $\mathcal{L}(A')$.

5.3. Complexity Analysis

Let $A = \langle Q, \Sigma, F, \Delta\rangle$ be a $TA_{\neq,hom}$. We start by analyzing the complexity of testing emptiness of A. According to Lemma 5.10, whenever $\mathcal{L}(A)$ is not empty, there exists a term accepted by A whose representation as a directed acyclic graph (DAG) has a number of nodes bounded by $\mathcal{P}_1(|A|) \cdot 2^{2^{\mathcal{P}_2(N,h)}}$ for some polynomials $\mathcal{P}_1, \mathcal{P}_2$. In order to represent a term, for each node of the DAG we must choose a function symbol and the direct subterms/child nodes of that node. Thus, the number of different terms that can be represented is bounded by $2^{\mathcal{P}_3(|A|)\cdot 2^{2^{\mathcal{P}_4(N,h)}}}$ for some polynomials $\mathcal{P}_3, \mathcal{P}_4$. Acceptation of each of such terms can be checked in $\mathcal{O}(\mathcal{P}_1'(|A|) \cdot 2^{2^{\mathcal{P}_2(N,h)}})$ for some polynomial $\mathcal{P}_1'$: according to Lemma 4.8, identity of subruns can be replaced by identity of subterms, and identity of subterms corresponds to identity of nodes if the DAG is minimum. Thus, it suffices to keep all states reaching each subterm/node in order to test whether the term is accepted. Hence, testing emptiness of A has $\mathcal{O}(2^{\mathcal{P}_3'(|A|)\cdot 2^{2^{\mathcal{P}_4(N,h)}}})$ space and time complexity for some polynomial $\mathcal{P}_3'$.

Now, we consider the problem of finiteness. Note that $\check{h}(A)$ is of the form $\mathcal{P}_1(|A|)\cdot 2^{\mathcal{P}_2(N,h)}$ for some polynomials $\mathcal{P}_1, \mathcal{P}_2$. First, we want a new $TA_{\neq,hom}$ rejecting all terms with height smaller than or equal to $\check{h}(A)$, and preserving the language $\mathcal{L}(A)$ for higher terms. It is easy to construct a TA B with size $\check{h}(A)+1$ recognizing the language of terms with height greater than $\check{h}(A)$, and intersect it with A, thus giving a new $\mathrm{TA}_{\neq,hom}$ $A \cap B$ recognizing the desired language. The size of $A \cap B$ is $\mathcal{O}(\check{h}(A))$, but $N(A \cap B)$ and $h(A \cap B)$ coincide with $N(A)$ and $h(A)$, respectively. Second, we want to test emptiness of $A \cap B$. By this analysis, this has $\mathcal{O}(2^{\mathcal{P}'_3(\check{h}(A)) \cdot 2^{2^{\mathcal{P}_4(N(A),h(A))}}})$ space and time complexity, that is, $\mathcal{O}(2^{\mathcal{P}_5(|A|) \cdot 2^{2^{\mathcal{P}_6(N(A),h(A))}}})$ for some polynomials $\mathcal{P}_5, \mathcal{P}_6$.

6. CONSEQUENCES

We first mention the consequences from the complementary construction.

THEOREM 6.1. *The inclusion problem $\mathcal{L}(A) \supseteq \mathcal{L}(B)$ is decidable for a $\mathrm{TA}_=$ A and a TA B given as input.*

PROOF. By Theorem 3.6, the complement $\mathrm{TA}_{\neq}$ $\bar{A}$ of A recognizes $\overline{\mathcal{L}(A)}$. It is well known how to compute a new $\mathrm{TA}_{\neq}$ recognizing the intersection of the languages represented by a $\mathrm{TA}_{\neq}$ and a TA (in fact, our Definition 4.11 subsumes this construction). Thus, a $\mathrm{TA}_{\neq}$ $\bar{A} \cap B$ recognizing $\overline{\mathcal{L}(A)} \cap \mathcal{L}(B)$ can be computed. It is well known (see Comon and Jacquemard [1994, 1997]) that emptiness of a $\mathrm{TA}_{\neq}$ is decidable (in fact, our Corollary 5.11 subsumes this result). Thus, we conclude by noting that deciding emptiness of $\overline{\mathcal{L}(A)} \cap \mathcal{L}(B)$ is equivalent to deciding $\mathcal{L}(A) \supseteq \mathcal{L}(B)$.

COROLLARY 6.2. *The universality problem is decidable for $\mathrm{TA}_=$.*

PROOF. Deciding $\mathcal{L}(A) = \mathcal{T}(\Sigma)$ is equivalent to deciding $\mathcal{L}(A) \supseteq \mathcal{T}(\Sigma)$, and since $\mathcal{T}(\Sigma)$ is regular, from Theorem 6.1 it follows that universality is decidable.

COROLLARY 6.3. *The finiteness problem of $\mathcal{L}(B) - \mathcal{L}(A)$ is decidable for a $\mathrm{TA}_=$ A and a TA B given as input.*

PROOF. As in the proof of Theorem 6.1, we can construct a $\mathrm{TA}_{\neq}$ recognizing $\overline{\mathcal{L}(A)} \cap \mathcal{L}(B)$. In order to conclude, we mention that it is well known that finiteness of a $\mathrm{TA}_{\neq}$ is decidable (in fact, our Corollary 5.20 subsumes this result).

COROLLARY 6.4. *The regularity test is undecidable for $\mathrm{TA}_{\neq}$, and thus, for reduction tree automata.*

PROOF. Since regularity is undecidable for $\mathrm{TA}_=$, our transformation of a $\mathrm{TA}_=$ into a $\mathrm{TA}_{\neq}$ recognizing the complement is a reduction from this problem into regularity of a $\mathrm{TA}_{\neq}$. Since $\mathrm{TA}_{\neq}$ are a particular case of reduction automata, the second part of the statement follows, too.

Now, we mention the consequences of combining the complement construction and the decidability of emptiness and finiteness of $\mathrm{TA}_{\neq,hom}$. Some of them generalize previous ones.

THEOREM 6.5. *The inclusion problem is decidable for images of tree homomorphisms, that is, $\mathcal{L}(H_A(A)) \supseteq \mathcal{L}(H_B(B))$ is decidable for a TA A, a TA B, and tree homomorphisms H_A and H_B given as input.*

PROOF. By Proposition 4.6 and Lemma 4.8, two $\mathrm{TA}_=$ A' and B' recognizing $\mathcal{L}(H_A(A))$ and $\mathcal{L}(H_B(B))$, respectively, can be computed. By Theorem 3.6 and Corollary 3.5, the complement $\mathrm{TA}_{\neq}$ $\bar{A}'$ of A' recognizes $\overline{\mathcal{L}(A')}$ and admits deterministic accepting runs. By Proposition 4.13, $\bar{A}' \cap B'$ recognizes $\overline{\mathcal{L}(A')} \cap \mathcal{L}(B')$. By Corollary 5.11, emptiness of

$\overline{\mathcal{L}(A')} \cap \mathcal{L}(B')$ is decidable. Thus, we conclude by noting that deciding emptiness of $\overline{\mathcal{L}(A')} \cap \mathcal{L}(B')$ is equivalent to deciding $\mathcal{L}(H_A(A)) \supseteq \mathcal{L}(H_B(B))$.

COROLLARY 6.6. *The equivalence problem is decidable for images of tree homomorphisms, that is, $\mathcal{L}(H_A(A)) = \mathcal{L}(H_B(B))$ is decidable for a TA A, a TA B, and tree homomorphisms H_A and H_B given as input.*

COROLLARY 6.7. *The inclusion and equivalence problems are decidable for ranges of bottom-up tree transducers.*

COROLLARY 6.8. *The finiteness problem of $\mathcal{L}(H_B(B)) - \mathcal{L}(H_A(A))$ is decidable for a TA A, a TA B, and tree homomorphisms H_A and H_B given as input.*

PROOF. As in the proof of Theorem 6.5, we can construct a $\text{TA}_{\neq,hom}$ recognizing $\overline{\mathcal{L}(H_A(A))} \cap \mathcal{L}(H_B(B))$. In order to conclude, we note that, by Corollary 5.20, finiteness of the previous set is decidable.

Our results have also implications in the context of term rewriting. The set of reducible terms by a term rewrite system can be described as the image of a tree homomorphism, and the set of normal forms, that is, the set of terms for which no rule can be applied, is just its complement. Thus, we can compare by inclusion and equality such sets with respect to two different term rewrite systems.

COROLLARY 6.9. *The questions $Red(R_1) = Red(R_2)$, $Red(R_1) \subseteq Red(R_2)$, $NF(R_1) = NF(R_2)$, $NF(R_1) \subseteq NF(R_2)$ and $NF(R_1) \subseteq Red(R_2)$ are decidable for given term rewrite systems R_1, R_2, where $Red(R)$ and $NF(R)$ denote the set of reducible terms and the set of normal forms, respectively, with respect to R.*

In Gilleron and Tison [1995], the question $Rel(L_1) \subseteq L_2$ is shown decidable for given regular tree languages L_1, L_2 and where the relation Rel is defined in several ways according to a given term rewrite system R. Tree homomorphisms are used to describe the image of L_1 through this relation: two tree homomorphisms H_l and H_r, and a tree language R_c are defined holding $Rel(L_1) = H_r(H_l^{-1}(L_1) \cap R_c)$, so that deciding $Rel(L_1) \subseteq L_2$ is done by testing $H_r(H_l^{-1}(L_1) \cap R_c) \subseteq L_2$. The tree homomorphisms H_l, H_r depend only on the rewrite system R. The tree language R_c depends also on the relation Rel. Our results allow to improve the results in Gilleron and Tison [1995] where R_c is a regular tree language. These are when Rel is one of the following relations: the one rewriting step, the one parallel rewriting step, the one-pass innermost-outermost step for left-linear term rewrite systems, and the one-pass outermost-innermost step for right-linear term rewrite systems (see Gilleron and Tison [1995] for details). In those cases, we are able to extend the results to decide the question $Rel_1(L_1) \subseteq Rel_2(L_2)$. Analogously, tree homomorphisms $H_{1,l}, H_{1,r}, H_{2,l}, H_{2,r}$ and regular tree languages $R_{1,c}, R_{2,c}$ can be defined holding $Rel_1(L_1) = H_{1,r}(H_{1,l}^{-1}(L_1) \cap R_{1,c})$ and $Rel_2(L_2) = H_{2,r}(H_{2,l}^{-1}(L_2) \cap R_{2,c})$, so that deciding $Rel_1(L_1) \subseteq Rel_2(L_2)$ is done by testing $H_{1,r}(H_{1,l}^{-1}(L_1) \cap R_{1,c}) \subseteq H_{2,r}(H_{2,l}^{-1}(L_2) \cap R_{2,c})$. Under the given assumptions, $H_{1,l}^{-1}(L_1) \cap R_{1,c}$ and $H_{2,l}^{-1}(L_2) \cap R_{2,c}$ are regular languages. Thus, this inclusion relates two images of regular languages through tree homomorphisms.

COROLLARY 6.10. *The question $H_{1,r}(H_{1,l}^{-1}(L_1) \cap R_{1,c}) \subseteq H_{2,r}(H_{2,l}^{-1}(L_2) \cap R_{2,c})$ is decidable for given tree homomorphisms $H_{1,l}, H_{1,r}, H_{2,l}, H_{2,r}$ and given regular tree languages $L_1, L_2, R_{1,c}, R_{2,c}$.*

COROLLARY 6.11. *The questions $Rel_1(L_1) = Rel_2(L_2)$ and $Rel_1(L_1) \subseteq Rel_2(L_2)$ are decidable for given regular tree languages L_1, L_2 and a given term rewrite system R, where Rel_1, rel_2 are defined as either the one rewriting step, the one parallel rewriting step,* *the one-pass innermost-outermost step if R is left-linear, and the one-pass outermost-innermost step if R is right-linear.*

6.1. Complexity Analysis

We first analyze the complexity of checking $\mathcal{L}(A) \supseteq \mathcal{L}(B)$ for a TA$_{=}$ A and a TA B. Recall that the complement construction from A generates a TA$_{\neq}$ $\bar{A}$ of size $2^{2 \cdot Maxarity(\Sigma) \cdot |\Sigma| \cdot |A|}$, but $N(\bar{A})$ coincides with $N(A)$, and $h(\bar{A})$ coincides with $h(A)$. The intersection $\bar{A} \cap B$ has a size bounded by $|B| \cdot 2^{2 \cdot Maxarity(\Sigma) \cdot |\Sigma| \cdot |A|}$. Moreover, $N(\bar{A} \cap B)$ coincides with $N(A)$, and $h(\bar{A} \cap B)$ coincides with $h(A)$. Comon and Jacquemard [1997] present an algorithm deciding emptiness of a TA$_{\neq}$ C with $\mathcal{O}(|C|^{4 \cdot N(C) \cdot h(C)})$ time and space complexity. Therefore, we can decide emptiness of $\bar{A} \cap B$ with $\mathcal{O}((2|B|)^{8 \cdot Maxarity(\Sigma) \cdot |\Sigma| \cdot |A| \cdot N(A) \cdot h(A)})$ time and space complexity, that is, $\mathcal{O}(2^{\mathcal{P}(|A|, |B|)})$ for some polynomial $\mathcal{P}$. Moreover, it follows that the complexity of the universality problem is exponential on the size of the input.

The analysis of the cost of deciding $\mathcal{L}(H_A(A)) \supseteq \mathcal{L}(H_B(B))$ is analogous to the previous one. The only difference is that instead of using the result of Comon and Jacquemard [1997] we use the decision result of emptiness of Section 5. The obtained cost is $\mathcal{O}(2^{2^{\mathcal{P}_1(|H_A|, |A|, |H_B|, |B|)} \cdot 2^{2^{\mathcal{P}_2(N(H_A) \cdot N_{Pos}(H_B), h(H_A) + h(H_B))}}})$ in time and space, for some polynomials $\mathcal{P}_1, \mathcal{P}_2$. Note that this is bounded by a tower of three exponentials on $|A| \cdot |B|$.

Similarly, the decision of finiteness of $\mathcal{L}(H_B(B)) - \mathcal{L}(H_A(A))$ has $\mathcal{O}(2^{2^{\mathcal{P}_3(|H_A|, |A|, |H_B|, |B|)} \cdot 2^{2^{\mathcal{P}_4(N(H_A) \cdot N_{Pos}(H_B), h(H_A) + h(H_B))}}})$ time and space complexity, for some polynomials $\mathcal{P}_3, \mathcal{P}_4$.

7. DECISION OF THE HOM PROBLEM

In this section, we prove that the HOM problem is decidable. The algorithm, which is simple, is presented in the first subsection. The remaining subsections prove its correctness. To this end, we make use of almost all results obtained for TA$_{\neq, hom}$ in the previous sections.

7.1. The Algorithm Deciding the HOM Problem

Definition 7.1. Let $A_{hom} = \langle Q, \Sigma, F, \Delta \rangle$ be a TA$_{hom}$. Let $\mathtt{h}$ be a natural number. The *linearization* of A_{hom} by $\mathtt{h}$ is the TA$_{hom}$ $\langle Q, \Sigma, F, \Delta' \rangle$, denoted *linearize*($A_{hom}, \mathtt{h}$), where Δ' is the set of all rules of the form $s[s_1]_{p_1} \cdots [s_n]_{p_n} \to q$ such that:

replace constrained subtrees
by trees of bounded height

—a rule of the form $s \xrightarrow{c} q$ occurs in Δ;
—$p_1, \ldots, p_n$ are the positions occurring in c;
—for each i in $\{1, \ldots, n\}$, s_i is a term in $\mathcal{L}(A_{hom}, s|_{p_i})$ such that $height(s_i) \leq \mathtt{h}$;
—for each i, j in $\{1, \ldots, n\}$ such that $(p_i = p_j)$ occurs in c, $s_i = s_j$ holds.

It is straightforward that a linearization of any TA$_{hom}$ is computable and recognizes a regular language, since no equality constraints appear. It is also clear that $\mathcal{L}(A_{hom})$ includes the language of any of its linearizations. Moreover, in the case where $\mathcal{L}(A_{hom})$ is included in some of its linearizations, we can conclude that $\mathcal{L}(A_{hom})$ is regular. i.e. when constraints are applied only to finite languages
and thus equal to

Example 7.2. Let $A_{ex3} = \langle Q, \Sigma, F, \Delta \rangle$ be a TA$_{hom}$ such that $Q = \{q, q'\}$ and $F = \{q'\}$ hold, $\Sigma = \{f^{(2)}, a^{(0)}$, $h^{(1)}\}$ holds, and the rules of Δ are the following.

—$\rho_1 = a \to q$.
—$\rho_2 = h(q) \to q$.
—$\rho_3 = f(q, q) \xrightarrow{1=2} q'$.
—$\rho_4 = f(q, h(q)) \to q'$.

Then, the linearization *linearize*($A_{ex3}, \underline{0}$) is $\langle Q, \Sigma, F, \Delta' \rangle$, where Δ' is the set of rules $\{\rho_1, \rho_2, \rho'_3, \rho_4\}$ such that ρ'_3 is the rule $f(a, a) \to q'$. It is easy to see that the language

$f(h(a), h(a)) \in L(A_{ex_3})$, $\in L(lin(A_{ex3}, 0))$

of the linearization contains the original language of A_{ex3}, and it follows that $\mathcal{L}(A_{ex3})$ is regular.

But if we consider the TA_{hom} $A_{ex4} = \langle Q, \Sigma, F, \Delta'' \rangle$, where Δ'' is $\Delta - \{\rho_4\}$, it can be proved that no linearization of A_{ex4} contains the original language.

The key point for deciding the HOM problem using linearization is stated by the following lemma.

LEMMA 7.3. *Let A_{hom} be a TA_{hom}. Let $\tilde{h}$ be $\check{h}(A_{hom} \cap \overline{A_{hom}})$. Suppose that $\mathcal{L}(A_{hom})$ is not included in $\mathcal{L}(linearize(A_{hom}, \tilde{h}))$. Then, $\mathcal{L}(A_{hom})$ is not regular.*

The proof of this lemma is done along the next subsections. It provides a simple decision algorithm for the HOM problem, described as follows.

—*Input.* A tree automaton A and a tree homomorphism H.
—Construct a TA_{hom} A_{hom} recognizing $H(A)$.
—Construct the linearization B of A_{hom} by $\tilde{h} = \check{h}(A_{hom} \cap \overline{A_{hom}})$.
—If $\mathcal{L}(linearize(A_{hom}, \tilde{h}))$ includes $\mathcal{L}(A_{hom})$ then halt with output "REGULAR".
—Otherwise, halt with output "NON-REGULAR".

The construction of the second item can be done according to Proposition 4.6. The $\mathrm{TA}_{\neq}$ $\overline{A_{hom}}$ is constructed according to Definitions 3.1 and 4.10. Recall that, by Corollary 3.5, $\overline{A_{hom}}$ admits deterministic accepting runs. The $\mathrm{TA}_{\neq,hom}$ $A_{hom} \cap \overline{A_{hom}}$ is constructed according to Definition 4.11. The natural number $\tilde{h}$ is computed according to $\check{h}$, which is the one of Definition 5.12. There is a straightforward and fast algorithm based on a fixpoint computation for deciding the inclusion of the fourth item. In Section 7.5, we comment how to decide such an inclusion in triple exponential time.

THEOREM 7.4. *The HOM problem is decidable.*

To conclude, it remains to prove Lemma 7.3.

7.2. A Nonterminating Process Detecting Nonregularity

In order to prove Lemma 7.3, we describe a nonterminating process. We emphasize that this process is not executed in order to decide regularity (the algorithm has already been presented in the previous section). It will just help us to argue that the statement of Lemma 7.3 is true.

The nonterminating process deals with sets of terms with equality constraints. They represent an infinite set of ground terms: the ones obtained by applying substitutions holding the constraints.

Definition 7.5. Let $A_{hom} = \langle Q, \Sigma, F, \Delta \rangle$ be a TA_{hom}. A *constrained term* with respect to A_{hom} is a pair $t|c$, where t is a term in $\mathcal{T}(\Sigma \cup Q)$, and c is a conjunction/set of equalities of positions $(p_1 = p_2)$ satisfying that $t|_{p_1} = t|_{p_2} \in Q$ holds and p_1 and p_2 are different. Moreover, if $(p_1 = p_2), (p_2 = p_3)$ occur in c, then $(p_1 = p_3)$ also occurs in c, for arbitrary positions p_1, p_2, p_3. We identify a term t with a constrained term $t|\emptyset$, that is, with an empty conjunction of equalities. We also define the replacement of a subterm $t|_p$ in a term t at position p by a constrained term $s|c$, denoted $t[s|c]_p$, as $t[s]_p| \bigwedge_{(p_1=p_2)\in c}(p.p_1 = p.p_2)$.

An *instance* of $t|c$ is a ground term of the form $t[s_1]_{p_1} \cdots [s_n]_{p_n}$, where $\{p_1, \ldots, p_n\}$ are the positions p_i satisfying $t|_{p_i} \in Q$, $\{s_1, \ldots, s_n\}$ are ground terms satisfying $s_i \in \mathcal{L}(A_{hom}, t|_{p_i})$ such that, for each $(p_i = p_j)$ occurring in c, $s_i = s_j$ holds. The set of instances of a constrained term $t|c$ with respect to A_{hom} is denoted by *instances*$(t|c, A_{hom})$, or by *instances*$(t|c)$ when A_{hom} is clear from the context. The *set of instances* of a set S of constrained terms with respect to a TA_{hom} A_{hom}, denoted *instances*(S, A_{hom}) or *instances*(S) when A_{hom} is clear from the context, is $\bigcup_{(t|c)\in S}(instances(t|c))$.

Definition 7.6. Let $A_{hom} = \langle Q, \Sigma, F, \Delta\rangle$ be a TA_{hom}. Let $\mathtt{h}$ be a natural number. Let $t|c$ be a constrained term with respect to A_{hom}. The *linearization* of $t|c$ by $\mathtt{h}$, denoted $linearize(t|c, \mathtt{h})$, is the set of terms $t[s_1]_{p_1} \cdots [s_n]_{p_n}$, where $\{p_1, \ldots, p_n\}$ are the positions occurring in c, $\{s_1, \ldots, s_n\}$ are ground terms satisfying $s_i \in \mathcal{L}(A_{hom}, t|_{p_i})$, each $height(s_i)$ is smaller than or equal to $\mathtt{h}$, and for each $(p_i = p_j)$ occurring in c, $s_i = s_j$ holds.

Example 7.7. Consider the TA_{hom} of Example 7.2 and the constrained term $t|c = f(q, f(q, q))|\{1 = 2.2\}$. Then, $linearize(t|c, 0)$ is $f(a, f(q, a))$.

The nonterminating process has a TA_{hom} $A_{hom} = \langle Q, \Sigma, F, \Delta\rangle$ as input, constructs a set S of constrained terms with respect to A_{hom}, and proceeds by modifying S until it (eventually) detects a condition implying non-regularity. The following invariants are satisfied at the beginning of each step of the process.

I1. Each $t|c$ in S satisfies that c is $\emptyset$, i.e. S contains just terms over $\mathcal{T}(\Sigma \cup Q)$.
I2. $instances(S)$ is equal to $\mathcal{L}(A_{hom})$.
I3. For each two terms t_1, t_2 in S and each two positions p_1, p_2 satisfying that $t_1|_{p_1}$ and $t_2|_{p_2}$ are in Q, $\big||p_2| - |p_1|\big| \le h(A_{hom})$.

The description of the nonterminating process is as follows.

(1) Input: A TA_{hom} $A_{hom} = \langle Q, \Sigma, F, \Delta\rangle$.
(2) Assign $S := F$.
(3) If all terms in S are in $\mathcal{T}(\Sigma)$, that is, no state in Q occurs in S, then halt with output "REGULAR".
(4) Otherwise, let p be a position minimal in size among $\{p' \mid \exists t \in S : t|_{p'} \in Q\}$. Let t be a term of S satisfying $t|_p \in Q$. Let q be $t|_p$. Assign $S := (S - \{t\}) \cup \{t[s|c]_p \mid (s \xrightarrow{c} q) \in \Delta\}$. (Note that, at this point, S satisfies all the invariants except I1.)
(5) While S does not satisfy I1, do:
    (a) Let $t'|c'$ be a constrained term in S where c' is not empty.
    (b) If there exists a position p' in c' and a term t'' in $instances(t'|c') - instances(S - \{t'|c'\})$ such that $height(t''|_{p'}) > \check{h}(A_{hom} \cap \overline{A_{hom}})$, then halt with output "NON-REGULAR".
    (c) Otherwise, assign $S := (S - \{t'|c'\}) \cup linearize(t'|c', \check{h}(A_{hom} \cap \overline{A_{hom}}))$.
(6) Go to (3).

It is clear that this process satisfies all the invariants when it passes through item (3). We insist again that this process is not executed. Thus, it does not matter if the used instructions are computable (nevertheless they are).

LEMMA 7.8. *If the process does not halt with output "NON-REGULAR", then* $\mathcal{L}(A_{hom}) \subseteq \mathcal{L}(linearize(A_{hom}, \check{h}(A_{hom} \cap \overline{A_{hom}})))$.

PROOF. Let Δ' the set of rules of $linearize(A_{hom}, \check{h}(A_{hom} \cap \overline{A_{hom}}))$. Note that, when step 3 is executed, any term u inside S satisfies $u \to^*_{\Delta'} q'$ for some $q' \in F$. Thus, in order to conclude, it suffices to prove that any term u in $\mathcal{L}(A_{hom})$ is included in S at some point of the execution, under the assumptions of the lemma.

Assume that the process does not halt with output "NON-REGULAR". Let u be a term in $\mathcal{L}(A_{hom})$. Note that, either the process halts with output "REGULAR" at step (3), or it does not halt. In the first case, by Invariant I2, u belongs to S when the process halts. In the second case, at some point of the execution in step (3), the minimal position p in size among $\{p' \mid \exists t \in S : t|_{p'} \in Q\}$ satisfies $|p| > height(u)$. Thus, by Invariant I2, at this point of the execution, u is also in S, and we are done.

The contraposition of the statement in Lemma 7.8 says that, if $\mathcal{L}(A_{hom})$ is not included into $\mathcal{L}(linearize(A_{hom}, \check{h}(A_{hom} \cap \overline{A_{hom}})))$, then the process halts with output "NON-REGULAR". Hence, in order to prove Lemma 7.3, it remains to see that, when the process halts with output "NON-REGULAR", $\mathcal{L}(A_{hom})$ is not regular. This reduces to check that, when the condition in step 5b of the process is satisfied, it follows that $\mathcal{L}(A_{hom})$ is not regular.

7.3. Intuition behind the Nonregularity Condition

The proofs of the following subsection are very technical. Here, we briefly try to provide some intuition, in order to show one of the fundamental tricks of the proofs, and how the constructions of previous sections play an important role.

For simplifying, first we rename A_{hom} to A, and suppose that A is just a deterministic TA instead of an arbitrary TA$_{hom}$, and that S is the set of constrained terms $\{f(q_1, q_1)|(1 = 2),\ f(q_2, q_1)\}$ for different states q_1, q_2 of A. Also suppose that we have a "big enough" term $f(t, t)$ in $instances(f(q_1, q_1)|(1 = 2)) - instances(f(q_2, q_1))$. Our goal is to prove that $instances(S)$ is not regular.

Note that, by the definition of instance, t belongs to $\mathcal{L}(A, q_1)$, and since A is deterministic and q_1 and q_2 are different, t does not belong to $\mathcal{L}(A, q_2)$. Since $f(t, t)$ is "big", by means of pumpings on t we can construct infinitely many ~~infinite~~ terms $t_1, t_2, \ldots$ such that each t_i reaches q_1 with A. Thus, $f(t_i, t_i)$ is in $instances(f(q_1, q_1)|(1 = 2)) - instances(f(q_2, q_1))$ by the same reason, that is, each t_i is in $\mathcal{L}(A, q_1)$, and hence, it is not in $\mathcal{L}(A, q_2)$.

Now, in order to prove that $instances(S)$ is not regular, we proceed by contradiction by assuming the existence of a TA C satisfying $\mathcal{L}(C) = instances(S)$. From these infinite terms we can choose a t_i which is "big enough" with respect to the size of A and C. Hence, by pigeonhole principle, we can choose a pumping on t_i, that produces a new t_i' different from t_i but reaching the same state with A and the same state with C. Thus, $f(t_i', t_i)$ reaches an accepting state with C, and t_i' reaches q_1 with A. Note that $f(t_i', t_i)$ is not an instance of $f(q_1, q_1)|(1 = 2)$ because t_i' and t_i are different, neither an instance of $f(q_2, q_1)$ because t_i' reaches q_1 with A. Hence, we have reached a contradiction since $f(t_i', t_i)$ is in $\mathcal{L}(C) - instances(S)$.

The previous ideas have a big obstacle when A is an arbitrary TA$_{hom}$ instead of just a TA. The reason is that the starting homomorphism could be noninjective, and, in fact, TA$_{hom}$ are inherently nondeterministic: thus, even when t reaches q_1 and not q_2 with A, it may happen that the generated $t_1, t_2, \ldots$ by pumping reach both q_1 and q_2 with A. Hence, the new infinitely many terms $f(t_i, t_i)$ are not necessarily in $instances(f(q_1, q_1)|(1 = 2)) - instances(f(q_2, q_1))$, and we cannot repeat this argument. Here is when $\overline{A}$ plays an important role. Note that there exists an execution on t with $\overline{A}$ reaching a set S this is a different S containing q_2. If t is "big enough" with respect to the size of A and $\overline{A}$, then, we can perform infinite pumpings producing terms $t_1, t_2, \ldots$ such that each t_i reaches q_1 with A, and S with $\overline{A}$, thus ensuring that each t_i cannot reach q_2 with A. This allows to reproduce the same argument as before.

7.4. Correctness of the Process

We will use the following lemma, which characterizes when a term is not an instance of a constrained term. It holds due to the fact that each symbol has a fixed arity.

LEMMA 7.9. *Let $A_{hom} = \langle Q, \Sigma, F, \Delta \rangle$ be a TA$_{hom}$. Let $s|d$ be a constrained term. Let t be a term in $\mathcal{T}(\Sigma) - instances(\{s|d\})$. Then, one of the following conditions hold.*

—*There is a position p in $Pos_{\Sigma}(s) \cap Pos(t)$ such that $root(s|_p)$ is different from $root(t|_p)$.*
—*There is a position p in $Pos_Q(s)$ such that $t|_p$ is not in $\mathcal{L}(A, s|_p)$.*
—*There is an equality $(p_1 = p_2)$ in d such that $t|_{p_1}$ is different from $t|_{p_2}$.*

Condition in step (5b) of the process says that there exists a position p' in c' and a term t'' in $instances(t'|c') - instances(S - \{t'|c'\})$ such that $height(t''|_{p'}) > \check{h}(A_{hom} \cap \overline{A_{hom}})$. From this condition, we must prove that $\mathcal{L}(A_{hom})$ is not regular. This is done in two steps, by Lemmas 7.10 and 7.11.

As a first step, Lemma 7.10 (where t'' is renamed to t') proves that when such a t'' exists, then there exist infinitely many other terms satisfying the same assumptions as t''. Moreover, these variants of t'' are obtained by replacing the subterm of t'' at position p', and replacing also all those other positions that c' forces to have identical terms to the one pending at p'. The infinite variants are obtained by intersecting a run of A_{hom} and a maximal run of $\overline{A_{hom}}$ on t'' and applying Lemma 5.19 to obtain the desired infinite pumpings below p'. The condition $height(t''|_{p'}) > \check{h}(A_{hom} \cap \overline{A_{hom}})$ guarantees that the pumping is possible. The fact that the variants of t'' are also instances of $t'|c'$ follows from the fact that the state reached at p' with the alternative runs of A_{hom} is the same, and because we are also replacing all those other positions which c' force to have identical terms to the one pending at p'. The fact that the variants of t'' are neither in $instances(S - \{t'|c'\})$ uses the following idea: $\overline{A_{hom}}$ ensures that the alternative runs cannot reach the states which were not reachable in t'' at the same corresponding positions. Thus, if t'' is not an instance of a constrained term in $S - \{t'|c'\}$ because a state is not reachable at a certain position, then the same happens with their variants.

LEMMA 7.10. *Let $A_{hom} = \langle Q, \Sigma, F, \Delta\rangle$ be a TA_{hom}. Let S be a set of constrained terms satisfying I2 and I3. Let $t|c$ be a constrained term in S. Let $\check{p}_1, \dots, \check{p}_n$ be the positions $\check{p}_i$ satisfying $t|_{\check{p}_i} \in Q$. Suppose that $\check{p}_1$ occurs in c, and that, without loss of generality, $\check{p}_2, \dots, \check{p}_k$ are all positions $\check{p}_j$ such that $(\check{p}_1 = \check{p}_j)$ occurs in c. Suppose that $t' = t[t_1]_{\check{p}_1} \cdots [t_1]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}} \cdots [t_n]_{\check{p}_n}$ is a term in $instances(t|c) - instances(S - \{t|c\})$ such that $height(t_1) > \check{h}(A_{hom} \cap \overline{A_{hom}})$ holds.*

Then, there exist infinitely many terms $t_{1,1}, t_{1,2}, \dots$ such that all $t[t_{1,j}]_{\check{p}_1} \cdots [t_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}} \cdots [t_n]_{\check{p}_n}$ are also terms in $instances(t|c) - instances(S - \{t|c\})$.

PROOF. By the definition of instance, there exists a run r_1 of A_{hom} such that $\pi_\Sigma(r_1) = t_1$ and the resulting state of r_1 is $t|_{\check{p}_1}$. Let $\bar{r}$ be a run of $\overline{A_{hom}}$ satisfying $\pi_\Sigma(\bar{r}) = t'$ and all conditions given by Corollary 4.9. In particular, $\bar{r}$ is a deterministic run, for each $p \in Pos(\bar{r})$ it holds that $\bar{r}|_p$ is a run with a resulting state including $\{q \in Q \mid t'|_p \notin \mathcal{L}(A_{hom}, q)\}$, and $\bar{r}|_{\check{p}_1} = \cdots = \bar{r}|_{\check{p}_k}$ holds. Let $\hat{r}_1$ be the run $r_1 \cap \bar{r}|_{\check{p}_1}$ of $A_{hom} \cap \overline{A_{hom}}$. By Lemma 5.19, there exist a position $\bar{p}$ in $\hat{r}_1$ and infinitely many different runs $\hat{r}_{1,1}, \hat{r}_{1,2}, \dots$ of $A_{hom} \cap \overline{A_{hom}}$ such that:

—$|\bar{p}| > h(A_{hom} \cap \overline{A_{hom}})^2$ and $\hat{r}_1|_{\bar{p}}$ is a run.
—All $root(\hat{r}_1|_{\bar{p}}), root(\hat{r}_{1,1}), root(\hat{r}_{1,2}), \dots$ coincide.
—All pumpings $\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}}, \hat{r}_1[[\hat{r}_{1,2}]]_{\bar{p}}, \dots$ are runs.

We define $\hat{t}_{1,1} := \pi_\Sigma(\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}})$, $\hat{t}_{1,2} := \pi_\Sigma(\hat{r}_1[[\hat{r}_{1,2}]]_{\bar{p}}), \dots$.

Now, consider p as any position in $Pos(\hat{r}_1)$ satisfying $|p| \le h(A_{hom} \cap \overline{A_{hom}})$. Since $|\bar{p}| > h(A_{hom} \cap \overline{A_{hom}})^2$, by the last part of the statement in Lemma 5.2, no replaced position in $\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}}$ is a prefix of p. Thus, for each of such positions p and each $j \ge 1$, $root(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}|_p)$ coincides with $root(\hat{r}_1|_p)$. By Lemma 4.18, $\pi_{\neq}(\hat{r}_1) = \bar{r}|_{\check{p}_1}$ and $\pi_{hom}(\hat{r}_1) = r_1$ hold, and by Lemma 4.19, for each of such positions p and each $j \ge 1$, the resulting states of $\bar{r}|_{\check{p}_1}|_p$ and $(\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}))|_p$ coincide. Moreover, again by Lemma 4.19, the resulting states of r_1 and $\pi_{hom}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})$ coincide, that is, they are $t|_{\check{p}_1}$.

One of the particular implications of these comments is that each $t[\hat{t}_{1,j}]_{\check{p}_1} \cdots [\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}} \cdots [t_n]_{\check{p}_n}$ is an instance of $t|c$. Thus, at first look, we could try to define the desired terms $t_{1,1}, t_{1,2}, \dots$ of the statement of the lemma as $\hat{t}_{1,1}, \hat{t}_{1,2}, \dots$, respectively. The problem is that some $t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n}$ could also be in *instances*$(S-\{t|c\})$. In order to conclude, it suffices to see that only a finite number of them satisfy this condition. To this end, we show that, for each $s|d$ in $S-\{t|c\}$, at most a finite number of terms $t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n}$ are instances of $s|d$.

Consider any constrained term $s|d$ in $S-\{t|c\}$. Since t' is not an instance of $s|d$, according to Lemma 7.9, we can distinguish the following cases.

—Assume that there is a position p in $Pos_\Sigma(s)\cap Pos(t')$ such that $root(s|_p)$ is different from $root(t'|_p)$. We distinguish the following cases:
 —Suppose first that p is of the form $\check{p}_i.p'.p''$ for some i in $\{1,\ldots,k\}$ and some positions p',p'' such that $|p'|=h(A_{hom}\cap\overline{A_{hom}})+1>h(A_{hom})$. By Invariant I3, it holds that $s|_{\check{p}_i.p'}$ is a term in $\mathcal{T}(\Sigma)$, that is, without any symbol in Q. Note that $t'|_{\check{p}_i.p'}$ is a term different from $s|_{\check{p}_i.p'}$ because they differ at the symbol located at their relative position p''. Recall that no replaced position in $\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}}$ is a proper prefix of p'. Thus, by Lemma 5.6, at most one term $\hat{t}_{1,j}$ makes $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{\check{p}_i.p'}$ equal to $s|_{\check{p}_i.p'}$, and we are done.
 —Second, suppose p is not of the form $\check{p}_i.p'.p''$ for some i in $\{1,\ldots,k\}$ and some positions p',p'' such that $|p'|=h(A_{hom}\cap\overline{A_{hom}})+1$. In this case, $root(t'|_p)$ coincides with all $root((t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_p)$, and thus, it differs from $root(s|_p)$. Hence, none of these terms is an instance of $s|d$, and we are done.

—Assume that there is a position p in $Pos_Q(s)$ such that $t'|_p$ is not in $\mathcal{L}(A,s|_p)$. Let q be $s|_p$ and let S_p be the resulting state of $\bar{r}|_p$. By the selection of $\bar{r}$, it holds that q is in S_p.
 Let us fix a $j\geq 1$ and suppose that $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}\cdots[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_k}$ is a run of $\overline{A_{hom}}$. Note that, by Invariant I3, either no position in $\check{p}_1,\ldots,\check{p}_k$ is a prefix of p, or p is of the form $\check{p}_i.p'$ for some i in $\{1,\ldots,k\}$ and some p' satisfying $|p'|\leq h(A_{hom})\leq h(A_{hom}\cap\overline{A_{hom}})$. In the second case, recall that, for such a position p', $root(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}|_{p'})$ coincides with $root(\hat{r}_1|_{p'})$. Thus, in any case, the resulting state of $(\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}\cdots[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_k})|_p$ is S_p. By Lemma 4.9, it follows that $t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n}$ is not an instance of $s|d$.
 Hence, in order to conclude, it suffices to argue that the terms $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}\cdots[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_k}$ are runs of $\overline{A_{hom}}$ except for a finite number of j's.
 Note that $\bar{r}$ and each term $\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})$ is a run of $\overline{A_{hom}}$, and $root(\bar{r}|_{\check{p}_1})=\cdots=root(\bar{r}|_{\check{p}_k})=root(\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}))$ holds. Thus, if, for a concrete j, $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}\cdots[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_k}$ is not a run of $\overline{A_{hom}}$, there must exist a position p'' satisfying the following assumptions:
 —$p''<\check{p}_i$ for some i in $\{1,\ldots,k\}$.
 —$root(\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}\cdots[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_k}|_{p''})$, that is, $root(\bar{r}|_{p''})$, is a rule $f(q_1,\ldots,q_m)\xrightarrow{e}q'$ where e contains a disequality $p_1\neq p_2$ such that $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{p''.p_1}$ is equal to $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{p''.p_2}$.
 Since $\bar{r}$ is a run of $\overline{A_{hom}}$, $t'|_{p''.p_1}$ is different from $t'|_{p''.p_2}$. Recall that, all replaced positions in $\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}$ have length greater than $h(A_{hom}\cap\overline{A_{hom}})\geq h(\overline{A_{hom}})$. Thus, $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{p''.p_1}$ and $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{p''.p_2}$ can be obtained from $t'|_{p''.p_1}$ and $t'|_{p''.p_2}$, respectively, by replacing $t'|_{\check{p}_1.\bar{p}}$ by $\pi_\Sigma(\hat{r}_{1,j})$ at some positions which are independent from j. By Lemma 5.6, only one term can satisfy this statement for such p'' and $p_1\neq p_2$.
 The selections for p'' and $p_1\neq p_2$ are finitely bounded. Thus, at most for a finite number of j's, $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}\cdots[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_k}$ is not a run, and we are done.

—Finally, assume that there is an equality $(p_1=p_2)$ in d such that $t'|_{p_1}$ is different from $t'|_{p_2}$. Note that both $s|_{p_1}$ and $s|_{p_2}$ are identical and in Q. Hence, by Invariant I3,

p_1 and p_2 are not of the form $\check{p}_i.p'.p''$ for some i in $\{1,\ldots,k\}$ and some positions p', p'' such that $|p'| = h(A_{hom} \cap \overline{A_{hom}}) + 1 > h(A_{hom})$. Note also that, in the case where some $\check{p}_i$ for i in $\{1,\ldots,k\}$ is a prefix of p_1 (p_2), no replaced position in $\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}}$ is a proper prefix of $p_1 - \check{p}_i$ ($p_2 - \check{p}_i$). Thus, for each $j \geq 1$ $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{p_1}$ and $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{p_2}$ can be obtained from $t'|_{p_1}$ and $t'|_{p_2}$, respectively, by replacing $t'|_{\check{p}_1.\bar{p}}$ by $\pi_\Sigma(\hat{r}_{1,j})$ at some positions which are independent from j. By Lemma 5.6, at most one term $\hat{t}_{1,j}$ makes $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{p_1}$ equal to $(t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[\hat{t}_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n})|_{p_2}$. Hence, for all the remaining j's, $t[\hat{t}_{1,j}]_{\check{p}_1}\cdots[t'_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n}$ is not an instance of $s|d$, due to the same reason as t', and we are done.

As a second step, Lemma 7.11 proves that, when such infinite variants of t'' exist, nonregularity of *instances*(S), which is $\mathcal{L}(A_{hom})$ according to Invariant I2, can be concluded. It proceeds by contradiction by assuming that a TA C recognizes *instances*(S). Among the infinitely many variants of t'', we choose one, say t''' (called t' in Lemma 7.11), such that the subterm pending at position p' is bigger than $\check{h}(C \cap \overline{A_{hom}})$. This allows us to apply again Lemma 5.19 in order to obtain the infinite variants of t'''. Nevertheless, this time, the variants are obtained by replacing the subterm at position p', but all those other positions that c' forces to have identical terms to the one pending at p' are not replaced. This way, the variants of t''' are not in *instances*$(t'|c')$. Moreover, they are not in *instances*$(S - \{t'|c'\})$ by the same reasons as before: $\overline{A_{hom}}$ ensures that the alternative runs cannot reach the states that were not reachable in t''' at the same corresponding positions. Thus, the variants of t''' are not in *instances*(S). Since the pumping is also done with C, they are still in $\mathcal{L}(C)$. In conclusion, there exist terms in $\mathcal{L}(C)-$*instances*(S). This is in contradiction with the fact that C recognizes *instances*(S).

LEMMA 7.11. *Let $A_{hom} = \langle Q, \Sigma, F, \Delta\rangle$ be a TA$_{hom}$. Let S be a set of constrained terms satisfying I2 and I3. Let $t|c$ be a constrained term in S. Let $\check{p}_1,\ldots,\check{p}_n$ be the positions $\check{p}_i$ satisfying $t|_{\check{p}_i} \in Q$. Suppose that $\check{p}_1$ occurs in c, and that, without loss of generality, $\check{p}_2,\ldots,\check{p}_k$ are all positions $\check{p}_j$ such that $(\check{p}_1 = \check{p}_j)$ occurs in c. Suppose that there exist terms $t_{k+1},\ldots,t_n$ and infinitely many terms $t_{1,1}, t_{1,2},\ldots$ such that all $t[t_{1,j}]_{\check{p}_1}\cdots[t_{1,j}]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n}$ are in instances$(t|c)$ − instances$(S - \{t|c\})$.*
Then, instances(S) is not regular.

PROOF. We proceed by contradiction by assuming that *instances*(S) is regular. Thus, let C be a TA recognizing *instances*(S). Note that, in particular, C is a TA$_{hom}$. Among all the terms $t_{1,j}$ we choose one, called t_1, such that $height(t_1) > \check{h}(C \cap \overline{A_{hom}})$ holds. Note that $t' = t[t_1]_{\check{p}_1}\cdots[t_1]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}}\cdots[t_n]_{\check{p}_n}$ is in *instances*$(t|c)-$*instances*$(S-\{t|c\})$. In particular, t' is in *instances*(S), and hence, t' belongs to $\mathcal{L}(C)$. Thus, there exists a run r of C with a resulting accepting state such that $\pi_\Sigma(r) = t'$. Let $\bar{r}$ be a run of $\overline{A_{hom}}$ satisfying $\pi_\Sigma(\bar{r}) = t'$ and all conditions given by Lemma 4.9. In particular, $\bar{r}$ is a deterministic run, and for each $p \in Pos(\bar{r})$ it holds that $\bar{r}|_p$ is a run with a resulting state including $\{q \in Q \mid t'|_p \notin \mathcal{L}(A_{hom}, q)\}$. Let $\hat{r}_1$ be the run $r|_{\check{p}_1} \cap \bar{r}|_{\check{p}_1}$ of $C \cap \overline{A_{hom}}$. By Lemma 5.19, there exist a position $\bar{p}$ in $\hat{r}_1$ and infinitely many different runs $\hat{r}_{1,1}, \hat{r}_{1,2},\ldots$ of $C \cap \overline{A_{hom}}$ such that:

—$|\bar{p}| > h(C \cap \overline{A_{hom}})^2$ and $\hat{r}_1|_{\bar{p}}$ is a run;
—all $root(\hat{r}_1|_{\bar{p}}), root(\hat{r}_{1,1}), root(\hat{r}_{1,2}),\ldots$ coincide;
—all pumpings $\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}}, \hat{r}_1[[\hat{r}_{1,2}]]_{\bar{p}},\ldots$ are runs.

We define $\hat{t}_{1,1} := \pi_\Sigma(\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}})$, $\hat{t}_{1,2} := \pi_\Sigma(\hat{r}_1[[\hat{r}_{1,2}]]_{\bar{p}})$, ...

Now, consider p as any position in $Pos(\hat{r}_1)$ satisfying $|p| \leq h(C \cap \overline{A_{hom}})$. Since $|\bar{p}| > h(C \cap \overline{A_{hom}})^2$, by the last part of the statement in Lemma 5.2, no replaced position in $\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}}$ is a prefix of p. Thus, for each of such positions p and each $j \geq 1$, $root(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}|_p)$ coincides with $root(\hat{r}_1|_p)$. By Lemma 4.18, $\pi_{\neq}(\hat{r}_1) = \bar{r}|_{\check{p}_1}$ and $\pi_{hom}(\hat{r}_1) = r|_{\check{p}_1}$ hold, and by Lemma 4.19, for each of such positions p and each $j \geq 1$, the resulting states of $\bar{r}|_{\check{p}_1}|_p$ and $(\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}))|_p$ coincide. Moreover, again by Lemma 4.19, the resulting states of $r|_{\check{p}_1}$ and $\pi_{hom}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})$ coincide.

One of the particular implications of these comments is that each $t'[\hat{t}_{1,j}]_{\check{p}_1}$ is in $\mathcal{L}(C)$. It is also clear that each $t'[\hat{t}_{1,j}]_{\check{p}_1}$, for $\hat{t}_{1,j} \neq t_1$, is not an instance of $t|c$, since the terms at the positions $\check{p}_1$ and $\check{p}_2$ differ. Thus, in order to reach a contradiction, it suffices to prove that, for some j satisfying $\hat{t}_{1,j} \neq t_1$, $t'[\hat{t}_{1,j}]_{\check{p}_1}$ is not an instance of $S - \{t|c\}$. To conclude, we will prove that, only for a finite number of j's, the terms $t'[\hat{t}_{1,j}]_{\check{p}_1}$ are instances of $S - \{t|c\}$. To this end, it can be shown that, for each $s|d$ in $S - \{t|c\}$, at most a finite number of terms $t'[\hat{t}_{1,j}]_{\check{p}_1} = t[\hat{t}_{1,j}]_{\check{p}_1}[t_1]_{\check{p}_2} \cdots [t_1]_{\check{p}_k}[t_{k+1}]_{\check{p}_{k+1}} \cdots [t_n]_{\check{p}_n}$ are instances of $s|d$.

Consider any constrained term $s|d$ in $S - \{t|c\}$. Since t' is not an instance of $s|d$, according to Lemma 7.9, we can distinguish the following cases:

—Assume that there is a position p in $Pos_\Sigma(s) \cap Pos(t')$ such that $root(s|_p)$ is different from $root(t'|_p)$. We distinguish the following cases:
  —Suppose first that p is of the form $\check{p}_1.p'.p''$ for some positions p', p'' such that $|p'| = h(C \cap \overline{A_{hom}}) + 1 > h(A_{hom})$. By Invariant I3, it holds that $s|_{\check{p}_1.p'}$ is a term in $\mathcal{T}(\Sigma)$, that is, without any symbol in Q. Note that $t'|_{\check{p}_1.p'}$ is a term different from $s|_{\check{p}_1.p'}$ because they differ at the symbol located at their relative position p''. Recall that no replaced position in $\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}}$ is a proper prefix of p'. Thus, by Lemma 5.6, at most one term $\hat{t}_{1,j}$ makes $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{\check{p}_1.p'}$ equal to $s|_{\check{p}_1.p'}$, and we are done.
  —Second, suppose p is not of the form $\check{p}_1.p'.p''$ for some positions p', p'' such that $|p'| = h(C \cap \overline{A_{hom}}) + 1$. In this case, $root(t'|_p)$ coincides with all $root(t'[\hat{t}_{1,j}]_{\check{p}_1}|_p)$, and thus, it differs from $root(s|_p)$. Hence, none of these terms is an instance of $s|d$, and we are done.
—Assume that there is a position p in $Pos_Q(s)$ such that $t'|_p$ is not in $\mathcal{L}(A, s|_p)$. Let q be $s|_p$ and let S_p be the resulting state of $\bar{r}|_p$. By the selection of $\bar{r}$, it holds that q is in S_p.

  Let us fix a $j \geq 1$ and suppose that $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}$ is a run of $\overline{A_{hom}}$. Note that, by Invariant I3, either $\check{p}_1$ is not a prefix of p, or p is of the form $\check{p}_1.p'$ for some p' satisfying $|p'| \leq h(A_{hom}) \leq h(C \cap \overline{A_{hom}})$. In the second case, recall that, for such position p', $root(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}|_{p'})$ coincides with $root(\hat{r}_1|_{p'})$. Thus, in any case, the resulting state of $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}|_p$ is S_p. By Lemma 4.9, it follows that $t'[\hat{t}_{1,j}]_{\check{p}_1}$ is not an instance of $s|d$.

  Hence, in order to conclude, it suffices to argue that the terms $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}$ are runs of $\overline{A_{hom}}$ except for a finite number j's.

  Note that $\bar{r}$ and each term $\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})$ is a run of $\overline{A_{hom}}$, and $root(\bar{r}|_{\check{p}_1}) = root(\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}))$. Thus, if, for a concrete j, $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}$ is not a run of $\overline{A_{hom}}$, there must exist a position p'' satisfying the following assumptions:

  —$p'' < \check{p}_1$.
  —$root(\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1})|_{p''}$, that is, $root(\bar{r}|_{p''})$, is a rule $f(q_1, \ldots, q_m) \xrightarrow{e} q'$ where e contains a disequality $p_1 \neq p_2$ such that $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{p''.p_1}$ is equal to $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{p''.p_2}$.

  Since $\bar{r}$ is a run of $\overline{A_{hom}}$, $t'|_{p''.p_1}$ is different from $t'|_{p''.p_2}$. Recall that, all replaced positions in $\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}}$ have length greater than $h(C \cap \overline{A_{hom}}) \geq h(\overline{A_{hom}})$. Thus, $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{p''.p_1}$ and $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{p''.p_2}$ can be obtained from $t'|_{p''.p_1}$ and $t'|_{p''.p_2}$, respectively, by replacing $t'|_{\check{p}_1.\bar{p}}$ by $\pi_\Sigma(\hat{r}_{1,j}))$ at some positions that are independent from j. By Lemma 5.6, only one term can satisfy this statement for such p'' and $p_1 \neq p_2$.

  The selections for p'' and $p_1 \neq p_2$ are finitely bounded. Thus, at most for a finite number of j's, $\bar{r}[\pi_{\neq}(\hat{r}_1[[\hat{r}_{1,j}]]_{\bar{p}})]_{\check{p}_1}$ is not a run, and we are done.

—Finally, assume that there is an equality $p_1 = p_2$ in d such that $t'|_{p_1}$ is different from $t'|_{p_2}$. Note that both $s|_{p_1}$ and $s|_{p_2}$ are identical and in Q. Hence, by Invariant I3, p_1 and p_2 are not of the form $\check{p}_1.p'.p''$ for some positions p', p'' such that $|p'| = h(C \cap \overline{A_{hom}}) + 1 \geq h(A_{hom})$. Note also that, in the case where $\check{p}_1$ is a prefix of p_1 (p_2), no replaced position in $\hat{r}_1[[\hat{r}_{1,1}]]_{\bar{p}}$ is a prefix of $p_1 - \check{p}_1$ ($p_2 - \check{p}_1$). Thus, for each $j \geq 1$, $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{p_1}$ and $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{p_2}$ can be obtained from $t'|_{p_1}$ and $t'|_{p_2}$, respectively, by replacing $t'|_{\check{p}_1.\bar{p}}$ by $\pi_\Sigma(\hat{r}_{1,j})$ at some positions that are independent from j. By Lemma 5.6, at most one term $\hat{t}_{1,j}$ makes $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{p_1}$ equal to $t'[\hat{t}_{1,j}]_{\check{p}_1}|_{p_2}$. Hence, for all the remaining j's, $t'[\hat{t}_{1,j}]_{\check{p}_1}$ is not an instance of $s|d$, due to the same reason as t', and we are done.

7.5. Complexity Analysis

We analyze the complexity of deciding HOM. We do it directly on the size of A_{hom}, which is polynomially bounded on the sizes of the given A and H. According to the discussion of Sections 3.1 and 4.1, $N(A_{hom} \cap \overline{A_{hom}}) \leq N(A_{hom}) + N_{Pos}(A_{hom}) \cdot N(A_{hom})$, $N_{Pos}(A_{hom} \cap \overline{A_{hom}}) = N_{Pos}(A_{hom})$ and $h(A_{hom} \cap \overline{A_{hom}}) \leq 2 \cdot h(A_{hom})$ hold. Moreover, the size of $A_{hom} \cap \overline{A_{hom}}$ is bounded by $2^{\mathcal{P}_1(|A_{hom}|)}$ for some polynomial $\mathcal{P}_1$. Thus, according to the discussion of Section 5.3, $\tilde{h} = \check{h}(A_{hom} \cap \overline{A_{hom}})$ is bounded by $2^{\mathcal{P}_2(|A_{hom}|)}$ for some polynomial $\mathcal{P}_2$.

In order to test whether $\mathcal{L}(linearize(A_{hom}, \tilde{h}))$ includes $\mathcal{L}(A_{hom})$, we start by constructing a TA A_{lin} recognizing the linearization. To this end, for each term t with height bounded by $\tilde{h}$, we need a state q_t reachable by a run on t. Note that, since $\tilde{h}$ is exponential on $|A|$, the total number of such t's is triple exponential on $|A|$. Thus, the size of A_{lin} is triple exponential on $|A|$. Next, we have to complement A_{lin}. Usually, complementing a TA adds one level of exponentiation, because as a first step one has to determinize the TA producing sets of states as new states. But in our case this is not going to happen because, for each two different terms t, t' with height bounded by $\tilde{h}$, the states $q_t, q_{t'}$ are not ~~commonly~~ jointly reachable, and hence, they will not be part of the same state/set. In consequence, it can be constructed a TA $\bar{A}_{lin}$ recognizing $\overline{\mathcal{L}(linearize(A_{hom}, \tilde{h}))}$ and whose size is triple exponential on $|A_{hom}|$. Testing whether $\overline{\mathcal{L}(linearize(A_{hom}, \tilde{h}))} \cap \mathcal{L}(A_{hom})$ is empty, can be decided in triple exponential time on $|A_{hom}|$ by a simple fix-point computation ~~algorithm~~ applied on the intersection of A_{hom} and $\bar{A}_{lin}$. It suffices to iteratively compute the states q for which there exists a run reaching q, and check whether some of such q is accepting.

8. CONCLUSION

We have closed affirmatively the open question of the decidability of the HOM problem. The time and space complexity is a tower of three exponentials with respect to the size of the input. Comon and Jacquemard [1997] present a propagation technique for proving that emptiness for $TA_{\neq}$can be decided in exponential time. It could be interesting to study the applicability of this technique to the case of $TA_{\neq,hom}$ in order to improve the time complexity.

Our result implies decidability of other related questions. For example, Fülöp [1994] questions the decidability of the regularity of the range of a given tree homomorphism. This is a particular case of the HOM problem, and hence, we also close this question affirmatively. On the other hand, tree homomorphisms are a particular case of bottom-up tree transducers. Engelfriet [1975] showed that for any given regular language L and any given bottom-up tree transducer T, a regular language L' and a tree homomorphism H can be computed such that $T(L) = H(L')$ holds. Thus, given a regular language L and a bottom-up tree transducer T, it is decidable whether $T(L)$ is regular. This is in contrast with top-down tree transducers: tree homomorphisms are also a particular case of top-down tree transducer, but it is well known [Fülöp 1994] that the regularity of the image of a given regular language through a top-down tree transducer is undecidable. On the other hand, top-down tree transducers with just one state are, in fact, tree homomorphisms, so our results imply decidability of regularity for images of regular languages through top-down tree transducers with just one state.

We have introduced the new class $\mathrm{TA}_{\neq,hom}$. This class of automata is at the same time *meaningful* (it contains the images of regular languages through tree homomorphisms) and *tractable* (many properties are decidable for automata of this class). It would be interesting to know whether $\mathrm{TA}_{\neq,hom}$ could still be further extended in a meaningful way while still keeping at the same time its tractability. Also, according to Section 1.3, we believe that the study of $\mathrm{TA}_{\neq,hom}$ may provide further insights into other topics, like term rewriting and typechecking.

ACKNOWLEDGMENTS

We greatly appreciate the comments of Z. Fülöp, F. Jacquemard, S. Maneth, and Carles Creus.

REFERENCES

Baader, F. and Nipkow, T. 1998. *Term Rewriting and All That*. Cambridge University Press, New York.

Bogaert, B., Seynhaeve, F., and Tison, S. 1999. The recognizability problem for tree automata with comparison between brothers. In *Proceedings of the Symposium on Foundations of Software Science and Computation Structures (FOSSACS)*. 150–164.

Bogaert, B. and Tison, S. 1992. Equality and disequality constraints on direct subterms in tree automata. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS)*. 161–171.

Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., and Tommasi, M. 2007. Tree automata techniques and applications. http://www.grappa.univ-lille3.fr/tata.

Comon, H. and Jacquemard, F. 1994. Ground reducibility and automata with disequality constraints. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS)*. 151–162.

Comon, H. and Jacquemard, F. 1997. Ground reducibility is EXPTIME-complete. In *Proceedings of the Symposium on Logic in Computer Science (LICS)*. 26–34.

Dauchet, M., Caron, A.-C., and Coquidé, J.-L. 1995. Automata for reduction properties solving. *J. Symb. Comput. 20,* 2, 215–233.

Dauchet, M., Tison, S., and Tommasi, M. 2002. Reduction de la non-linearite des morphismes d'arbres recognizable tree-languages and non-linear morphisms. *Theoret. Comput. Sci. 281,* 1–2, 219–233.

Engelfriet, J. 1975. Bottom-up and top-down tree transformations—a comparison. *Math. Syst. Theory 9,* 3, 198–231.

Fülöp, Z. 1994. Undecidable properties of deterministic top-down tree transducers. *Theoret. Comput. Sci. 134*, 311–328.

Gascón, A., Godoy, G., and Jacquemard, F. 2009. Closure of tree automata languages under innermost rewriting. *Electronic Notes Theoret. Comput. Sci. 237*, 23–38.

Gécseg, F. and Steinby, M. 1984. *Tree Automata*. Akadémiai Kiadó.

Gécseg, F. and Steinby, M. 1997. Tree languages. In *Handbook of Formal Languages*, G. Rozenberg and A. Salomaa, Eds., Vol. 3, Springer-Verlag, 1–68.

Gilleron, R. 1991. Decision problems for term rewriting systems and recognizable tree languages. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS)*. 148–159.

Gilleron, R. and Tison, S. 1995. Regular tree languages and rewrite systems. *Fund. Info. 24,* 1/2, 157–174.

Giménez, O., Godoy, G., and Maneth, S. 2011. Deciding regularity of the set of instances of a set of terms with regular constraints is EXPTIME-complete. *SIAM J. Comput. 40,* 2, 446–464.

Godoy, G., Giménez, O., Ramos, L., and Àlvarez, C. 2010. The HOM problem is decidable. In *Proceedings of the Symposium on Theory of Computing (STOC)*. 485–494.

Godoy, G. and Huntingford, E. 2007. Innermost-reachability and innermost-joinability are decidable for shallow term rewrite systems. In *Proceedings of the Symposium on Term Rewriting and Applications (RTA)*. 184–199.

GODOY, G., MANETH, S., AND TISON, S. 2008. Classes of tree homomorphisms with decidable preservation of regularity. In *Proceedings of the Symposium on Foundations of Software Science and Computation Structures (FOSSACS)*. 127–141.

HOFBAUER, D. AND HUBER, M. 1992. Computing linearizations using test sets. In *Proceedings of the 3rd International Workshop on Conditional Term Rewriting Systems (CTRS)*. 287–301.

KUCHEROV, G. AND TAJINE, M. 1995. Decidability of regularity and related properties of ground normal form languages. *Info. Compuat. 118*, 91–100.

RTA-LOOP. 1991. International Conference on Rewriting Techniques and Applications. The list of open problems. http://www.lri.fr/∼rtaloop/.

SCHWENTICK, T. 2007. Automata for XML - a survey. *J. Comput. Syst. Sci. 73,* 3, 289–315.

SUCIU, D. 2002. The XML typechecking problem. *SIGMOD Record 31,* 1, 89–96.

THATCHER, J. W. 1969. Transformations and translations from the point of view of generalized finite automata theory. In *Proceedings of the Symposium on Theory of Computing (STOC)*, 129–142.

VÁGVÖLGYI, S. 2009. On ranges of bottom-up tree transducers. In *Automata, Formal Languages, and Related Topics*. 129–143.

VÁGVÖLGYI, S. AND GILLERON, R. 1992. For a rewrite system it is decidable whether the set of irreducible, ground terms is regognizable. *Bull. EATCS 48*, 197–209.

Received September 2010; revised January 2012 and December 2012; accepted June 2013