Math. Struct. in Comp. Science (2018), *vol.* 28, *pp.* 1786–1846. © Cambridge University Press 2017
doi:10.1017/S096012951700024X First published online 6 December 2017

Paths, tree homomorphisms and disequalities for $\mathcal{H}_1$-clauses

ANDREAS REUß* and HELMUT SEIDL

Institut für Informatik I2, Technische Universität München, Boltzmannstraße 3, D-85748 Garching, Germany
Emails: a.reuss@in.tum.de, seidl@in.tum.de

Received 10 October 2013; revised 11 October 2017

It is well known that satisfiability is decidable for Horn clauses of the class $\mathcal{H}_1$. Since arbitrary Horn clauses can naturally be approximated by $\mathcal{H}_1$-clauses, $\mathcal{H}_1$ can be used for realizing any program analysis which can be specified by means of Horn clauses. Recently, we have shown that decidability for Horn clauses from $\mathcal{H}_1$ is retained if the clauses are either extended with tests for disequality between subterms identified by paths or for disequality between homomorphic images of terms. These two results refer to orthogonal extensions of $\mathcal{H}_1$-clauses. Here, we provide a generalization of both results. For that, we introduce hom-path disequalities and show that for each finite set of $\mathcal{H}_1$-clauses extended with such tests an equivalent tree automaton with hom-path disequalities can be constructed. Since emptiness for that class of automata has been shown decidable by Godoy et al. in 2010, we conclude that satisfiability is decidable for $\mathcal{H}_1$-clauses with hom-path disequalities.

1. Introduction

Analyses of tree-manipulating programs may conveniently be specified by means of Horn clauses. This technique has been applied, e.g., to the analysis of cryptographic protocols (Blanchet 2001; Frühwirth *et al.* 1991), and also to the implementation of these in real-world programming languages (Goubault-Larrecq and Parrennes 2005). Since satisfiability of finite sets of Horn clauses in general is undecidable, there is considerable interest in useful decidable subclasses of Horn clauses. In Weidenbach (1999), a large class of clauses is identified for which satisfiability is decidable. This class has been called $\mathcal{H}_1$ in Nielson *et al.* (2002). The class $\mathcal{H}_1$ differs from general Horn clauses in that the terms in the heads contain at most one constructor and no variable occurs twice in any head. Finite sets of $\mathcal{H}_1$-clauses can be transformed to equivalent finite bottom-up tree automata. On the other hand, $\mathcal{H}_1$ can abstract in a natural way arbitrary finite sets of Horn clauses. Therefore, $\mathcal{H}_1$ is particularly useful for approximative analyses of term-manipulating programs (Frühwirth *et al.* 1991).

Pure Horn clauses, though, have no built-in mechanism that allows one to express negative conditions such as that two values must be different. In order to compensate for this deficiency, we have extended Horn clauses with disequality constraints. It turns out that $\mathcal{H}_1$ remains decidable when the clauses are extended with various kinds of

* This work was partly supported by the DFG Graduiertenkolleg 1480 (PUMA).

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

disequalities. In Seidl and Reuß (2011), we considered disequalities between terms. This basic setting has been extended to disequalities between subterms identified by paths (Seidl and Reuß 2012) or disequalities between homomorphic images of terms (Reuß and Seidl 2012). In each of these cases, every finite conjunction of $\mathcal{H}_1$-clauses with disequalities can be normalized to a finite conjunction of *normal clauses* with such disequalities. The latter in turn can be transformed to an equivalent finite set of automaton clauses which may be considered as a bottom-up tree automaton with disequality constraints.

Disequalities between terms as considered in Seidl and Reuß (2011) can be used, for instance, to express properties of cryptographic protocols like freshness of keys or nonces. Disequalities of terms, though, may not be expressive enough for expressing more involved properties. Consider, e.g., a voting protocol where we want to check whether a person has already voted (Baskar *et al.* 2007; Fujioka *et al.* 1993). Then we must not compare the submitted vote as a whole to the list of already submitted votes. Instead we must ensure that the voter has not submitted *any* vote so far. As *tree homomorphisms* allow one to abstract from irrelevant information, they can be used for that purpose.

Example 1.1. Assume that we are given a list l of votes where each element of l is of the form $\mathrm{vote}(p, v)$ for a constructor symbol vote, a person p who submits the vote, and his or her vote v. The next vote $\mathrm{vote}(p', v')$ then should not only be not contained in the list l but should differ from all elements of l in the first argument. This can be expressed by the predicate valid as defined by

$$\begin{array}{lcl} \mathrm{valid}(X) & \Leftarrow & q(X, [\,]) \\ q(X, Z) & \Leftarrow & q(X, ::(Y, Z)),\ [X]_{\mathrm{H}} \neq [Y]_{\mathrm{H}} \\ q(X, Y) & \Leftarrow & \mathrm{some_vote}(X), \mathrm{votes}(Y) \end{array}$$

Here, the upper case letters represent variables, [] denotes an empty list and :: is the list constructor. The expression $[X]_{\mathrm{H}}$ denotes the image of the value of X under the tree homomorphism H. We define H such that terms $\mathrm{vote}(t_1, t_2)$ are mapped to the homomorphic image of t_1, while all other constructors are preserved.

The second clause involves a comparison of the new vote, identified by X, and the first list entry, Y, which yields one of the already submitted votes. Thanks to H, the compared values are in fact the first subtrees under the root constructor vote, i.e., the person who wants to vote and one of the persons who have already voted. The definition of the predicate q in the second clause thus ensures that $q(t, [\,])$ only holds if $q(t, l)$ holds for a list l where the voter of t is not contained in any vote of l.

The disadvantage of an interpretations modulo tree homomorphisms is that each occurrence of the same constructor is treated identically. In situations where this is not desired, pieces of information may still be identified by means of access paths. The simplest example where homomorphisms do not help in an obvious way is when two successors of a node must be compared for disequality, as in the constraint $X.(f, 1) \neq X.(f, 2)$. Here, the variable X denotes the node whose successors are compared, and $(f, 1)$ and $(f, 2)$ are the labelled paths to the first and second child below the constructor f, respectively. As an extension for finite bottom-up tree automata, disequalities of subterms identified by paths

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

are strictly more expressive than disequalities of terms, but incomparable to disequalities modulo homomorphisms.

In the present paper, we consider disequalities which combine the concepts of identifying subterms by means of paths and interpretation modulo homomorphisms by introducing *hom-path disequalities*, i.e., disequalities of paths applied to homomorphic images of terms. We then show that $\mathcal{H}_1$-clauses with hom-path disequalities can still be normalized into an equivalent tree automaton with such disequalities. Since it is shown in Godoy *et al.* (2010) that emptiness for this class of automata is decidable, we conclude that satisfiability remains decidable even for finite sets of $\mathcal{H}_1$-clauses with hom-path disequalities.

As path disequalities and hom-disequalities yield incomparable extensions of tree automata (and finite sets of $\mathcal{H}_1$-clauses, respectively), hom-path disequalities constitute a proper generalization of both these concepts, i.e., $\mathcal{H}_1$-clauses with hom-path disequalities are strictly more expressive than path or hom-disequalities alone.

Outline. In Section 2, we review basic notations and concepts such as paths, tree homomorphisms and useful subclasses of Horn clauses with disequality constraints. Section 3 compares different classes of tree automata with constraints w.r.t. their expressiveness. Section 4 introduces the normalization framework for $\mathcal{H}_1$-clauses with constraints, instantiated with hom-disequalities. Section 5 shows how to extend the procedure for hom-path disequalities.

2. Preliminaries

This section is organized as follows. In Section 2.1, we introduce basic notions of terms, trees, *paths* as subterm identifiers and variable substitutions. Section 2.2 introduces tree homomorphisms. In Section 2.3, various kinds of disequality constraints are defined. Section 2.4 defines the class $\mathcal{H}_1$ of Horn clauses, useful subclasses of $\mathcal{H}_1$ and their respective extensions with disequalities. Sections 2.3 and 2.4 also give some basic lemmas on constraints and Horn clauses with constraints, respectively.

2.1. *Terms and trees*

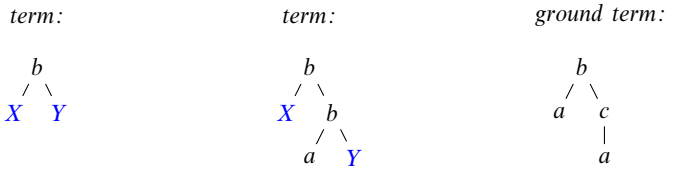
First-order Horn clauses define sets of trees. Here we only consider Horn clauses with unary predicates; non-unary predicates, though, can also be integrated in our framework by equipping their arguments with an *implicit* constructor of the same arity as the predicate. The trees are made up of symbols from a ranked alphabet (Σ, ar), where Σ denotes a finite set of *constructors* and $\mathrm{ar} : \Sigma \to \mathbb{N} \cup \{0\}$ specifies each constructor's arity. If the arities of symbols are understood, then the ranked alphabet is denoted by Σ alone. Nullary constructors are also called atoms. In order to avoid trivial cases, we assume that there is at least one atom and one constructor of arity greater than 0. For a ranked alphabet Σ and a set $\mathcal{X} = \{X_1, X_2, \ldots\}$ of *variables*, the set $\mathcal{T}_\Sigma(\mathcal{X})$ of (finite ordered ranked) trees over Σ and $\mathcal{X}$ consists of all terms t defined by the grammar:

$$t \quad ::= \quad X_i \mid b(t_1, \ldots, t_k),$$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

where $b \in \Sigma$ has arity $k \geqslant 0$. In case that $k = 0$, we call b also *atom* and abbreviate $b()$ with b. The *depth* of a term t is given by 0 if t is a variable, and by 1 plus the maximum depth of the terms t_i if $t = f(t_1, \ldots, t_k), k \geqslant 0$ – where in case $k = 0$, i.e., when t is an atom, the depth is 1. A tree t is called *ground* if it contains no variables. The set of ground terms (or ground trees) is denoted $\mathcal{T}_\Sigma$. Note that $\mathcal{T}_\Sigma$ is infinite according to our assumption on Σ. A (positive) *literal* A is an expression of the form $p(t)$, where p is a unary predicate and $t \in \mathcal{T}_\Sigma(\mathcal{X})$.

Example 2.1. The following figure shows two terms containing variables, and a variable-free term (or tree) over the alphabet $\Sigma = \{a, b, c\}$ with arities 0, 2 and 1, respectively.



Subterms are identified by *paths*. A labelled path π is given by a finite sequence $(f_1, i_1).(f_2, i_2). \ldots .(f_n, i_n)$, where for every j, $f_j \in \Sigma$ and $1 \leqslant i_j \leqslant m_j$ if m_j is the arity of f_j. The empty sequence is denoted ϵ. An expression $X.\pi$ for a variable X and a path π is called *path expression*. We also write X instead of $X.\epsilon$.

General terms, as a generalization of both ordinary terms and path expressions, are terms which may contain path expressions $X.\pi$ instead of variables X. For a general term t, the sub-term at path π, denoted by $t.\pi$, is recursively defined by

— $t.\epsilon = t$;
— $(Y.\pi_1).\pi_2 = Y.(\pi_1.\pi_2)$;
— $t.(f, i).\pi' = t_i.\pi'$ if $t = f(t_1, \ldots, t_m)$ and $1 \leqslant i \leqslant m$. In particular, for $g \neq f$, $t.(g, i).\pi'$ is *undefined*.

Example 2.2. For an atom a and a binary constructor b, the following expressions t_i with

$$t_1 = a \qquad t_2 = b(b(a, a), X) \qquad t_3 = b(a, X.(b, 1).(b, 2))$$

all are general terms. We have, e.g.,

$$\begin{array}{lcllcl} t_2.(b,1).(b,1) & = & a & t_3.(b,2) & = & X.(b,1).(b,2) \\ t_2.(b,2).(b,2) & = & X.(b,2) & t_3.(b,2).(b,1) & = & X.(b,1).(b,2).(b,1) \end{array}$$

A *substitution* θ is a mapping $\theta : \mathcal{X} \to \mathcal{T}_\Sigma(\mathcal{X})$. The mapping may either be written as $\theta = [t_{i_1}/X_{i_1}, \ldots, t_{i_j}/X_{i_j}]$ or as $\theta = \{X_{i_1} \mapsto t_{i_1}, \ldots, X_{i_j} \mapsto t_{i_j}\}$. Then, $X_i\theta = X_i$ for all variables X_i that are not mentioned. We write $e\theta$ for the result of applying θ to the expression e. For a variable X, we also denote by $\theta(X)$ the term with which X is replaced according to θ. θ is called *ground* w.r.t. an expression e if $\theta(X_i)$ is ground for all variables X_i occurring

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

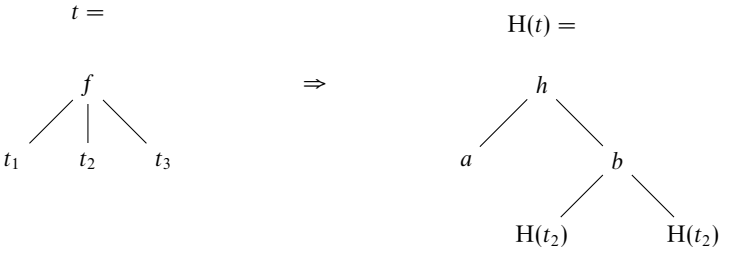
in e. The expression $e\theta$ is obtained from e by replacing each occurrence of X_i by the term $\theta(X_i)$ defined through θ. We write $\theta \oplus \{X_i \mapsto t\}$ for the substitution which maps X_i to t and all variables X_j different from X_i to $\theta(X_j)$.

2.2. *Tree homomorphisms*

A tree homomorphism H maps each symbol $f \in \Sigma$ of arity $k \geqslant 0$ to a *hom-term* t_f over constructors from some alphabet Δ and new *hom-variables* from the set $\{[X_1]_\mathrm{H}, \ldots, [X_k]_\mathrm{H}\}$. Here, the expression $[X_i]_\mathrm{H}$ is meant to denote the homomorphic image under H of the value of X_i. Thus, for a ground substitution θ, the expression $([X]_\mathrm{H})\theta$ evaluates to $\mathrm{H}(X\theta)$. We write He or H(e) when applying H to the expression e. For a term $s = f(t_1, \ldots, t_k)$ from $\mathcal{T}_\Sigma(\mathcal{X})$, the mapping $\mathrm{H} : \mathcal{T}_\Sigma(\mathcal{X}) \mapsto \mathcal{T}_\Delta([\mathcal{X}]_\mathrm{H})$ then is recursively defined as $\mathrm{H}(X) = [X]_\mathrm{H}$ for $X \in \mathcal{X}$ and $\mathrm{H}(f(t_1, \ldots, t_k)) = t_f[\mathrm{H}(t_1)/[X_1]_\mathrm{H}, \ldots, \mathrm{H}(t_k)/[X_k]_\mathrm{H}]$. For ground terms, tree homomorphisms can be seen as second-order substitutions of function symbols.

Here, we consider only the case that $\Delta = \Sigma$. Note that this is without loss of generality – at least for all applications where it is possible to enlarge the underlying alphabet. Note that H(s) is ground whenever s is a ground term and that the expression $\mathrm{H}(f(X_1, \ldots, X_k))$ evaluates to the term t defined by H(f). The function H^{-1} reverses the effect of applying H, i.e., $\mathrm{H}^{-1}s = \{t \mid \mathrm{H}t = s\}$. H and H^{-1} are extended to sets of terms T by defining $\mathrm{H}T = \{\mathrm{H}t \mid t \in T\}$ and $\mathrm{H}^{-1}T = \{t \mid \mathrm{H}t \in T\}$.

Example 2.3. Let H be a tree homomorphism with $\mathrm{H}(f) = h(a, b([X_2]_\mathrm{H}, [X_2]_\mathrm{H}))$, and let $t = f(t_1, t_2, t_3)$ with $t_i \in \mathcal{T}_\Sigma(\mathcal{X})$. The term t is then recursively transformed by H according to the following picture:



Tree homomorphisms are rather expressive. They may, e.g.,

— rename constructors: $\mathrm{H}(g(X_1)) = h([X_1]_\mathrm{H})$,
— delete constructors: $\mathrm{H}(g(X_1)) = [X_1]_\mathrm{H}$,
— delete subtrees: $\mathrm{H}(f(X_1, X_2)) = g([X_2]_\mathrm{H})$,
— add constructors: $\mathrm{H}(f(X_1, X_2)) = f(g([X_1]_\mathrm{H}), h(g([X_2]_\mathrm{H})))$,
— copy subtrees: $\mathrm{H}(g(X_1)) = f([X_1]_\mathrm{H}, [X_1]_\mathrm{H})$,
— permute subtrees: $\mathrm{H}(f(X_1, X_2)) = f([X_2]_\mathrm{H}, [X_1]_\mathrm{H})$,
— combine two or more of these features.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Hom-paths are expressions $[X]_{\mathrm{H}}.\pi$, where $[X]_{\mathrm{H}}$ is a hom-variable and π is a path. *General hom-terms*, as an extension of both hom-terms and general terms (where H would be the identity), are terms containing hom-paths instead of variables. For a (not necessarily ground) term t, $[t]_{\mathrm{H}}.\pi$ evaluates to $(\mathrm{H}t).\pi$. Hence, $[t]_{\mathrm{H}}.\pi$ evaluates to a general hom-term (or to *undefined*).

Example 2.4. Let $\Sigma = \{a, g, h\}$ where the symbols have arities 0, 1 and 1, respectively, and let $\mathrm{H} = \{a \mapsto a, g \mapsto h([X_1]_{\mathrm{H}}), h \mapsto g([X_1]_{\mathrm{H}})\}$, i.e., H exchanges the constructors g and h. Then $[g(h(X))]_{\mathrm{H}}.(g, 1)$ is undefined, while $[g(h(X))]_{\mathrm{H}}.(h, 1) = g([X]_{\mathrm{H}})$ and $[g(X)]_{\mathrm{H}}.(h, 1).(g, 1) = [X]_{\mathrm{H}}.(g, 1)$.

2.3. *Constraints*

We extend the pre-conditions of clauses by additional conditions or constraints. The constraint ϕ of a clause is a positive Boolean combination of atomic disequalities. Constraints without disjunctions, though, are of particular interest. These are called *conjunctive*. We consider constraints using different kinds of atomic disequalities. A disequality $s \neq t$ is

— a *term disequality* if s, t are terms;
— a *path disequality* if s is a path expression $X_i.\pi$ and t is a path expression $X_j.\pi'$ or a ground term;
— a *general term disequality* if s, t are general terms;
— a *hom-disequality* if $s, t \in \mathcal{T}_\Sigma([\mathcal{X}]_{\mathrm{H}})$, i.e., s and t are hom-terms;
— a *hom-path disequality* if s is a hom-path $[X]_{\mathrm{H}}.\pi$ and t is a hom-path $[X']_{\mathrm{H}}.\pi'$ or a ground term;
— a *general hom-term disequality* if s, t are general hom-terms.

Consider a general term t and a ground substitution θ which provides ground terms for all variables occurring in t. If $(X\theta).\pi$ is defined for each path expression $X.\pi$ occurring in t, then $t\theta$ is obtained from t by replacing each occurrence of $X.\pi$ with the ground term $(X\theta).\pi$. Otherwise, $t\theta$ is undefined.

This definition is extended to general hom-terms by additionally taking the homomorphism into account. Consider a general hom-term t and a ground substitution θ. If $(\mathrm{H}(\theta(X))).\pi$ is defined for each hom-path $[X]_{\mathrm{H}}.\pi$ occurring in t, then $t\theta$ is obtained from t by replacing each occurrence of $[X]_{\mathrm{H}}.\pi$ by the ground term $(\mathrm{H}(\theta(X))).\pi$. Otherwise, $t\theta$ is undefined.

The ground substitution θ *satisfies* the atomic constraint $l \neq r$, denoted by $\theta \models (l \neq r)$, if either $l\theta$ or $r\theta$ is undefined, or both are defined and $l\theta \neq r\theta$. The disequality $l \neq r$ is *satisfiable* if there is a ground substitution θ with $\theta \models (l \neq r)$. Note that for hom-terms, our notion of disequalities is more general than the notion of hom-disequality used in Reuß and Seidl (2012) as it applies the homomorphism to the variables in the terms, but leaves the symbols in the compared terms unchanged. For convenience, we also define for each disequality $l \neq r$ the corresponding equality $l = r$ as its logical negation. This means that the substitution θ satisfies $l = r$ (denoted by $\theta \models l = r$) if $l\theta$ and $r\theta$ are defined and $l\theta = r\theta$.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Satisfiability is extended to positive Boolean combinations in the usual way, i.e.,

$$\theta \models (\phi_1 \wedge \phi_2) \quad \text{iff} \quad (\theta \models \phi_1) \wedge (\theta \models \phi_2),$$
$$\theta \models (\phi_1 \vee \phi_2) \quad \text{iff} \quad (\theta \models \phi_1) \vee (\theta \models \phi_2).$$

Consider constraints ϕ_1, ϕ_2 which may contain occurrences of hom-paths $[X_i]_{\mathrm{H}}.\pi$. Then, $\phi_1 \Rightarrow \phi_2$ iff for all ground substitutions θ of the variables occurring in ϕ_1, ϕ_2, $\theta \models \phi_1$ implies that $\theta \models \phi_2$. The latter property may be non-trivial to decide. Instead, we therefore introduce an *abstract* notion of implication which is clearly decidable and implies ordinary implication. Let $[X_{i_1}]_{\mathrm{H}}.\pi_1, \ldots, [X_{i_r}]_{\mathrm{H}}.\pi_r$ be the sequence of distinct hom-paths occurring in ϕ_1, ϕ_2. Let E denote the set of all indices l where $\pi_l = \epsilon$. For a subset $E \subseteq I \subseteq \{1, \ldots, r\}$ and $i = 1, 2$, define $\phi_{i,I}$ as the constraint which is obtained from ϕ_i by replacing each atomic constraint in ϕ_i which mentions $[X_{i_l}]_{\mathrm{H}}.\pi_l$ for some $l \notin I$ with *true*. Let $Y_l, l \in I$ denote a sequence of distinct fresh variables and $\bar{\phi}_{1,I}, \bar{\phi}_{2,I}$ denote the term disequality constraints obtained from $\phi_{1,I}, \phi_{2,I}$ by replacing $[X_{i_l}]_{\mathrm{H}}.\pi_l$ with Y_l for all $l \in I$. Then we say that ϕ_1 *abstractly* implies ϕ_2, $\phi_1 \Rightarrow^\sharp \phi_2$, iff $\bar{\phi}_{1,I} \implies \bar{\phi}_{2,I}$ for all $E \subseteq I \subseteq \{1, \ldots, r\}$.

Intuitively, the abstraction gets rid of two phenomena which render the implication difficult. We get rid of the homomorphism H, and we get rid of the paths π_l. The naive idea to arrive at an implication between plain term constraints is to just replace each occurring hom-path $[X_{i_l}]_{\mathrm{H}}.\pi_l$ with a distinct variable Y_l. This idea is sufficient as long as for every substitution θ and every occurring hom-path $[X_{i_l}]_{\mathrm{H}}.\pi_l$, $\mathrm{H}(X_{i_l}\theta).\pi_l$ is defined. If this were not the case, some atomic constraint in ϕ_1, ϕ_2 could be vacuously satisfied for some substitution – while it would no longer be satisfied by a corresponding substitution of the Y_l. In order to take care of such cases, we consider all possible subsets I of hom-paths $[X_{i_l}]_{\mathrm{H}}.\pi_l$, where $\mathrm{H}(X_{i_l}\theta).\pi_l$ is defined for a given substitution θ. The following example illustrates the difference between (concrete) implication and abstract implication.

Example 2.5. Consider the homomorphism H which maps each symbol to the ground term $g(a)$. Let $\phi_1 = \mathit{true}$ and $\phi_2 = [X_1]_{\mathrm{H}}.(f, 2) \neq [X_1]_{\mathrm{H}}.(f, 2)$. Since $g(a).(f, 2)$ is undefined, $\theta \models \phi_2$ for every substitution $\theta = \{X_1 \mapsto t\}$, $t \in \mathcal{T}_\Sigma$. Therefore, $\phi_1 \Rightarrow \phi_2$ holds.

Now consider the sequence of hom-paths occurring in ϕ_1, ϕ_2. This sequence just consists of $[X_1]_{\mathrm{H}}.(f, 2)$. There are two possibilities for I, namely, $\varnothing$ and $\{1\}$. For the latter case, we have $\phi_{2,I} = (Y_1 \neq Y_1)$ – which is unsatisfiable. Therefore, $\phi_{1,I} \Rightarrow \phi_{2,I}$ does not hold. Accordingly, ϕ_1 does not abstractly imply ϕ_2.

Still, we remark that, for plain term constraints (no homomorphism, no paths), the notions of abstract implication and concrete implication coincide. Moreover, we have

Lemma 2.1. Let ϕ_1, ϕ_2 be general hom-term disequalities. Then, $\phi_1 \Rightarrow^\sharp \phi_2$ implies that $\phi_1 \Rightarrow \phi_2$ holds, i.e., for every substitution θ of the variables occurring in ϕ_1, ϕ_2, $\theta \models \phi_1$ implies that $\theta \models \phi_2$ holds as well.

Proof. Assume that $\phi_1 \Rightarrow^\sharp \phi_2$ holds. Consider a ground substitution θ of the variables in ϕ_1, ϕ_2 so that $\theta \models \phi_1$. We need to show that then $\theta \models \phi_2$ holds as well. Let $[X_{i_1}]_{\mathrm{H}}.\pi_1, \ldots, [X_{i_r}]_{\mathrm{H}}.\pi_r$ the sequence of distinct hom-paths occurring in ϕ_1, ϕ_2, let I denote the set of all $l \in \{1, \ldots, r\}$ so that $\mathrm{H}(X_{i_l}\theta).\pi_l$ is defined. In particular, I contains each index l where $\pi_l = \epsilon$. By definition, $\theta \models d$ for every atomic constraint d occurring in ϕ_1, ϕ_2

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

which contains a hom-path $[X_{i_l}]_{\mathrm{H}}.\pi_l$ with $l \notin I$. For $i = 1, 2$, we construct the constraint $\phi_{i,I}$ by replacing every atomic constraint in ϕ_i which mentions a hom-path $[X_{i_l}]_{\mathrm{H}}.\pi_l$ for some $l \notin I$ with *true*. Then, $\theta \models \phi_{1,I}$ and $\theta \models \phi_2$ iff $\theta \models \phi_{2,I}$.

Now, we define $\bar{\theta} = \{Y_l \mapsto \mathrm{H}(X_{i_l}\theta).\pi_l \mid l \in I\}$. For $i = 1, 2$, let $\bar{\phi}_i$ denote the term disequality constraint which is obtained from $\phi_{i,I}$ by replacing for all $l \in I$, each occurrence of $[X_{i_l}]_{\mathrm{H}}.\pi_l$ with Y_l. Then, $\bar{\phi}_{1,I}\bar{\theta} = \phi_{1,I}\theta$. Therefore, $\bar{\theta} \models \bar{\phi}_{1,I}$ and thus, due to the abstract implication, also $\bar{\theta} \models \bar{\phi}_2$. Since $\bar{\phi}_2\bar{\theta} = \phi_{2,I}\theta = \phi_2\theta$, we conclude that $\theta \models \phi_2$ – which completes the proof. □

Also, abstract implication turns out to be decidable. In order to show that, we rely on a decision procedure for implications between ordinary term constraints (no homomorphisms, no paths).

In Müller-Olm *et al.* (2005), it is shown that every positive Boolean combination ϕ of term *equalities* can be brought into *canonical* form. This canonical form consists of a disjunction $\psi_1 \vee \ldots \vee \psi_r$ of distinct conjunctions ψ_j of equalities with the following additional properties:

1. The left-hand side of each occurring equality is a variable.
2. Each variable X_i occurring on the left-hand side in an equality of ψ_j does not occur on the right-hand side of any equality in ψ_j.
3. In any occurring equality, $X_i = X_{i'}$, $i < i'$ holds.

Moreover, the sets of equalities occurring in the conjunctions ψ_j and $\psi_{j'}$ are incomparable whenever $j \neq j'$.

In case that ϕ is unsatisfiable, the canonical form is the *empty* disjunction, whereas if ϕ is universally true, the disjunction consists of a single empty conjunction. We have the following:

Lemma 2.2. Assume that ϕ_1, ϕ_2 are positive Boolean combinations of term equalities. Then, ϕ_1 and ϕ_2 are equivalent iff their canonical forms are equal (up to reorderings).

As a corollary, we obtain effective algorithms for deciding implication not only for positive Boolean combinations of term equalities, but also of term disequalities.

Corollary 2.1. Assume that ϕ_1, ϕ_2 are positive Boolean combinations, either both of term equalities or both of term disequalities. Then it is decidable whether $\phi_1 \Rightarrow \phi_2$ holds, i.e., whether for all substitutions θ, $\theta \models \phi_1$ implies $\theta \models \phi_2$.

Proof. Let us consider the case of term equalities first. Then, $\phi_1 \Rightarrow \phi_2$ iff ϕ_1 is equivalent to $\phi_1 \wedge \phi_2$ where the latter can be decided by means of Lemma 2.2.

Now assume that ϕ_1 and ϕ_2 are positive Boolean combinations of term disequalities. Then $\phi_1 \Rightarrow \phi_2$ is equivalent to $\neg\phi_2 \Rightarrow \neg\phi_1$. By De Morgan's laws, $\neg\phi_1$ as well as $\neg\phi_2$ can be re-written to positive Boolean combinations of term equalities only, for which we have just shown decidability of implication. □

Corollary 2.2. For positive Boolean combinations ϕ_1, ϕ_2 of general hom-term disequalities, it is decidable whether $\phi_1 \Rightarrow^{\sharp} \phi_2$ holds.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Proof. Let $[X_{i_l}]_{\mathrm{H}}.\pi_l, l = 1,\ldots,r$ denote the sequence of hom-paths occurring in ϕ_1, ϕ_2. Let E denote the set of all l where $\pi_l = \epsilon$, and consider any subset I with $E \subseteq I \subseteq \{1,\ldots,r\}$. For I, the constraints $\phi_{i,I}$ and $\bar{\phi}_{i,I}$ $(i = 1,2)$ according to the definition of abstract implication, can be effectively constructed. The constraints $\bar{\phi}_{1,I}$ and $\bar{\phi}_{2,I}$ are positive Boolean combinations of term disequalities. Thus, by Corollary 2.1, it is decidable whether or not $\bar{\phi}_{1,I} \Rightarrow \bar{\phi}_{2,I}$ holds. Since there are only finitely many subsets I with $E \subseteq I \subseteq \{1,\ldots,r\}$, the statement follows. □

Concerning expressiveness, it turns out that general term disequalities can be split into (possibly empty) disjunctions of satisfiable path disequalities. Likewise, general hom-term disequalities can be split into disjunctions of hom-path disequalities.

First, we remark that satisfiability of hom-path disequalities is decidable. For that, we first prove the following properties decidable for tree homomorphisms.

Lemma 2.3. Let H be a tree homomorphism. Then the following problems are decidable:

1. Given a path π, is $(\mathrm{H}t).\pi$ defined for all $t \in \mathcal{T}_\Sigma$?
2. Given a path π so that $(\mathrm{H}t).\pi$ defined for all $t \in \mathcal{T}_\Sigma$, is there a ground term s such that $(\mathrm{H}t).\pi = s$ for all $t \in \mathcal{T}_\Sigma$?
3. Given paths π_1, π_2 so that for $i = 1,2$ and all $t \in \mathcal{T}_\Sigma$, $(\mathrm{H}t).\pi_i$ is defined, does $(\mathrm{H}t).\pi_1 = (\mathrm{H}t).\pi_2$ hold for all $t \in \mathcal{T}_\Sigma$?

Proof. For the first statement, we proceed by induction on the length of π. For $\pi = \epsilon$, $(\mathrm{H}t).\pi$ is defined for all $t \in \mathcal{T}_\Sigma$. Therefore, assume that π is different from ϵ and that the question can be decided for all proper suffixes of π. Then we first check that $(\mathrm{H}a).\pi$ is defined for all atoms a in $\mathcal{T}_\Sigma$. Now assume that this is the case. Then we perform a check for each constructor $f \in \Sigma$ where $\mathrm{H}f$ is not just a hom-variable $[X_j]_{\mathrm{H}}$ for some j. For that, we construct $(\mathrm{H}f).\pi$. If this selection is undefined, the check immediately fails. Otherwise, we consider the following two situations:

1. $(\mathrm{H}f).\pi$ equals some subterm of $\mathrm{H}f$. Then the check succeeds.
2. $(\mathrm{H}f).\pi = [X_j]_{\mathrm{H}}.\pi'$ for some j and some proper suffix π' of π. In this case, the check succeeds whenever $(\mathrm{H}t).\pi'$ is defined for all $t \in \mathcal{T}_\Sigma$.

The latter property is decidable by inductive hypothesis. If all these checks succeed, we answer 'yes,' otherwise 'no.' This completes the algorithm.

By induction on the length of paths π, it can be seen that the decision procedure terminates. It remains to argue that the algorithm always returns the correct result. Again by induction on the length of π, we show the following two claims:

— If $(\mathrm{H}t).\pi$ is defined for all $t \in \mathcal{T}_\Sigma$, then the algorithm will return 'yes.'
— If the algorithm returns 'yes,' $(\mathrm{H}t).\pi$ is defined for all $t \in \mathcal{T}_\Sigma$.

For a proof of the first claim, we note that $(\mathrm{H}a).\pi$ is defined for all atoms a. Now consider a constructor $f \in \Sigma$ of arity $k > 0$, where $\mathrm{H}f$ is not just a hom-variable together with arbitrary terms $t_1,\ldots,t_k \in \mathcal{T}_\Sigma$. By assumption, $(\mathrm{H}(f(t_1,\ldots,t_k))).\pi$ is defined for each such terms. Then, in particular, $(\mathrm{H}f).\pi$ must be defined. Then we distinguish two cases. If $(\mathrm{H}f).\pi$ equals a subterm of $\mathrm{H}f$, then the check of the algorithm for f succeeds. Otherwise, $(\mathrm{H}f).\pi = [X_j]_{\mathrm{H}}.\pi'$ for some j and proper suffix of π, where $(\mathrm{H}t_j).\pi'$ is defined. Therefore,

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

by inductive hypothesis, the check of f succeeds as well. Since all checks succeed, the algorithm necessarily answers 'yes.'

For a proof of the second claim, consider any tree $t \in \mathcal{T}_\Sigma$. Our goal is to prove that $(\mathrm{H}t).\pi$ is defined. We proceed by structural induction on t. If t is an atom, the claim is trivially true. Otherwise, $t = f(t_1, \ldots, t_k)$ for some $k > 0$. Now consider the hom-term $\mathrm{H}f$. If it equals a hom-variable $[X_j]_\mathrm{H}$, then $(\mathrm{H}t).\pi = (\mathrm{H}t_j).\pi$ and the claim follows. So assume that it equals a term with at least one constructor. According to the algorithm, $(\mathrm{H}f).\pi$ must be defined. If $(\mathrm{H}f).\pi$ equals a subterm of $\mathrm{H}f$, $(\mathrm{H}f(t_1, \ldots, t_k)).\pi$ is also defined. Otherwise, $(\mathrm{H}f).\pi = [X_j]_\mathrm{H}.\pi'$ for a proper suffix π' of π. Since the check for f succeeds, we can apply the inductive hypothesis for π' to deduce that $(\mathrm{H}t_j).\pi'$ is defined. But then also $(\mathrm{H}f(t_1, \ldots, t_k)).\pi$ is defined.

Now consider the second statement of the lemma. So assume that $(\mathrm{H}t).\pi$ is defined for every $t \in \mathcal{T}_\Sigma$. Then this is in particular the case for an atom $t = a \in \Sigma$. Let $s = (\mathrm{H}a).\pi$. It remains to verify that for all trees $t \in \mathcal{T}_\Sigma$, $(\mathrm{H}t).\pi = s$ holds. We perform an induction on the length of π as we did in the proof of the first statement.

Assume that the property can be decided for all proper suffixes of π. Then the property holds for π if and only if for all $f \in \Sigma$, one of the following three properties is satisfied:

a. $(\mathrm{H}f).\pi = s$.
b. $\mathrm{H}f$ equals a hom-variable $[X_j]_\mathrm{H}$.
c. $(\mathrm{H}f).\pi$ equals $[X_j]_\mathrm{H}.\pi'$ for some proper suffix π' of π, and $(\mathrm{H}t).\pi' = s$ for all $t \in \mathcal{T}_\Sigma$.

Properties (a) and (b) are obviously decidable. Since by inductive hypothesis, the third property is decidable, the assertion follows. For the correctness of the approach, first consider the *if* direction. Since we assume Σ to contain atoms, (b) cannot apply to all $f \in \Sigma$. Therefore, there is at least one $f \in \Sigma$ for which either (a) or (c) applies, and correctness follows inductively. For a proof of the *only if* direction by contradiction, assume that $t = f(t_1, \ldots, t_k)$ with $k \geqslant 0$ is a ground term such that $(\mathrm{H}t).\pi = s$ but none of the properties (a), (b) or (c) holds. Since (a) does not apply, $(\mathrm{H}f).\pi$ cannot be ground, i.e., $(\mathrm{H}f).\pi$ must contain at least one hom-variable or hom-path. Since (b) does not apply, $\mathrm{H}f$ contains at least one constructor. Therefore, if $(\mathrm{H}f).\pi$ equals $[X_j]_\mathrm{H}.\pi'$ as in (c), then π' must be a proper suffix of π. Since (c) does not apply, however, $(\mathrm{H}f).\pi$ cannot be a hom-path $[X_j]_\mathrm{H}.\pi'$. Therefore, $(\mathrm{H}f).\pi$ must be a hom-term s' which contains at least one hom-variable $[X_j]_\mathrm{H}$. But then H must map all terms to the same ground term. The latter, however, is not possible if $[X_j]_\mathrm{H}$ is a subterm of $(\mathrm{H}f).\pi$ and therefore – since $\mathrm{H}f$ contains at least one constructor – a proper subterm of $\mathrm{H}f$ itself – contradiction.

Finally, consider the third statement. By assumption, $(\mathrm{H}t).\pi_1$ and $(\mathrm{H}t).\pi_2$ are defined for all $t \in \mathcal{T}_\Sigma$ and thus also $(\mathrm{H}f).\pi_1$ and $(\mathrm{H}f).\pi_2$ are defined for all $f \in \Sigma$. If $\pi_1 = \pi_2$ then the equality holds trivially. Assume therefore that $\pi_1 \neq \pi_2$. Again, we perform an induction on the maximal length of the paths π_1, π_2.

If for some i, $(\mathrm{H}t).\pi_i$ always equals the same ground term s, then the equality holds iff $(\mathrm{H}t).\pi_{3-i}$ also equals s for all $t \in \mathcal{T}_\Sigma$ – which is decidable.

Therefore, subsequently assume that there are terms t_1, t_2 so that $(\mathrm{H}t_i).\pi_1$ and hence also $(\mathrm{H}t_i).\pi_2$ are distinct. For each $f \in \Sigma$, where $\mathrm{H}f$ is different from a hom-variable $[X_j]_\mathrm{H}$, we perform a check for $(\mathrm{H}f).\pi_1$ and $(\mathrm{H}f).\pi_2$. First, assume that for both $i = 1, 2$,

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

$(\mathrm{H}f).\pi_i = [X_{j_i}]_{\mathrm{H}}.\pi_i'$ for some j_i and a proper suffix π_i' of π_i. Then the check succeeds iff either $j_1 = j_2$ and $(\mathrm{H}t).\pi_1' = (\mathrm{H}t).\pi_2'$ for all $t \in \mathcal{T}_\Sigma$, or $j_1 \neq j_2$ and there is a ground term s so that for all $t \in \mathcal{T}_\Sigma$, $(\mathrm{H}t).\pi_1' = (\mathrm{H}t).\pi_2' = s$. Second, assume that for some i, $(\mathrm{H}f).\pi_i = [X_j]_{\mathrm{H}}.\pi_i'$ for some j where π_i' is a proper suffix of π_i, and $(\mathrm{H}f).\pi_{3-i}$ equals some ground term s. Then the check succeeds iff for all $t \in \mathcal{T}_\Sigma$, $(\mathrm{H}t).\pi_i' = s$.

Otherwise, the check succeeds iff $(\mathrm{H}f).\pi_1$ and $(\mathrm{H}f).\pi_2$ are identical.

Altogether, the algorithm returns 'yes' iff for every symbol $f \in \Sigma$, the check succeeds.

It is easy to see that for all $t \in \mathcal{T}_\Sigma$, $(\mathrm{H}t).\pi_1 = (\mathrm{H}t).\pi_2$ holds whenever the algorithm answers 'yes.' For the correctness of the algorithm, we therefore only consider the reverse direction.

Assume that for all $t \in \mathcal{T}_\Sigma$, $(\mathrm{H}t).\pi_1 = (\mathrm{H}t).\pi_2$. Assume further that there are terms t_1 and t_2 in $\mathcal{T}_\Sigma$ so that $(\mathrm{H}t_1).\pi_1 \neq (\mathrm{H}t_2).\pi_1$ and $(\mathrm{H}t_1).\pi_2 \neq (\mathrm{H}t_2).\pi_2$ (if for a given pair t_1, t_2, one disequality holds while the other does not hold, we immediately arrive at a contradiction to the assumption). In particular, we have $\mathrm{H}t_1 \neq \mathrm{H}t_2$. Consider a constructor $f \in \Sigma$ of arity $k \geqslant 0$. We have to verify that the check for f necessarily succeeds.

Case 1. $(\mathrm{H}f).\pi_1 = [X_{j_1}]_{\mathrm{H}}.\pi_1'$ and $(\mathrm{H}f).\pi_2 = [X_{j_2}]_{\mathrm{H}}.\pi_2'$ for proper suffixes π_i' of π_i. First, assume that $j_1 = j_2$. For every ground term t, we construct the term $t' = f(t, \ldots, t)$. Then, $(\mathrm{H}t).\pi_1' = (\mathrm{H}t').\pi_1 = (\mathrm{H}t').\pi_2 = (\mathrm{H}t).\pi_2'$. Therefore, the check for f succeeds. Second, assume that $j_1 \neq j_2$. Consider any pair of ground trees t, t' and construct a term $t_{t,t'} = f(t_1, \ldots, t_k)$ where for some fixed ground term t_0, $t_i = t$ if $i = j_1$, $t_i = t'$ if $i = j_2$, and $t_i = t_0$ otherwise. Then, $(\mathrm{H}t).\pi_1' = (\mathrm{H}t_{t,t'}).\pi_1 = (\mathrm{H}t_{t,t'}).\pi_2 = (\mathrm{H}t').\pi_2'$. Since t, t' are arbitrary, it follows that there is some ground term s such that $(\mathrm{H}t).\pi_i' = s$ for all $t \in \mathcal{T}_\Sigma$. Thus, the check for f again succeeds.

Case 2. Both $(\mathrm{H}f).\pi_1$ and $(\mathrm{H}f).\pi_2$ are subterms of $\mathrm{H}f$. We prove that these subterms are necessarily identical. Assume for a contradiction, that this were not the case. Then there is a shortest path π where the two subterms differ. If both $(\mathrm{H}f).\pi_1.\pi$ and $(\mathrm{H}f).\pi_2.\pi$ are not hom-variables, then $(\mathrm{H}t').\pi_1$ cannot be equal to $(\mathrm{H}t').\pi_2$ for every ground term $t' = f(t_1, \ldots, t_k)$. Therefore, w.l.o.g., assume that $(\mathrm{H}f).\pi_1.\pi = [X_j]_{\mathrm{H}}$ for some j, while $(\mathrm{H}f).\pi_2.\pi$ is not identical to $[X_j]_{\mathrm{H}}$. If $(\mathrm{H}f).\pi_2.\pi$ contains the hom-variable $[X_j]_{\mathrm{H}}$, then for no ground term $t' = f(t_1, \ldots, t_k)$, $(\mathrm{H}t').\pi_1.\pi = (\mathrm{H}t').\pi_2.\pi$ – because the occurs-check for unification fails in that case. Then, $(\mathrm{H}t').\pi_1 = (\mathrm{H}t').\pi_2$ cannot hold either – in contradiction to our assumption. So assume that $(\mathrm{H}f).\pi_2.\pi$ does not contain $[X_j]_{\mathrm{H}}$. For $t \in \mathcal{T}_\Sigma$, consider all terms $t' = f(t_1, \ldots, t_k)$ where $t = t_j$. Then it follows that $(\mathrm{H}t').\pi_1.\pi$ must be all equal to some ground term s, implying that $\mathrm{H}t = s$ for all $t \in \mathcal{T}_\Sigma$ – in contradiction to our assumption.

Case 3. There is some i so that $(\mathrm{H}f).\pi_i$ equals $[X_j]_{\mathrm{H}}.\pi_i'$ for some proper suffix π_i' of π_i, while $(\mathrm{H}f).\pi_{3-i}$ equals a ground subterm s of $\mathrm{H}f$. By defining $t' = f(t_1, \ldots, t_k)$ with $t_j = t$, we conclude that $(\mathrm{H}t).\pi_i' = (\mathrm{H}t').\pi_i = s$ for all t. Therefore, the check for f will succeed.

Case 4. None of the cases 1, 2 or 3 applies. Thus, there is some i so that $(\mathrm{H}f).\pi_i$ equals $[X_j]_{\mathrm{H}}.\pi_i'$ for some proper suffix π_i' of π_i, while $(\mathrm{H}f).\pi_{3-i}$ equals a non-ground subterm of $\mathrm{H}f$. In fact, here similar arguments apply as in Case 2. If $[X_j]_{\mathrm{H}}$ also occurs in $(\mathrm{H}f).\pi_{3-i}$, we immediately arrive at a contradiction since no subterm can be equal to a proper

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

superterm. And if $[X_j]_{\mathrm{H}}$ does not occur in $(\mathrm{H}f).\pi_{3-i}$, but some $[X_{j'}]_{\mathrm{H}}$ then again we may conclude that Ht must be equal for all $t \in \mathcal{T}_\Sigma$ – in contradiction to our assumption.

This concludes the proof. □

Lemma 2.4. The following problems are decidable:

1. Whether or not a hom-disequality $[X_j]_{\mathrm{H}} \neq s$ is satisfiable.
2. Whether or not a hom-path disequality $[X_j]_{\mathrm{H}}.\pi \neq s$ for a ground term s is satisfiable.
3. Whether or not a hom-path disequality $[X_j]_{\mathrm{H}}.\pi \neq [X_{j'}]_{\mathrm{H}}.\pi'$ is satisfiable.

Proof. In the first problem, the disequality is unsatisfiable whenever $s = [X_j]_{\mathrm{H}}$. Otherwise, we distinguish several cases. First, assume that H maps all terms to the same ground term s'. This is the case iff the homomorphism maps each constructor f either to s' or $\mathrm{H}f = [X_i]_{\mathrm{H}}$ for some i. Then consider any ground substitution θ. W.r.t. this substitution, $\mathrm{H}(X_i\theta) = s'$ for each variable X_i. Therefore, the disequality is satisfiable iff $s' \neq s\{[X_i]_{\mathrm{H}} \mapsto s' \mid X_i \in \mathcal{X}\}$ – which is trivially decidable. Otherwise, there are terms t_1, t_2 so that $\mathrm{H}t_1 \neq \mathrm{H}t_2$. If $[X_j]_{\mathrm{H}}$ occurs in s and s is different from $[X_j]_{\mathrm{H}}$, then the disequality is satisfied by every substitution. Therefore, finally assume that $[X_j]_{\mathrm{H}}$ does not occur in s. For $i = 1, 2$, define the substitution θ_i which maps all variables X_k occurring in s to t_1 and X_j to t_i. Since $\mathrm{H}t_1 \neq \mathrm{H}t_2$, either $\theta_1 \models [X_j]_{\mathrm{H}} \neq s$ or $\theta_2 \models [X_j]_{\mathrm{H}} \neq s$. Therefore, the disequality is satisfiable.

Now consider the second decision problem. By Lemma 2.3, it is decidable whether $(\mathrm{H}t).\pi$ is defined for all $t \in \mathcal{T}_\Sigma$. If $(\mathrm{H}t).\pi$ is undefined for some $t \in \mathcal{T}_\Sigma$, then the disequality is satisfiable. Therefore, now assume that $(\mathrm{H}t).\pi$ is defined for all trees t. Again by Lemma 2.3, it is decidable whether $(\mathrm{H}t).\pi$ always equals the same term, and if so, this term can be effectively computed. Thus, if Ht always equals the same term s_0, then the disequality is satisfiable iff $s_0 \neq s$. Otherwise, there are terms t_1, t_2 so that $(\mathrm{H}t_1).\pi \neq (\mathrm{H}t_2).\pi$. In this case, either $(\mathrm{H}t_1).\pi \neq s$ or $(\mathrm{H}t_2).\pi \neq s$. Hence, the disequality is necessarily satisfiable. Altogether, decidability of satisfiability of the second problem follows.

It remains to consider the third decision problem. Again by Lemma 2.3, it is decidable whether there is a term t so that either $(\mathrm{H}t).\pi$ or $(\mathrm{H}t).\pi'$ is undefined. In this case, the constraint is satisfiable. Therefore, assume that both $(\mathrm{H}t).\pi$ and $(\mathrm{H}t).\pi'$ are defined for all $t \in \mathcal{T}_\Sigma$. First, assume that X_j is different from $X_{j'}$. If for all $t \in \mathcal{T}_\Sigma$, $(\mathrm{H}t).\pi'$ is always equal to the same ground term s, then we can compute that term s and apply the method from the previous problem to decide satisfiability. The same holds true if $(\mathrm{H}t).\pi$ is always equal to the same ground term s. Therefore, now assume that there are terms $t, t' \in \mathcal{T}_\Sigma$ so that $(\mathrm{H}t).\pi \neq (\mathrm{H}t').\pi$. Consider the substitutions $\theta = \{X_j \mapsto t, X_{j'} \mapsto t'\}$ and $\theta' = \{X_j \mapsto t', X_{j'} \mapsto t'\}$. Then either $\theta \models [X_j]_{\mathrm{H}}.\pi \neq [X_{j'}]_{\mathrm{H}}.\pi'$ or $\theta' \models [X_j]_{\mathrm{H}}.\pi \neq [X_{j'}]_{\mathrm{H}}.\pi'$, and the disequality is satisfiable.

So it remains to consider the case where the same variable occurs on both sides, i.e., $j = j'$. In this case, the disequality is unsatisfiable iff for all terms $t \in \mathcal{T}_\Sigma$, $(\mathrm{H}t).\pi$ and $(\mathrm{H}t).\pi'$ are defined and equal. The latter problem is decidable by Lemma 2.3. □

The following lemma provides the construction for splitting general hom-term disequalities into disjunctions of hom-path disequalities.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Lemma 2.5. For every general hom-term disequality $t_1 \neq t_2$ w.r.t. a tree homomorphism H, a finite disjunction ϕ_{H} of satisfiable hom-path disequalities can be constructed which is equivalent to $t_1 \neq t_2$, i.e., for every ground substitution θ, it holds that $\theta \models (t_1 \neq t_2)$ iff $\theta \models \phi_{\mathrm{H}}$. Moreover, the ground terms occurring in the disjunction may be chosen as atoms.

Proof. In the first step, we reuse the construction for general terms from Seidl and Reuß (2012). For that, we observe that the disequality $t_1 \neq t_2$ is equivalent to a finite disjunction of disequalities $[X]_{\mathrm{H}}.\pi \neq t$ for suitable hom-paths $[X]_{\mathrm{H}}.\pi$ and subterms t occurring in t_1 or t_2. In order to see this, assume that $t_1 \neq t_2$ does not have the form $[X_j]_{\mathrm{H}}.\pi \neq t$. If it is of the form $t \neq [X_j]_{\mathrm{H}}.\pi$, we just swap the two sides of the disequality. Otherwise, the disequality has the form $f(t_1,\ldots,t_k) \neq f'(t'_1,\ldots,t'_n)$. Now, if $f \neq f'$, then the disequality is equivalent to true. The assertion of the lemma can then be proven by choosing, e.g., a single disequality of the form $f([X_j]_{\mathrm{H}},\ldots) \neq [X_j]_{\mathrm{H}}$, which is true for every ground substitution. Otherwise, if $f = f'$, then $k = n$, and the disequality is equivalent to the disjunction

$$\bigvee_{1 \leqslant i \leqslant k} t_i \neq t'_i.$$

Using structural induction, we may assume that we can construct finite disjunctions ψ_i of disequalities of the form $[X_j]_{\mathrm{H}}.\pi \neq t$ equivalent to $t_i \neq t'_i$, for all $1 \leqslant i \leqslant k$. Then, $t_1 \neq t_2$ is equivalent to $\bigvee_{1 \leqslant i \leqslant k} \psi_i$.

Let us now proceed with one fixed disequality $[X_j]_{\mathrm{H}}.\pi \neq t$ and show that an equivalent finite disjunction of hom-path disequalities can be constructed. The proof is by induction on the size of t. If $t = a$ for some atom $a \in \Sigma$ or a hom-path, then the disequality can be chosen as the result. Therefore, assume that $t = f(t_1,\ldots,t_k)$. First we claim:

Claim. $[X_j]_{\mathrm{H}}.\pi \neq t$ iff $\bigvee_{i=1}^{k} [X_j]_{\mathrm{H}}.\pi.(f,i) \neq t_i$.

Proof. By negating the two assertions, the claim is equivalent to

$$[X_j]_{\mathrm{H}}.\pi = t \quad \text{iff} \quad \bigwedge_{i=1}^{k} [X_j]_{\mathrm{H}}.\pi.(f,i) = t_i.$$

In order to prove this statement, first assume that $\theta \models [X_j]_{\mathrm{H}}.\pi = t$ for some ground substitution θ. Since $t.(f,i) = t_i$, we have $\theta \models [X_j]_{\mathrm{H}}.\pi.(f,i) = t_i$ for all $i = 1,\ldots,k$.

For the reverse implication, now assume that $\theta \models [X_j]_{\mathrm{H}}.\pi.(f,i) = t.(f,i) = t_i$ holds for all $i = 1,\ldots,k$. Then, $\mathrm{H}(X_j\theta).\pi = f(t_1\theta,\ldots,t_k\theta) = t\theta$ holds and therefore also $\theta \models [X_j]_{\mathrm{H}}.\pi = t$. This completes the proof of the claim. □

Returning to the main proof, we know by induction hypothesis that for all $i = 1,\ldots,k$, there is a disjunction ψ_i containing only disequalities between hom-paths and atoms, which is equivalent to $[X_j]_{\mathrm{H}}.\pi.(f,i) \neq t_i$. Since $[X_j]_{\mathrm{H}} \neq t$ is equivalent to $\bigvee_{i=1}^{k} [X_j]_{\mathrm{H}}.\pi.(f,i) \neq t_i$, we obtain that $[X_j]_{\mathrm{H}} \neq t$ is also equivalent to $\psi_1 \vee \ldots \vee \psi_k$.

Finally, since we can identify and thus remove unsatisfiable disequalities according to Lemma 2.4, the assertion of the lemma follows. □

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Example 2.6. The disequality

$$g([X_1]_{\mathrm{H}}.(f,2)) \neq g(f([X_2]_{\mathrm{H}}.(g,1), g(a)))$$

is equivalent to

$$[X_1]_{\mathrm{H}}.(f,2).(f,1) \neq [X_2]_{\mathrm{H}}.(g,1) \vee [X_1]_{\mathrm{H}}.(f,2).(f,2).(g,1) \neq a.$$

2.4. *Horn clauses with constraints*

A *constrained Horn clause* c is an implication

$$B_0 \Leftarrow B, \phi,$$

where B_0 is a literal, B is a finite set $\{B_1, \ldots, B_m\}$ of literals B_i and ϕ is a constraint, i.e., a positive Boolean combination of atomic disequalities. The left-hand side B_0 is the *head* of the clause c, while the sequence B, ϕ denotes the *body* or *precondition* of c. Variables occurring in the body of the clause, but not in the head, are called *auxiliary*. As a textual representation, B may also be written as a comma-separated sequence $B_1, \ldots, B_m$. Then, $B_1, \ldots, B_m$ is considered as the (textual) representation of the *set* $\{B_1, \ldots, B_m\}$, i.e., we silently may drop repeated occurrences of literals and also consider two sequences as equivalent if they agree up to reordering. The constraint ϕ in the body of the clause imposes a restriction on the applicability of the clause. A constraint ϕ which is always true can be omitted. If ϕ is a conjunctive constraint, the atomic constraints in ϕ may be written as a comma-separated sequence.

Let $\mathcal{C}$ denote a set of Horn clauses. Then the *least model* $\mathcal{M}_{\mathcal{C}}$ of $\mathcal{C}$ is the set of facts $p(t)$ (p a predicate, t a ground term) which can be obtained by means of a *resolution tree* with clauses from $\mathcal{C}$. Assume that $c \in \mathcal{C}$ is a clause of the form $p(t_0) \Leftarrow p_1(t_1), \ldots, p_k(t_k), \phi$ and that there is a ground substitution θ of the variables occurring in c such that $\theta \models \phi$, and for $i = 1, \ldots, k$, $p_i(t_i\theta)$ can be obtained by means of the resolution tree ρ_i. Then, $\langle c, \theta \rangle (\rho_1, \ldots, \rho_k)$ is a resolution tree for the fact $p(t_0\theta)$.

The language $\{t \in \mathcal{T}_\Sigma \mid p(t) \in \mathcal{M}_{\mathcal{C}}\}$ of p is also denoted $[\![p]\!]_{\mathcal{C}}$. For convenience, we also consider the set $[\![p]\!]^i_{\mathcal{C}}$ which consists of all trees t where the ground fact $p(t)$ can be obtained by a resolution tree of depth at most i. Consider as an example the set $\mathcal{A}$ consisting of two (non-constrained) Horn clauses:

$$p(f(X_1)) \Leftarrow p(X_1) \qquad \text{and} \qquad p(a) \Leftarrow .$$

Here, we have $[\![p]\!]^i_{\mathcal{A}} = \{f^j(a) \mid 0 \leqslant j \leqslant i-1\}$, the number of elements in $[\![p]\!]^i_{\mathcal{A}}$ is given by $|[\![p]\!]^i_{\mathcal{A}}| = i$, and the language *accepted* (or *recognized*) by p is given by

$$[\![p]\!]_{\mathcal{A}} = \bigcup \{[\![p]\!]^i_{\mathcal{A}} \mid i \geqslant 0\} = \{f^i(a) \mid i \geqslant 0\} = \{a, f(a), f(f(a)), \ldots\}.$$

We remark that each finite set of constrained Horn clauses is equivalent to a finite set of Horn clauses using conjunctive constraints only.

Lemma 2.6. Let $\mathcal{C}$ denote a finite set of constrained Horn clauses. Then a finite set $\mathcal{C}'$ of constrained Horn clauses can be constructed using conjunctive constraints only where $[\![p]\!]_{\mathcal{C}} = [\![p]\!]_{\mathcal{C}'}$ for all predicates p.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Proof. The set $\mathcal{C}'$ is constructed as follows. Consider each clause c of the form $p(t) \Leftarrow \alpha, \phi$ from $\mathcal{C}$. Let $\phi_1, \ldots, \phi_r$ be conjunctions of atomic constraints in ϕ so that $\phi_1 \vee \ldots \vee \phi_r$ is a DNF equivalent to ϕ. Then the clause c is simulated by $\mathcal{C}'$ by means of the r clauses

$$p(t) \Leftarrow \alpha, \phi_j, \quad j = 1, \ldots, r.$$

Then for each ground substitution θ, $p(t\theta) \Leftarrow \alpha\theta, \phi\theta$ is equivalent to the conjunction of the implications $p(t\theta) \Leftarrow \alpha\theta, \phi_j\theta, j = 1, \ldots, r$. Here, for $\alpha = p_1(t_1), \ldots, p_m(t_m)$, $\alpha\theta$ is given by $p_1(t_1\theta), \ldots, p_m(t_m\theta)$. Therefore, the least models of $\mathcal{C}$ and $\mathcal{C}'$ coincide. □

$\mathcal{H}_1$-clauses, normal clauses, automaton clauses. Let us introduce the subclasses of constrained Horn clauses which we consider here. A constrained Horn clause is an $\mathcal{H}_1$-clause if the term t in the head $p(t)$ is *flat* and *linear*, i.e., t contains at most one constructor, and no variable occurs twice in t. For convenience, we adopt the convention that the variables in the heads of $\mathcal{H}_1$-clauses are enumerated $X_1, \ldots, X_k$, i.e., an $\mathcal{H}_1$-clause has one of these two forms:

$$p(X_1) \quad \Leftarrow \quad \alpha, \phi \qquad \textbf{or} \qquad p(f(X_1, \ldots, X_k)) \Leftarrow \alpha, \phi$$

for an arbitrary sequence of literals α and an arbitrary constraint ϕ. The case of atoms:

$$p(a) \Leftarrow \alpha, \phi$$

is subsumed by choosing $k = 0$.

An $\mathcal{H}_1$-clause is *simple* if no literal in α contains a constructor, i.e., each literal in α is of the form $q(X_j)$ where X_j may or may not occur in the head of the clause. The $\mathcal{H}_1$-clause is a *normal clause* if it is of the form:

$$q(f(X_1, \ldots, X_k)) \Leftarrow p_1(X_{i_1}), \ldots, p_r(X_{i_r}), \phi \quad (r \geqslant 0),$$

where all variables occurring in the body of the clause, i.e., in the literals as well as in the constraint ϕ, also occur in the head. Note that according to our convention, the variable X_i is distinct from X_j for $i \neq j$, while predicates p_i, p_j may coincide for $i \neq j$. Also the (universal) quantification of variables is implicit in each clause.

A normal clause is an *automaton clause* if additionally each variable X_i occurring in the head occurs exactly once in the literals occurring in the body, and the head contains exactly one constructor, i.e., the clause has the form:

$$q(f(X_1, \ldots, X_k)) \Leftarrow p_1(X_1), \ldots, p_k(X_k), \phi$$

for $k \geqslant 0$ where ϕ only contains variables $X_1, \ldots, X_k$. We also will consider *generalized* automaton clauses. These are of the form

$$q(t_0) \Leftarrow p_1(X_1), \ldots, p_k(X_k), \phi$$

for some $k \geqslant 0$, where t_0 is an arbitrary term (not necessarily linear).

While each normal clause as well as each automaton clause is an $\mathcal{H}_1$-clause, a generalized automaton clause need not be contained in $\mathcal{H}_1$. Finite sets of automaton clauses can be seen as finite tree automata with constraints. The predicates then correspond to the states, while the clauses describe the transitions of the automaton. Different classes of disequality constraints correspond to different classes of constrained tree automata.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

We illustrate these definitions by small examples. For a distinction, variables that do *not* occur in the head are denoted $Y, Y_1, \ldots$

Example 2.7. The clauses

$$\begin{array}{ll} p(f(X_1, X_1)) & \Leftarrow q(X_1) \\ p(f(X_1, g(X_2))) & \Leftarrow q(h(X_2)) \end{array}$$

do not belong to $\mathcal{H}_1$. While the first clause has a non-linear head (X_1 occurs more than once), the second one has more than one constructor in the head. The clause

$$p(f(X_1, X_2)) \Leftarrow p(f(X_1, g(X_2))), X_1 \neq g(X_2)$$

on the other hand, is an $\mathcal{H}_1$-clause with a term disequality constraint. Moreover, the following clauses are simple, normal and automaton clauses, respectively:

$$\begin{array}{lll} p(X_1) & \Leftarrow & q_1(X_1), q_2(X_1), q_3(Y), \quad X_1 \neq h(Y, a), \\ p(f(X_1, X_2)) & \Leftarrow & q_1(X_1), q_2(X_1), q_3(X_2), X_1 \neq h(X_2, a), \\ p(f(X_1, X_2)) & \Leftarrow & q_1(X_1), q_2(X_2), \qquad\quad X_1 \neq h(X_2, a). \end{array}$$

Using dynamic programming, we obtain the following proposition:

Lemma 2.7. Membership of a ground term in $[\![p]\!]_{\mathcal{A}}$ is decidable in polynomial time for every finite set $\mathcal{A}$ of automaton clauses with general hom-term disequalities (and predicate p).

Proof. First, we observe that for a given ground substitution θ and a constraint ϕ of general hom-term disequalities, it is efficiently decidable whether or not $\theta \models \phi$. For a ground term t', let $\mathsf{valid}_{\mathcal{A}}(t')$ denote the set of all predicates p with $t' \in [\![p]\!]_{\mathcal{A}}$. The algorithm for testing membership of a term t determines for all subterms t' of t the sets $\mathsf{valid}_{\mathcal{A}}(t')$. Then, $t' \in [\![p]\!]_{\mathcal{A}}$ iff $p \in \mathsf{valid}_{\mathcal{A}}(t')$. The algorithm proceeds bottom-up over t. Assume that the algorithm has already determined the sets $\mathsf{valid}_{\mathcal{A}}(t')$ for all proper subterms t' of a term t_1. In order to determine $\mathsf{valid}_{\mathcal{A}}(t_1)$, the algorithm iterates over all clauses of $\mathcal{A}$. The set $\mathsf{valid}_{\mathcal{A}}(t_1)$ then consists of all predicates p for which there is a clause $p(s) \Leftarrow p_1(X_1), \ldots, p_k(X_k), \phi$ with $t_1 = s\theta$, $\theta = \{X_1 \mapsto s_1, \ldots, X_k \mapsto s_k\}$, such that for all $i = 1, \ldots, k$, $p_i \in \mathsf{valid}_{\mathcal{A}}(s_i)$ and $\theta \models \phi$. Thus, the set $\mathsf{valid}_{\mathcal{A}}(t_1)$ can be computed for all subterms t' of t_1 in time polynomial in the size of $\mathcal{A}$, t_1, and the given tree homomorphism. □

Every set $\mathcal{N}$ of normal clauses without constraints can be transformed to an equivalent set $\mathcal{A}$ of automaton clauses whose predicates correspond to conjunctions $p_1 \wedge \ldots \wedge p_j, j \geqslant 0$, of original predicates p_i from $\mathcal{N}$. This also holds for constrained normal clauses, since preconditions are closed under conjunction.

Lemma 2.8. For every finite set $\mathcal{N}$ of normal clauses with constraints, a finite set $\mathcal{A}$ of automaton clauses with the same sort of constraints can be effectively constructed with $[\![p]\!]_{\mathcal{N}} = [\![p]\!]_{\mathcal{A}}$ for each predicate p of $\mathcal{N}$.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Proof. For every subset $P = \{p_1, \ldots, p_n\} (n \geqslant 0)$ of predicates p_i occurring in $\mathcal{N}$, we introduce a new predicate $[P]$ which will accept the set of terms

$$\bigcap_{i=1,\ldots,n} [\![p_i]\!]_{\mathcal{N}}.$$

For every such predicate $[P]$ and $f \in \Sigma$, the set of clauses in $\mathcal{A}$ has all clauses of the form

$$[P](f(X_1, \ldots, X_k)) \Leftarrow [P_1](X_1), \ldots, [P_k](X_k), \phi$$

for clauses $p_i(f(X_1, \ldots, X_k)) \Leftarrow \alpha_i, \phi_i$, $i = 1, \ldots, n$, from $\mathcal{N}$ where

$$\begin{array}{lcl} P_i & = & \{q \mid \exists\, j.\, q(X_i) \text{ occurs in } \alpha_j\}, \\ \phi & = & \phi_1, \ldots, \phi_n. \end{array}$$

Let $\mathcal{C}$ be a set of clauses. We define $\models_{\mathcal{C}}$ jointly for conjunctions of literals $p(X_i)$ and constraints as follows. Let θ be a ground substitution. Then, $\theta \models_{\mathcal{C}} c$ for a constraint c iff $\theta \models c$, and $\theta \models_{\mathcal{C}} p(X_i)$ for a literal $p(X_i)$ iff $p(X_i\theta)$ is contained in the least model of $\mathcal{C}$. For a conjunction $F = F_1 \wedge F_2$, we recursively define $\theta \models_{\mathcal{C}} F$ iff $\theta \models_{\mathcal{C}} F_1 \wedge \theta \models_{\mathcal{C}} F_2$.

We claim that for all subsets of predicates $p_1, \ldots, p_n$, $[\![p_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![p_n]\!]_{\mathcal{N}} = [\![[\{p_1, \ldots, p_n\}]]\!]_{\mathcal{A}}$, and, in particular, $[\![p]\!]_{\mathcal{N}} = [\![[\{p\}]]\!]_{\mathcal{A}}$ for all predicates p of $\mathcal{N}$. Thus, we obtain the desired result by renaming the predicates $[\{p\}]$ of $\mathcal{A}$ to p.

For a proof of '$\supseteq$,' assume for a contradiction that $t \in [\![[\{p_1, \ldots, p_n\}]]\!]_{\mathcal{A}} \setminus ([\![p_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![p_n]\!]_{\mathcal{N}})$ is of minimal depth. Let $t = f(t_1, \ldots, t_k)$ for some $k \geqslant 0$. Let

$$[\{p_1, \ldots, p_n\}](f(X_1, \ldots, X_k)) \Leftarrow [P_1](X_1), \ldots, [P_k](X_k), \phi$$

be the clause in $\mathcal{A}$ which allows us to obtain the fact $[\{p_1, \ldots, p_n\}](t)$ from facts $[P_j](t_j)$ with the ground substitution $\theta = \{X_1 \mapsto t_1, \ldots, X_k \mapsto t_k\}$. By construction, there are clauses

$$p_i(f(X_1, \ldots, X_k)) \Leftarrow \alpha_i, \phi_i$$

in $\mathcal{N}$ for $i = 1, \ldots, n$ so that $\phi = \phi_1 \wedge \ldots \wedge \phi_n$ and $P_j = \{q \mid \exists\, i.\, \alpha_i \text{ contains } q(X_j)\}$. Since t is of minimal depth and $t_j \in [\![[P_j]]\!]_{\mathcal{A}}$ $(j = 1, \ldots, k)$, we have that $t_j \in [\![q]\!]_{\mathcal{N}}$ for all $q \in P_j$. Moreover, $\theta \models \phi$. Therefore, $\theta \models_{\mathcal{N}} \alpha_i, \phi_i$ for $i = 1, \ldots, n$. Accordingly, $f(t_1, \ldots, t_k) \in [\![p_i]\!]_{\mathcal{N}}$ for all i – in contradiction to our assumption.

For a proof of '$\subseteq$,' assume by contradiction that $t \in ([\![p_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![p_n]\!]_{\mathcal{N}}) \setminus [\![[\{p_1, \ldots, p_n\}]]\!]_{\mathcal{A}}$ is of minimal depth. Let $t = f(t_1, \ldots, t_k)$ for some $k \geqslant 0$ and $\theta = \{X_1 \mapsto t_1, \ldots, X_k \mapsto t_k\}$. Since $f(t_1, \ldots, t_k) \in [\![p_i]\!]_{\mathcal{N}}$ for all $i \in \{1, \ldots, n\}$, there must be corresponding clauses

$$p_i(f(X_1, \ldots, X_k)) \Leftarrow \alpha_i, \phi_i$$

in $\mathcal{N}$ for each $i \in \{1, \ldots, n\}$ so that $\theta \models_{\mathcal{N}} \alpha_i, \phi_i$. Now consider the automaton clause

$$[P](f(X_1, \ldots, X_k)) \Leftarrow [P_1](X_1), \ldots, [P_k](X_k), \phi$$

in $\mathcal{A}$ corresponding to this sequence of normal clauses, i.e., where $\phi = \phi_1 \wedge \ldots \wedge \phi_n$ and $P_j = \{q \mid \exists\, i.\, p(X_j) \text{ occurs in } \alpha_i\}$ for $j = 1, \ldots, k$. Then, $\theta \models \phi$. Moreover, for $j = 1, \ldots, k$, $t_j \in [\![q]\!]_{\mathcal{N}}$ for all $q \in P_j$. By minimality of t, we therefore have that $t_j \in [\![[P_j]]\!]_{\mathcal{A}}$. But then also $f(t_1, \ldots, t_k) \in [\![[P]]\!]_{\mathcal{A}}$ – in contradiction to our assumption. □

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Example 2.8. We exemplify the construction from Lemma 2.8 for a set of normal clauses with (ordinary) term disequalities. For unary constructors s and arbitrary terms t, let $s^0(t) = t$ and $s^{i+1}(t) = s(s^i(t))$. Consider the following set $\mathcal{N}$ of normal clauses:

$$\begin{array}{lcl}
\mathrm{adult}(\mathrm{pers}(X_1, X_2)) & \Leftarrow & \mathrm{age}(X_1), \mathrm{name}(X_2), \mathrm{old}(X_1) \\
\mathrm{old}(X_1) & \Leftarrow & \bigwedge_{i \leqslant 18} X_1 \neq s^i(0) \\
\mathrm{age}(0) & \Leftarrow & \\
\mathrm{age}(s(X_1)) & \Leftarrow & \mathrm{age}(X_1)
\end{array}$$

The variable X_1 occurs twice in the literals of the precondition of the first clause, while it does not at all occur in the literals of the body of the second clause. Furthermore, the head of the second clause contains no constructor.

The first step towards constructing an equivalent finite set $\mathcal{A}$ of automaton clauses consists in instantiating the variables X_1 occurring in the heads of the normal clauses $p(X_1) \Leftarrow \phi$ with all terms $c(X_1, \ldots, X_k)$, where c is a constructor of arity $k \geqslant 0$. This gives rise to the following clauses:

$$\begin{array}{lcl}
(1)\quad \mathrm{old}(0) & \Leftarrow & \bigwedge_{i \leqslant 18} 0 \neq s^i(0) \\
(2)\quad \mathrm{old}(s(X_1)) & \Leftarrow & \bigwedge_{i \leqslant 18} s(X_1) \neq s^i(0) \\
(3)\quad \mathrm{old}(\mathrm{pers}(X_1, X_2)) & \Leftarrow & \bigwedge_{i \leqslant 18} \mathrm{pers}(X_1, X_2) \neq s^i(0)
\end{array}$$

by which we replace the single clause $\mathrm{old}(X_1) \Leftarrow \bigwedge_{i \leqslant 18} X_1 \neq s^i(0)$. The constraint of clause (1) is unsatisfiable because it contains the disequality $0 \neq 0$. Consequently, the clause may be removed. The constraint of clause (3) is equivalent to true. Consequently, the constraint may be removed. Finally, the constraint of clause (2) is equivalent to $\bigwedge_{i \leqslant 17} X_1 \neq s^i(0)$. We simplify $\mathcal{N}$ accordingly and then proceed with the transformation to automaton clauses. In what follows, we denote by $\top$ the predicate $[\{\}]$. Applying the construction from Lemma 2.8 to $\mathcal{N}$, we obtain the set $\mathcal{A}$ of clauses:

$$\begin{array}{lcl}
[\{\mathrm{adult}\}](\mathrm{pers}(X_1, X_2)) & \Leftarrow & [\{\mathrm{age}, \mathrm{old}\}](X_1), [\{\mathrm{name}\}](X_2) \\
[\{\mathrm{old}\}](s(X_1)) & \Leftarrow & \top(X_1), \bigwedge_{i \leqslant 17} X_1 \neq s^i(0) \\
[\{\mathrm{age}\}](0) & \Leftarrow & \\
[\{\mathrm{age}\}](s(X_1)) & \Leftarrow & [\{\mathrm{age}\}](X_1) \\
[\{\mathrm{old}\}](\mathrm{pers}(X_1, X_2)) & \Leftarrow & \top(X_1), \top(X_2) \\
[\{\mathrm{age}, \mathrm{old}\}](s(X_1)) & \Leftarrow & [\{\mathrm{age}\}](X_1), \bigwedge_{i \leqslant 17} X_1 \neq s^i(0) \\
\top(0) & \Leftarrow & \\
\top(s(X_1)) & \Leftarrow & \top(X_1) \\
\top(\mathrm{pers}(X_1, X_2)) & \Leftarrow & \top(X_1), \top(X_2) \\
[\{\mathrm{old}, \mathrm{adult}\}](\mathrm{pers}(X_1, X_2)) & \Leftarrow & [\{\mathrm{age}, \mathrm{old}\}](X_1), [\{\mathrm{name}\}](X_2)
\end{array}$$

The predicate $\top$ accepts all terms of $\mathcal{T}_\Sigma$, where $\Sigma = \{0, s, \mathrm{pers}\}$ is given by the atoms and constructors occurring in $\mathcal{N}$. The new predicate $[\{\mathrm{age}, \mathrm{old}\}]$ represents the conjunction of the predicates age and old, i.e., it accepts all terms that are accepted by both age and old. By renaming the predicates $[\{\mathrm{adult}\}], [\{\mathrm{age}\}], [\{\mathrm{old}\}], [\{\mathrm{name}\}]$ to adult, age, old, name, we achieve the final result with $[\![p]\!]_\mathcal{N} = [\![p]\!]_\mathcal{A}$ for all predicates occurring in $\mathcal{N}$.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

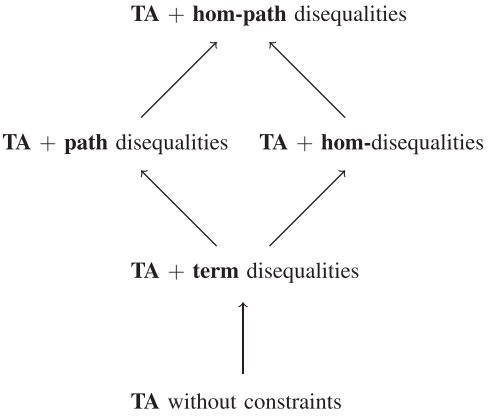


Fig. 1. Relative expressiveness of tree automata (TA) with different kinds of disequality constraints. The transitive reduction of all proper inclusions is indicated by arrows. All classes are decidable.

3. Expressiveness

We compare automata classes with different sorts of disequalities w.r.t. their expressiveness. Figure 1 shows a summary of classes of *normalizable* $\mathcal{H}_1$-extensions concerning their expressiveness. Let us first demonstrate by a small example that the addition of term disequalities leads to a proper generalization for finite bottom-up tree automata or finite sets of $\mathcal{H}_1$-clauses.

Example 3.1. Let $\Sigma = \{a, f\}$ where the symbols have arities 0 and 2, respectively. Then the language $\{f(t_1, t_2) \mid t_1 = t_2\}$ is not regular, i.e., is not expressible by means of finite tree automata or finite sets of automaton clauses. Accordingly, the complement language $\{f(t_1, t_2) \mid t_1 \neq t_2\}$ cannot be regular either. This language, however, can be expressed by the following set of automaton clauses with term disequality constraints:

$$\begin{array}{lcl} p(f(X_1, X_2)) & \Leftarrow & \top(X_1), \top(X_2), X_1 \neq X_2 \\ \top(f(X_1, X_2)) & \Leftarrow & \top(X_1), \top(X_2) \\ \top(a) & \Leftarrow & \end{array}$$

Consequently, term disequalities increase the expressiveness of finite sets of automaton clauses.

We show in Sections 4 and 5 how finite sets of $\mathcal{H}_1$-clauses with disequalities can be *normalized* into finite tree automata with disequalities ($\mathcal{H}_1$-normalization). Based on that, the expressiveness results of this section also apply to the respective classes of $\mathcal{H}_1$-clauses with corresponding disequalities.

Unlabelled paths and ground terms. Tree automata with path disequalities have been investigated by Comon and Jacquemard (1994). They have considered disequalities between *unlabelled* paths. Unlabelled paths $i_1 \ldots . i_n$ are obtained from labelled paths $(f_1, i_1). \ldots . (f_n, i_n)$ by omitting the labels f_j. The resulting automata, though, are equally

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

expressive, since the constructors occurring in paths can be recorded by means of specialized predicates – given that the set Σ of constructors and atoms is finite.

More generally, recording the topmost constructors is possible provided that the literals in pre-conditions do not contain any constructors. A corresponding construction is given in the following lemma. For that, we consider *patterns*, i.e., terms possibly containing occurrences of the *anonymous variable* _ (underscore) instead of ordinary variables. Thus, the set of patterns is given by $\mathcal{T}_\Sigma(\{_\})$. For patterns s, s', we define the combination operator $s \wedge s'$ recursively as follows:

— $_ \wedge t = t \wedge _ = t$.
— Assume $s = f(t_1, \ldots, t_n)$ and $s' = g(t'_1, \ldots, t'_{n'})$. If $f = g$ and thus $n = n'$, and $t_i \wedge t'_i$ does not *fail* for any $i \in \{1, \ldots, n\}$, then $s \wedge s' = f(t_1 \wedge t'_1, \ldots, t_n \wedge t'_n)$. Otherwise, $s \wedge s'$ *fails*.

Intuitively, $\wedge$ when applied to patterns s, s', either *succeeds* and returns a pattern which combines s, s' into the least pattern containing all non-_ nodes from s and s', or *fails* whenever no such pattern exists. In particular, the pattern s *matches* a ground term t if $s \wedge t$ does not fail and returns t. We remark that for any finite sequence $s_1, \ldots, s_r$ of patterns all matching the same ground term t, the pattern $s = s_1 \wedge \ldots \wedge s_r$ is defined and matches t.

The application of a path π to a pattern t is defined analogously to the case of ordinary terms, i.e., the expression $t.\pi$ is either *undefined* or results in a (sub-)pattern of t. We remark that for every path π, there is a minimal pattern $\mathsf{pat}(\pi)$ so that $\mathsf{pat}(\pi).\pi$ is defined. This pattern is defined by

$$\begin{array}{lcl} \mathsf{pat}(\epsilon) & = & _ \\ \mathsf{pat}((f,i).\pi') & = & f(t_1, \ldots, t_n) \qquad \text{if } f \text{ has arity } n \text{ where} \\ & & t_i = \mathsf{pat}(\pi') \text{ and } t_j = _ \text{ for all } j \neq i \end{array}$$

We also require the notion of a *full* pattern of depth $d \geqslant 0$. A pattern t is *full* of depth d if for every path π of length d, $t.\pi$ is either undefined or equals _. Let F_d denote the set of all patterns which are full of depth d. This set is inductively defined by

$$\begin{array}{lcl} F_0 & = & \{_\} \qquad \text{and for } d > 0, \\ F_d & = & \{f(t_1, \ldots, t_k) \mid f \in \Sigma \text{ of arity } k, t_1, \ldots, t_k \in F_{d-1}\}. \end{array}$$

We have the following:

Lemma 3.1. Assume that $d \geqslant 0$. Then the following statements hold:

1. For every pattern t of depth at most d, and every $t' \in F_d$, either $t \wedge t' = t'$ or $t \wedge t'$ fails.
2. For any $t, t' \in F_d$, either $t = t'$ holds or $t \wedge t'$ fails.
3. For any $s \in \mathcal{T}_\Sigma$, there is exactly one $t \in F_d$ so that $t \wedge s = s$ holds.

Proof. For a proof of the first statement, we proceed by induction on d. Since F_0 only contains _, the statement holds for $d = 0$. So assume that $d > 0$. If $t = _$, the assertion follows. Therefore, assume that t is different from _. By definition, t' is also different from _. If the root constructors of t, t' differ, $t \wedge t'$ fails. Otherwise, $t = f(t_1, \ldots, t_k)$ and $t' = f(t'_1, \ldots, t'_k)$ for some constructor f of arity $k \geqslant 0$ and $t_i, t'_i \in F_{d-1}$ for all $i = 1, \ldots, k$. By induction hypothesis, we know that for each $i = 1, \ldots, k$, either $t_i \wedge t'_i = t'_i$ or $t_i \wedge t'_i$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

fails. If for some $1 \leqslant i \leqslant k$, $t_1 \wedge t'_i$ fails, then $t \wedge t'$ must fail as well. Otherwise, for all $i = 1, \ldots, k$, $t_i \wedge t'_i = t'_i$—in which case also $t \wedge t' = t'$, and the statement follows.

The second statement immediately follows from statement 1: assume that $t \wedge t'$ does not fail. Then, $t = t' \wedge t = t \wedge t' = t'$, since the operator $\wedge$ is commutative.

For a proof of the third statement, we again proceed by induction on d. If $d = 0$, then we choose the pattern t as $_$. As $F_0 = \{_\}$, the statement follows for $d = 0$. So now assume that $d > 0$. Let $s = f(s_1, \ldots, s_k)$. By induction hypothesis, there exist patterns $t_1, \ldots, t_k \in F_{d-1}$ so that $t_i \wedge s_i = s_i$ for all $i = 1, \ldots, k$. But then the pattern $t = f(t_1, \ldots, t_k)$ is in F_d and matches s. Now assume for a contradiction that there is another pattern $t' \in F_d$ with $t' \wedge s = s$. Then t' must be of the form $t' = f(t'_1, \ldots, t'_k)$, where $t'_i \wedge s_i = s_i$ for all $i = 1, \ldots, k$. But then, by inductive hypothesis, $t'_i = t_i$ for $i = 1, \ldots, k$. Accordingly, $t = t'$ – in contradiction to our assumption. Hence, also the second statement follows. □

We also define the *unification* of a pattern t with a term s possibly containing variables. Unification either succeeds and then results in a minimal assignment θ of the variables in s to sub-patterns of t so that $t \wedge s\theta$ succeeds, or it *fails* – if no such substitution exists. Technically, the unification $\mathsf{unify}(t, s)$ relative to a superset $\mathcal{X}$ of variables occurring in s is defined as follows:

— $\mathsf{unify}(_, s) = \theta$ where $\theta(X) = _$ for all $X \in \mathcal{X}$.
— If $X \in \mathcal{X}$, then $\mathsf{unify}(t, X) = \theta$, where $\theta(X) = t$ and $\theta(X') = _$ for all $X' \neq X$.
— Assume that $t = f(t_1, \ldots, t_n)$ and $s = g(s_1, \ldots, s_{n'})$. If $f = g$ and thus also $n = n'$, and $\mathsf{unify}(t_i, s_i)$ succeeds and returns θ_i for all $i = 1, \ldots, n$, and furthermore a substitution θ can be defined so that for all $X \in \mathcal{X}$, $\theta(X) = \theta_1(X) \wedge \ldots \wedge \theta_n(X)$, then $\mathsf{unify}(t, s) = \theta$. Otherwise, $\mathsf{unify}(t, s)$ *fails*.

We remark that whenever $\mathsf{unify}(t, s)$ succeeds and returns θ, the depth of each pattern $X_j\theta$ is bounded by the depth of the pattern t.
We have the following:

Lemma 3.2. Consider a term t_0 containing variables $X_1, \ldots, X_k$, a ground substitution $\theta = \{X_1 \mapsto s_1, \ldots, X_k \mapsto s_k\}$ and a pattern t. Assume that t matches $t_0\theta$. Then the unification of t_0 and t succeeds with a substitution θ' (not necessarily ground) so that $X_j\theta' \wedge s_j = s_j$ for all $j = 1, \ldots, k$.

Proof. We proceed by induction on the structure of t. If $t = _$, then $\mathsf{unify}(t, t_0)$ is defined with $\theta' = \{X_j \mapsto _ \mid j = 1, \ldots, k\}$. So assume that $t = f(t_1, \ldots, t_n)$ for some $n \geqslant 0$. If $t_0 = X_j$ for some variable X_j, then $\mathsf{unify}(t, t_0)$ again succeeds, in this case with $\theta'(X_j) = t$ and $\theta'(X_{j'}) = _$ for $j' \neq j$. Then, $X_{j'}\theta' \wedge s_{j'} = s_{j'}$ trivially holds for $j' \neq j$. Since $s_j = X_j\theta = t_0\theta$, also $X_j\theta' \wedge s_j = s_j$ holds, and the claim follows.

Otherwise, t_0 necessarily is of the form $t_0 = f(t'_1, \ldots, t'_n)$ where for all $i = 1, \ldots, n$, t_i matches $t'_i\theta$. Therefore, by induction hypothesis, $\mathsf{unify}(t_i, t'_i)$ succeeds for each $i = 1, \ldots, n$ giving us a substitution θ'_i so that $X_j\theta'_i \wedge s_j = s_j$ for all $j = 1, \ldots, k$. In particular, the pattern $s'_j = X_j\theta'_1 \wedge \ldots \wedge X_j\theta'_n$ is defined for each $j = 1, \ldots, n$ where $s'_j \wedge s_j = s_j$ holds. Let $\theta' = \{X_1 \mapsto s'_1, \ldots, X_k \mapsto s'_k\}$. By definition, θ' is the substitution returned by $\mathsf{unify}(t, t_0)$. Therefore, the latter succeeds, and indeed, $X_j\theta' \wedge s_j = s_j$ holds for all $j = 1, \ldots, k$. □

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Finally, the *depth* of a pattern t is given by 0 if $t = _$, and by 1 plus the maximum depth of the patterns t_i if $t = f(t_1, \ldots, t_m), m \geqslant 0$ – where in case $k = 0$, i.e., when t is an atom, the depth is 1.

Lemma 3.3. Let $\mathcal{C}$ be a finite set of generalized automaton clauses and $d \geqslant 0$. For each predicate p of $\mathcal{C}$ consider fresh predicates $p_t, t \in F_d$. Then a set $\mathcal{C}'$ of generalized automaton clauses with the same sort of constraints as used in $\mathcal{C}$ can be constructed such that for each predicate p of $\mathcal{C}$ and each pattern $t \in F_d$, $[\![p_t]\!]_{\mathcal{C}'}$ equals the set of terms $s \in [\![p]\!]_{\mathcal{C}}$ where s is matched by the pattern t, i.e., $t \wedge s = s$.

Proof. W.l.o.g., we assume that the set $\mathcal{C}$ has a *universal* predicate $\top$ which accepts all terms in $\mathcal{T}_\Sigma$. Such a predicate could, e.g., be defined by the clauses:

$$\top(f(X_1, \ldots, X_k)) \Leftarrow \top(X_1), \ldots, \top(X_k)$$

for $f \in \Sigma$ with arity $k \geqslant 0$. If such a predicate is available, we may also w.l.o.g., assume that each clause in $\mathcal{C}$ is of the form

$$p(t_0) \Leftarrow p_1(X_1), \ldots, p_k(X_k), \phi$$

for some term t_0 and constraint ϕ, where each variable in t_0 is among the $X_1, \ldots, X_k$. This normal form can be achieved by adding literals $\top(X_j)$ to the body of the clause for variables X_j which occur in t_0 but not yet in literals of the body.

Consider a pattern $t \in F_d$, and a clause

$$p(t_0) \Leftarrow p_1(X_1), \ldots, p_k(X_k), \phi$$

of $\mathcal{C}$. Consider the unification of the pattern t and t_0. If it fails, the given clause of $\mathcal{C}$ cannot be used to obtain facts $p(s)$ so that s is matched by t. So assume that the unification succeeds and returns a substitution θ of the variables occurring in t_0. Then $\mathcal{C}'$ has all clauses:

$$p_t(t_0) \Leftarrow p_{1,t_1}(X_1), \ldots, p_{k,t_k}(X_k), \phi,$$

where for $i = 1, \ldots, k$, there is a pattern $t_i \in F_d$ so that $t_i = \theta(X_i) \wedge t_i$ if X_i occurs in t_0.

From the definition of the clauses in $\mathcal{C}'$, it follows that $[\![p_t]\!]_{\mathcal{C}'} \subseteq [\![p]\!]_{\mathcal{C}}$.

Now consider the reverse inclusion $[\![p_t]\!]_{\mathcal{C}'} \supseteq \{s \in [\![p]\!]_{\mathcal{C}} \mid t \wedge s = s\}$, and assume for a contradiction that $s \in [\![p]\!]_{\mathcal{C}} \setminus [\![p_t]\!]_{\mathcal{C}'}$ with $t \wedge s = s$, and the fact $p(s)$ has a minimal resolution tree in $\mathcal{C}$ for all such facts and patterns $t' \in F_d$. Let

$$p(t_0) \Leftarrow p_1(X_1), \ldots, p_k(X_k), \phi$$

be the root clause in a minimal resolution tree of the fact $p(s)$ in $\mathcal{C}$ with the ground substitution $\theta = \{X_1 \mapsto s_1, \ldots, X_k \mapsto s_k\}$. Thus, $s = t_0\theta$ and $\theta \models \phi$.

By Lemma 3.2, it follows that the unification of t_0 and t does not fail, but returns a substitution θ' (not necessarily ground) so that $X_j\theta'$ matches s_j for $j = 1, \ldots, k$ and has depth at most d. By Lemma 3.1, every ground term is matched by some pattern in F_d. Therefore, there are patterns $t_1, \ldots, t_k \in F_d$ so that for every $i = 1, \ldots, k$, s_i is matched by t_i. For every i where X_i occurs in t_0, s_i is also matched by $\theta'(X_i)$ and thus $\theta'(X_i) \wedge t_i$ is

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

defined and equals t_i. Therefore, $\mathcal{C}'$ contains the clause

$$p_t(t_0) \Leftarrow p_{1,t_1}(X_1), \ldots, p_{k,t_k}(X_k), \phi.$$

Since for each $i = 1, \ldots, k$, $p_i(s_i)$ holds and s_i is matched by the pattern t_i, the minimality of the resolution tree implies that $s_i \in [\![p_{i,t_i}]\!]_{\mathcal{C}'}$. But then also $s \in [\![p_t]\!]_{\mathcal{C}'}$ – in contradiction to our assumption. □

Example 3.2. Let $\Sigma = \{a, g, f\}$ where the symbols have arities 0, 1 and 2, respectively. Consider the three automaton clauses with path disequalities:

$$\begin{array}{lcl} p(f(X_1, X_2)) & \Leftarrow & q(X_1), q(X_2),\ X_1.(g,1) \neq X_2, X_2.(f,2) \neq a \\ q(g(X_1)) & \Leftarrow & q(X_1) \\ q(a) & \Leftarrow & \end{array}$$

As all variables from the heads always also occur in the bodies, we refrain from introducing the universal predicate $\top$ with its corresponding defining clauses.

Let us consider $d = 1$. Then the set of patterns is given by $F_1 = \{a, f(_,_), g(_)\}$. The construction of Lemma 3.3 results in the clauses:

$$\begin{array}{lcl} p_{f(_,_)}(f(X_1, X_2)) & \Leftarrow & q_{t_1}(X_1), q_{t_2}(X_2),\ X_1.(g,1) \neq X_2, X_2.(f,2) \neq a \\ q_{g(_)}(g(X_1)) & \Leftarrow & q_t(X_1) \\ q_a(a) & \Leftarrow & \end{array}$$

for $t_1, t_2, t \in F_1$.

The extra information attached to the predicates can be used to remove the labels from the paths in disequality constraints.

Example 3.3. Consider the clause:

$$p_{f(_,_)}(f(X_1, X_2)) \Leftarrow q_{g(_)}(X_1), q_{f(_,_)}(X_2),\ X_1.(g,1) \neq X_2, X_2.(f,2) \neq a$$

from Example 3.2. The pattern $g(_)$ of the index of the predicate from the first literal guarantees that for every fact $q_{g(_)}(s_1)$ that can be obtained, $s_1.\pi$ is defined. We therefore equivalently may replace the path $(g,1)$ by the unlabelled path 1. The same holds true for the second literal and the path $(f,2)$.

For the clause:

$$p_{f(_,_)}(f(X_1, X_2)) \Leftarrow q_{f(_,_)}(X_1), q_{f(_,_)}(X_2),\ X_1.(g,1) \neq X_2, X_2.(f,2) \neq a$$

from Example 3.2, we find that $X_1.(g,1)$ will always be undefined. Accordingly, the disequality containing $X_1.(g,1)$ can be discarded.

The example can be generalized. Using the information provided by the full patterns for every predicate occurring in the body of the generalized clause, we can in every path disequality $X.\pi \neq t$ occurring in the constraint, always either drop the labels in paths, or drop the disequality itself. Thus, we arrive at a simulation of a finite set of generalized automaton clauses using labelled paths in path disequalities by a finite set of generalized automaton clauses using unlabelled paths only.

We also remark that by increasing the depth of full patterns, we can provide enough information to conclude validity of a disequality $X_i.\pi \neq s$ for some ground term s. To do

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

so, we only need to perform the construction from Lemma 3.3 for $d_1 + d_2$, where d_1 is the maximal length of a path occurring in the set of clauses, and d_2 is the maximal depth of a ground term (an atom has depth 1).

Recalling Lemma 2.5, the following theorem therefore can be proven.

Theorem 3.1. Let $\mathcal{A}$ be a finite set of generalized automaton clauses with general term disequalities without auxiliary variables. Then finite sets $\mathcal{A}'$ and $\mathcal{A}''$ of generalized automaton clauses without auxiliary variables can be constructed such that

1. $\mathcal{A}'$ has (labelled) path disequalities;
2. $\mathcal{A}''$ has conjunctions of unlabelled-path disequalities $X_i.\pi_1 \neq X_j.\pi_2$,

where $[\![p]\!]_{\mathcal{A}} = [\![p]\!]_{\mathcal{A}'} = [\![p]\!]_{\mathcal{A}''}$ for all predicates p.

Proof. The finite set $\mathcal{A}'$ is constructed from $\mathcal{A}$ by decomposing in each clause of $\mathcal{A}$, each general term disequality into a disjunction of path disequalities as in Lemma 2.5 (using the identity homomorphism) without changing the least model. According to Lemma 2.6, we may w.l.o.g. even assume that $\mathcal{A}'$ uses conjunctive constraints only. Let $d \geqslant 0$ denote an upper bound to the sum of the maximum length of a path and the maximum depth of a ground term occurring in constraints of $\mathcal{C}'$. Now we apply the construction from Lemma 3.3 to construct for d a set of clauses $\mathcal{A}_1$ which for each predicate p of $\mathcal{A}'$ has predicates p_t, $t \in F_d$, so that $[\![p_t]\!]_{\mathcal{A}_1} = \{s \in [\![p]\!]_{\mathcal{A}'} \mid t \wedge s = s\}$. The set $\mathcal{A}_2$ then is obtained as the set of clauses

$$\mathcal{A}_2 = \mathcal{A}_1 \cup \{p(t_0) \Leftarrow \alpha, \phi \mid t \in F_d, p_t(t_0) \Leftarrow \alpha, \phi \in \mathcal{A}_1\}.$$

Now consider a clause $h \Leftarrow p_{1,t_1}(X_1), \ldots, p_{k,t_k}(X_k), \phi$ in $\mathcal{A}_2$. We use the patterns $t_1, \ldots, t_k$ in two ways. First, we can replace path disequalities using a path expression $X_i.\pi$, where $t_i.\pi$ is undefined, with true, i.e., remove it from the conjunction. Second, we can replace a disequality $X_i.\pi \neq s$ for some ground term s, with true (and thus remove it from the conjunction), if $t_i.\pi$ is defined, but $t_i.\pi \wedge s$ fails. If on the other hand, $t_i.\pi \wedge s$ succeeds, then the disequality cannot be satisfied by any ground substitution θ with $t_i \wedge (X_i\theta)$. Accordingly, the whole clause can be removed. For the remaining disequalities between labelled paths, we now safely can remove the labels to arrive at $\mathcal{A}''$. □

The set $\mathcal{A}''$ in Theorem 3.1 can be considered as the clausal representation of a $TA_{\neq,\mathrm{hom}}$ automaton as introduced in Godoy *et al.* (2010) (in the following called *Godoy automaton*). Thus, a Godoy automaton corresponds to a finite set of generalized automaton clauses which use as constraints conjunctions of disequalities of the form $X_i.\pi \neq X_j.\pi'$, where both X_i and X_j occur in the head of the respective clause, and π, π' are unlabelled paths. The states of the automaton correspond to predicates, while the transitions become clauses. By Theorem 5.1 of Godoy *et al.* (2010), emptiness is decidable for Godoy automata. Later, this problem has been shown to be DEXPTIME-complete (Creus *et al.* 2012). In light of Theorem 3.1, we therefore have the following:

Corollary 3.1. Let $\mathcal{A}$ be a finite set of generalized automaton clauses with general term disequalities without auxiliary variables. Then it is decidable for every predicate q whether or not $[\![q]\!]_{\mathcal{A}} = \varnothing$.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

In Lemma 5.2, we will show that auxiliary variables do not add to the expressiveness of finite sets of generalized automaton clauses. Therefore, we obtain the following:

Theorem 3.2. Let $\mathcal{A}$ be a finite set of generalized automaton clauses with general term disequalities, possibly with auxiliary variables. Then finite sets $\mathcal{A}'$ and $\mathcal{A}''$ of generalized automaton clauses without auxiliary variables can be constructed such that

1. $\mathcal{A}'$ has (labelled) path disequalities;
2. $\mathcal{A}''$ has conjunctions of unlabelled-path disequalities $X_i.\pi_1 \neq X_j.\pi_2$,

where $[\![p]\!]_{\mathcal{A}} = [\![p]\!]_{\mathcal{A}'} = [\![p]\!]_{\mathcal{A}''}$ for all predicates p.

Concerning the simulation of unlabelled-path constraints by means of labelled-path constraints, we remark that every unlabelled-path disequality is equivalent to a finite conjunction of labelled-path disequalities – again under the assumption that Σ is finite. Since unlabelled-path disequalities can be expressed by means of labelled-path disequalities, and labelled-path disequalities are special cases of general term disequalities, all three classes of automaton clauses compared in Theorem 3.2 are equally expressive.
Term disequalities are special cases of general term disequalities. By Theorem 3.2, automaton clauses with term disequalities therefore can be simulated by means of automaton clauses with labelled-path or unlabelled-path disequalities. In the following, we show by a counter-example that the reverse simulation of automaton clauses with path disequalities by means of automaton clauses with term disequalities is not possible. For this, we use a complementation procedure for tree automata which transforms a tree automaton with constraints into a tree automaton with complementary constraints (equalities in our case) that recognizes the complement language. With the complementation technique at hand, the result is already indicated by the fact that universality is decidable for tree automata with term disequalities (Reuß and Seidl 2010) – while it is undecidable for tree automata with path disequalities. The latter follows, since emptiness for tree automata with path *equalities* only is undecidable (Godoy *et al.* 2010).

We present a specific language T which can be expressed as $[\![p]\!]_{\mathcal{A}_0}$ for a finite set $\mathcal{A}_0$ of automaton clauses with unlabelled-path disequalities but which cannot be characterized by means of finite sets of automaton clauses with term disequalities only. We prove that there does not exist a finite set of automaton clauses with *hom-disequalities* that defines T. The set T, taken from Seidl and Reuß (2012), is given by $T = [\![p]\!]_{\mathcal{A}_0}$ for a predicate p and the following set $\mathcal{A}_0$ of clauses:

$$\begin{array}{lcl} \top(a) & \Leftarrow & \\ \top(f(X_1, X_2)) & \Leftarrow & \top(X_1), \top(X_2) \\ p(f(X_1, X_2)) & \Leftarrow & \top(X_1), \top(X_2), X_1 \neq X_2.1 \end{array}$$

for $\Sigma = \{a, f\}$. Since there is only one constructor with arity > 0, namely f, we have omitted its label in the disequality constraint. We have the following:

$$T = \{f(t, a) \mid t \in \mathcal{T}_\Sigma\} \cup \{f(t, f(t_1, t_2)) \mid t, t_1, t_2 \in \mathcal{T}_\Sigma, t \neq t_1\}.$$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Intuitively, T cannot be specified by means of finite sets of automaton clauses with hom-disequalities since hom-disequalities cannot directly access specific subtrees independently of the labels in the tree. The proof of the following lemma can be found in the appendix:

Lemma 3.4. There is no finite set $\mathcal{A}$ of automaton clauses with hom-disequalities only that defines a predicate p with $[\![p]\!]_{\mathcal{A}} = T$.

Let us now show that hom-disequalities cannot be simulated by path disequalities, by presenting a specific language defined through a finite set of automaton clauses with hom-disequalities which cannot be defined by a finite set of automaton clauses with path disequalities only. Intuitively, path constraints can only express disequalities between subterms of at most a certain depth d specified as part of the corresponding path expression. Hom-disequalities, on the other hand, may disregard an unbounded number of constructors on top of the tree. Accordingly, the following example language taken from Reuß and Seidl (2012) is not expressible by means of path disequalities only.

Next, we show that there are also languages which can be specified by means of finite sets of automaton clauses with hom-disequalities which cannot be specified by means finite sets of automaton clauses with path disequalities. Let $\Sigma = \{a, s, f\}$ where a, s, and f have arities 0, 1 and 2, respectively. Let H be the tree homomorphism defined by $\mathrm{H}a = a$, $\mathrm{H}s = [X_1]_{\mathrm{H}}$ and $\mathrm{H}f = f([X_1]_{\mathrm{H}}, [X_2]_{\mathrm{H}})$, i.e., H only deletes the constructor s. Consider the language L given by

$$L = \{f(t_1, t_2) \mid \mathrm{H}t_1 \neq \mathrm{H}t_2\}$$

The following set $\mathcal{A}$ of automaton clauses with hom-disequalities defines L as $[\![p]\!]_{\mathcal{A}}$:

$$\begin{array}{lcl} \top(a) & \Leftarrow & \\ \top(s(X_1)) & \Leftarrow & \top(X_1) \\ \top(f(X_1, X_2)) & \Leftarrow & \top(X_1), \top(X_2) \\ p(f(X_1, X_2)) & \Leftarrow & \top(X_1), \top(X_2), [X_1]_{\mathrm{H}} \neq [X_2]_{\mathrm{H}} \end{array}$$

Lemma 3.5. There is no finite set of automaton clauses for L with path disequalities only.

Proof. This lemma has been proven in Reuß and Seidl (2012) for the case of the less general disequalities $t_1 \neq_{\mathrm{H}} t_2$ which were used in that paper. For convenience and for self-containedness, this proof is repeated here in a version that applies to the more general hom-disequalities that we use in this paper.

Assume for a proof by contradiction that a finite set $\mathcal{A}'$ of automaton clauses with path disequalities defining a predicate q with $[\![q]\!]_{\mathcal{A}'} = [\![p]\!]_{\mathcal{A}} = L$. Recalling Lemma 2.6, we may w.l.o.g., assume that $\mathcal{A}'$ has conjunctive constraints only. As in Lemma 3.4, we construct a finite set $\mathcal{B}$ of automaton clauses with complementary constraints, i.e., *path equalities*. Then the set $\mathcal{B}$ contains a predicate $\overline{q}$ which accepts the language $[\![\overline{q}]\!]_{\mathcal{B}} = \overline{L} = \{f(t_1, t_2) \mid \mathrm{H}t_1 = \mathrm{H}t_2\} \cup \{a\} \cup \{s(t) \mid t \in \mathcal{T}_\Sigma\}$. Let d be the maximal length of a path occurring in a constraint of $\mathcal{B}$. For a ground term t, let $r_{t1}, r_{t2}, \ldots$ denote the infinite sequence of terms defined by $r_{ti} = s^d(f(t, s^i(a)))$. Then, $\mathrm{H}r_{ti} = f(\mathrm{H}t, a)$. Hence, for $\mathrm{H}t \neq \mathrm{H}t'$, it holds that for all i, j, $\mathrm{H}r_{ti} = \mathrm{H}r_{tj}$ and $\mathrm{H}r_{ti} \neq \mathrm{H}r_{t'j}$, but for all paths π occurring in $\mathcal{B}$, $r_{ti}.\pi \neq r_{tj}.\pi$ if $i \neq j$. Note that there are infinitely many terms t whose homomorphic

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

images Ht are pairwise distinct. Since there are infinitely many corresponding sequences (r_{ti}) but only finitely many clauses, $\mathcal{B}$ must contain a clause

$$\overline{q}(f(X_1, X_2)) \quad \Leftarrow \quad q_1(X_1), q_2(X_2), \phi$$

such that for two terms t, t' with $\mathrm{H}t \neq \mathrm{H}t'$, there are two terms t_1, t_2 from the sequence (r_{ti}) and two terms t_3, t_4 from the sequence $(r_{t'i})$ such that both $f(t_1, t_2)$ and $f(t_3, t_4)$ are in $[\![\overline{q}]\!]_{\mathcal{B}}$ by application of this clause. Especially, $t_1, t_3 \in [\![q_1]\!]_{\mathcal{B}}$ and $t_2, t_4 \in [\![q_2]\!]_{\mathcal{B}}$. Recall that ϕ is a conjunction of path equalities $Y.\pi = Z.\pi'$ for $Y, Z \in \{X_1, X_2\}$ and (labelled) paths π of length at most d. In particular, since the pre-condition of the clause is satisfied for the substitution $\theta = \{X_1 \mapsto t_1, X_2 \mapsto t_2\}$, every path π must be defined for t_1 or t_2 and therefore must equal $(s, 1)^i$ for some $0 \leqslant i \leqslant d$. Let us investigate which kinds of equality could possibly occur in ϕ.

— $X_i.\pi = X_{3-i}.\pi'$ cannot occur since $t_1.\pi$ (for a path π of length at most d) is either undefined (which is impossible since $\theta = \{X_1 \mapsto t_1, X_2 \mapsto t_2\}$ satisfies ϕ) or has the form $s^i(t, s^j(a))$, while $t_3.\pi'$ is either undefined (again impossible because $\{X_1 \mapsto t_3, X_2 \mapsto t_4\} \models \phi$) or has the form $s^i(t', s^j(a))$. Hence, the equality could only be satisfied if $t = t'$, which is not the case.

— $X_i.\pi = g$ for a ground term g cannot occur either, since we have $t_1.\pi \neq t_3.\pi$ and $t_2.\pi \neq t_4.\pi$ for all paths π of depth at most d.

— $X_i.(s, 1)^k = X_i.(s, 1)^j$ is always false for $k \neq j$. Therefore, only the case $k = j$ may occur.

Therefore, ϕ must be equivalent to a conjunction of equalities of the form $X_i.\pi = X_i.\pi$ with $\pi = (s, 1)^k$, $k \leqslant d$. But then the clause also accepts the term $f(t_1, t_4)$ for $\overline{q}$ – contradiction, since $\mathrm{H}t_1 = f(t, a) \neq f(t', a) = \mathrm{H}t_4$ implies that $f(t_1, t_4) \notin [\![\overline{q}]\!]_{\mathcal{B}}$. □

Automaton clauses with term disequalities can be simulated by automaton clauses with path disequalities according to Theorem 3.2. Choosing the trivial tree homomorphism, on the other hand, it follows that also hom-disequalities can simulate term disequalities. Thus, we conclude that automaton clauses with hom-disequalities are incomparable to automaton clauses with path disequalities, while both classes are more expressive than automaton clauses with term disequalities only. Finally, since path disequalities are incomparable to hom-disequalities, automaton clauses with hom-path disequalities properly extend both automaton clauses with path disequalities and automaton clauses with hom-disequalities.

4. Normalization of $\mathcal{H}_1$-clauses with hom-disequalities

In this section, we present the general framework of $\mathcal{H}_1$-normalization in presence of constraints, instantiated with hom-disequalities, i.e., disequalities of terms modulo a given tree homomorphism H. The base case where H yields identity has been considered in Seidl and Reuß (2011) – $\mathcal{H}_1$ with term disequalities.

For a non-trivial homomorphism, the procedure has to be adapted as described in Reuß and Seidl (2012). Here, we define hom-disequalities in a more general and more convenient way than in Reuß and Seidl (2012). This new definition allows for comparisons

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

like $[X]_{\mathrm{H}} \neq t$ where the topmost constructors of t need not correspond to any of the right-hand sides defined through H. The motivation for this new definition of hom-disequality is that it can smoothly be generalized to the *general hom-term disequalities* that we consider in Section 5 when combining paths and homomorphisms. Also, it is more convenient to use ordinary disequalities $l \neq r$ instead of special disequalities $l \neq_{\mathrm{H}} r$ as in Reuß and Seidl (2012).

Example 4.1. Consider the set of symbols $\Sigma = \{a, b, g, f\}$, where a, b are atoms and g, f are unary constructors. Let $\mathrm{H} = \{a \mapsto b, b \mapsto b, g \mapsto f(g([X_1]_{\mathrm{H}})), f \mapsto g([X_1]_{\mathrm{H}})\}$. Then the disequality

$$[X_3]_{\mathrm{H}} \neq f([X_4]_{\mathrm{H}})$$

has a right-hand side whose topmost constructor f does not correspond to the topmost constructors of any right-hand side of H, i.e., there is no symbol $s \in \Sigma$ such that $\mathrm{H}s = f([X_1]_{\mathrm{H}})$. For the substitution $\theta = \{X_3 \mapsto a, X_4 \mapsto a\}$, the disequality evaluates to true, since $b \neq f(b)$. For the substitution $\theta' = \{X_3 \mapsto g(a), X_4 \mapsto f(a)\}$, however, it evaluates to false, since both sides evaluate to $f(g(b))$.

The normalization procedure repeatedly applies three rules until the set of clauses becomes *saturated*. Each rule *adds* finitely many simpler clauses which are implied by the current set of clauses (denoted $\mathcal{C}$). *Saturation* is achieved if each application of a rule only adds clauses which are *abstractly subsumed* by corresponding clauses in $\mathcal{C}$. For a clause c of the form $p(t_0) \Leftarrow p_1(t_1), \ldots, p_r(t_r), \phi$, we define the *core* as the triple $\mathsf{core}(c) = (p(t_0), B, X)$, where $B = \{p_1(t_1), \ldots, p_r(t_r)\}$ is the set of literals occurring in the body, and X is the set of all variables occurring in c. Also, we refer to the constraint of c by $\mathsf{constr}(c) = \phi$. Then the clause c is *abstractly subsumed* by a set $\{c_1, \ldots, c_m\}$ of clauses iff $\mathsf{core}(c) = \mathsf{core}(c_1) = \ldots = \mathsf{core}(c_m)$ and the constraint of c abstractly implies the disjunction of the constraints of the c_i, i.e., $\mathsf{constr}(c) \Rightarrow^{\sharp} (\mathsf{constr}(c_1) \vee \ldots \vee \mathsf{constr}(c_m))$.

After saturation, the subset $\mathcal{N}$ of normal clauses is semantically equivalent to the whole set of clauses. The set $\mathcal{N}$ of normal clauses in turn can be transformed to tree automaton form by Lemma 2.8.

For a given set $\mathcal{C}$ of $\mathcal{H}_1$-clauses and the subset $\mathcal{N}$ of normal clauses in $\mathcal{C}$, the three rules are as follows:

— *Resolution*, which applies a resolution step with a normal clause to infer a clause where one literal in the body is replaced with a sequence of smaller literals.
— *Splitting*, which removes auxiliary variables from a given clause, i.e., variables which occur in the pre-condition, but not in the head.
— *Propagation*, which considers a clause $p(X_1) \Leftarrow q_1(X_1), \ldots, q_m(X_1), \phi$ where ϕ contains no variable other than X_1, and uses normal clauses for the predicates $q_1, \ldots, q_m$ to infer a normal clause for p.

Technically, these rules are defined as follows:

Resolution: Assume that $\mathcal{C}$ contains a clause

$$p(t_0) \Leftarrow \alpha_1,\ q(f(t_1, \ldots, t_k)),\ \alpha_2,\ \phi_1$$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

with $k \geqslant 0$. If $\mathcal{C}$ contains a normal clause $q(f(X_1,\ldots,X_k)) \Leftarrow \beta, \phi_2$, then the clause

$$p(t_0) \Leftarrow \alpha, \phi$$

is added where $\alpha = \alpha_1, (\beta\theta), \alpha_2$ and $\phi = \phi_1 \wedge (\phi_2\theta)$ for $\theta = \{X_1 \mapsto t_1, \ldots, X_k \mapsto t_k\}$.

Splitting: Assume that $\mathcal{C}$ contains a simple clause $p(t) \Leftarrow \alpha, \phi$. W.l.o.g., let us assume that ϕ has satisfiable disequalities of the form $[X_i]_{\mathrm{H}} \neq s$ only. Assume that α, ϕ contains a variable X_j which is not mentioned in the term t. As the clause is simple, α can be re-arranged into a sequence $\alpha', q_1(X_j), \ldots, q_r(X_j)$, where α' does not contain X_j. Assume that ϕ contains n disequalities that mention X_j. Then consider the language $L = [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$. If $\mathrm{H}L$ contains at least $n+1$ elements, the clause

$$p(t) \Leftarrow \alpha', \phi'$$

is added where ϕ' is obtained from ϕ by replacing each disequality mentioning X_j with *true*.

If on the other hand, $\mathrm{H}(L) = \{s_1, \ldots, s_{n'}\}$ for some $n' \leqslant n$, then the clause

$$p(t) \Leftarrow \alpha', \phi'$$

is added where $\phi' = \phi[s_1/[X_j]_{\mathrm{H}}] \vee \ldots \vee \phi[s_{n'}/[X_j]_{\mathrm{H}}]$.

Propagation: Assume that $\mathcal{C}$ contains a simple clause

$$p(X_1) \Leftarrow q_1(X_1), \ldots, q_m(X_1), \phi,$$

where ϕ contains no variable other than X_1. If for each $i = 1,\ldots,m$, $\mathcal{C}$ contains a normal clause $q_i(t) \Leftarrow \beta_i, \phi_i$ (sharing the same term t in the heads), the clause

$$p(t) \Leftarrow \beta_1, \ldots, \beta_m, \phi'$$

is added where $\phi' = \phi[t/X_1] \wedge \phi_1 \wedge \ldots \wedge \phi_m$.

In the case of term disequalities (no tree homomorphism), the key issue is to decide whether the intersection $L = [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$ contains less than $n+1$ terms – and if so, to compute all terms of this set. In presence of a (non-trivial) tree homomorphism H, this is no longer sufficient. Instead, we must refer to the number of *images* of terms from L under H. By Corollary 4.3, however, $(n+1)$-finiteness of $\mathrm{H}(L)$ is decidable. Moreover, if only $n' < n+1$ terms are in the set $\mathrm{H}(L)$, Corollary 4.3 provides us with an explicit enumeration of $\mathrm{H}(L)$.

The given rules generalize the corresponding rules for $\mathcal{H}_1$-normalization without constraints as considered, e.g., in Nielson *et al.* (2002), by a dedicated treatment of disequalities. Forgetting about disequalities, the rules can best be understood as a simple form of ordered resolution with splitting (Goubault-Larrecq 2005). The ordering enforces that only heads of normal clauses are resolved with literals in pre-conditions. Such resolution steps are applied in the resolution and the propagation rule.

Example 4.2. Consider again the voting protocol example (Example 1.1):

$$\begin{array}{lcl} \mathrm{valid}(X_1) & \Leftarrow & q(X_1, [\,]) \\ q(X_1, X_2) & \Leftarrow & q(X_1, :: (Y, X_2)),\ [X_1]_{\mathrm{H}} \neq [Y]_{\mathrm{H}} \\ q(X_1, X_2) & \Leftarrow & \mathrm{is_vote}(X_1), \mathrm{votes}(X_2) \end{array}$$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

where $\mathrm{H}(\mathrm{vote}) = [X_1]_{\mathrm{H}}$, while for all constructors g of arity $k \geqslant 0$ with $g \neq \mathrm{vote}$, $\mathrm{H}(g) = g([X_1]_{\mathrm{H}}, \ldots, [X_k]_{\mathrm{H}})$. For ease of comparison, we leave the constructor of q implicit, as explained in Section 2, and understand the given set of clauses as a set of $\mathcal{H}_1$-clauses with hom-disequalities. We add the following normal clauses:

$$\begin{array}{rcl@{\qquad}rcl}
\mathrm{empty}([\,]) & \Leftarrow & & h_b(\mathrm{pers}(X_1, X_2)) & \Leftarrow & n_{\mathrm{bob}}(X_1), \mathrm{age}_{25}(X_2) \\
\mathrm{age}_{15}(15) & \Leftarrow & & h_a(\mathrm{pers}(X_1, X_2)) & \Leftarrow & n_{\mathrm{alice}}(X_1), \mathrm{age}_{15}(X_2) \\
\mathrm{age}_{25}(25) & \Leftarrow & & p_{\mathrm{bob}}(\mathrm{vote}(X_1, X_2)) & \Leftarrow & h_b(X_1), \mathrm{yesorno}(X_2) \\
n_{\mathrm{alice}}(\mathrm{alice}) & \Leftarrow & & p_{\mathrm{alice}}(\mathrm{vote}(X_1, X_2)) & \Leftarrow & h_a(X_1), \mathrm{yesorno}(X_2) \\
n_{\mathrm{bob}}(\mathrm{bob}) & \Leftarrow & & \mathrm{votes}(::(X_1, X_2)) & \Leftarrow & p_{\mathrm{bob}}(X_1), v'(X_2) \\
\mathrm{yesorno}(\mathrm{yes}) & \Leftarrow & & v'(::(X_1, X_2)) & \Leftarrow & p_{\mathrm{alice}}(X_1), \mathrm{empty}(X_2) \\
\mathrm{yesorno}(\mathrm{no}) & \Leftarrow & &&&
\end{array}$$

that allow to fill the list of already submitted votes with entries $\mathrm{vote}(\mathrm{pers}(\mathrm{alice}, 15), _)$ and $\mathrm{vote}(\mathrm{pers}(\mathrm{bob}, 25), _)$ where $_$ represents one of the atoms $\mathrm{yes}, \mathrm{no} \in \Sigma$.

Resolving the first two clauses of this example with the third one for the substitution $X_2 \mapsto [\,]$ and $X_2 \mapsto ::(Y, X_2)$, respectively, yields the following clauses:

$$\begin{array}{rcl}
\mathrm{valid}(X_1) & \Leftarrow & \mathrm{is_vote}(X_1), \mathrm{votes}([\,]) \\
q(X_1, X_2) & \Leftarrow & \mathrm{is_vote}(X_1), \mathrm{votes}(::(Y, X_2)), [X_1]_{\mathrm{H}} \neq [Y]_{\mathrm{H}}
\end{array}$$

With the clause $\mathrm{votes}(::(X_1, X_2)) \Leftarrow p_{\mathrm{bob}}(X_1), v'(X_2)$, the second new clause can further be resolved to

$$q(X_1, X_2) \Leftarrow \mathrm{is_vote}(X_1), p_{\mathrm{bob}}(Y), v'(X_2), [X_1]_{\mathrm{H}} \neq [Y]_{\mathrm{H}}$$

In order to split the variable Y, we observe that $\mathrm{H}[\![p_{\mathrm{bob}}]\!]_{\mathcal{N}} = \{\mathrm{pers}(\mathrm{bob}, 25)\}$. Therefore, the clause

$$q(X_1, X_2) \Leftarrow \mathrm{is_vote}(X_1), v'(X_2), [X_1]_{\mathrm{H}} \neq \mathrm{pers}(\mathrm{bob}, 25)$$

is added. This new normal clause in turn enables two more resolution steps for the first two clauses of this example, yielding

$$\begin{array}{l}
\mathrm{valid}(X_1) \Leftarrow \mathrm{is_vote}(X_1), v'([\,]), \quad [X_1]_{\mathrm{H}} \neq \mathrm{pers}(bob, 25) \\
q(X_1, X_2) \Leftarrow \mathrm{is_vote}(X_1), v'(::(Y, X_2)), \; [X_1]_{\mathrm{H}} \neq \mathrm{pers}(\mathrm{bob}, 25), [X_1]_{\mathrm{H}} \neq [Y]_{\mathrm{H}}
\end{array}$$

for the substitution $X_2 \mapsto [\,]$ and $X_2 \mapsto ::(Y, X_2)$, respectively. Another resolution step with the clause $v'(::(X_1, X_2)) \Leftarrow p_{\mathrm{alice}}(X_1), \mathrm{empty}(X_2)$ allows to deduce

$$\begin{array}{rcll}
q(X_1, X_2) & \Leftarrow & \mathrm{is_vote}(X_1), p_{\mathrm{alice}}(Y), \mathrm{empty}(X_2), & [X_1]_{\mathrm{H}} \neq \mathrm{pers}(\mathrm{bob}, 25), \\
& & & [X_1]_{\mathrm{H}} \neq [Y]_{\mathrm{H}}
\end{array}$$

with substitution $X_1 \mapsto Y$. To the last clause, again splitting can be applied in order to replace the pre-condition $p_{\mathrm{alice}}(Y), [X_1]_{\mathrm{H}} \neq [Y]_{\mathrm{H}}$ with a disjunction of constraints not containing Y. As is the case with p_{bob}, we find a single term in the homomorphic image of $[\![p_{\mathrm{alice}}]\!]_{\mathcal{N}}$, namely $\mathrm{pers}(\mathrm{alice}, 15)$. Thus, the clause

$$\begin{array}{rcll}
q(X_1, X_2) & \Leftarrow & \mathrm{is_vote}(X_1), \mathrm{empty}(X_2), & [X_1]_{\mathrm{H}} \neq \mathrm{pers}(\mathrm{bob}, 25), \\
& & & [X_1]_{\mathrm{H}} \neq \mathrm{pers}(\mathrm{alice}, 15)
\end{array}$$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

is added. Again the obtained normal clause enables two resolution steps for the first two clauses of the example, yielding

$$\begin{array}{lcl} \mathrm{valid}(X_1) & \Leftarrow & \mathrm{is_vote}(X_1), \mathrm{empty}([\,]), \phi, \\ q(X_1, X_2) & \Leftarrow & \mathrm{is_vote}(X_1), \mathrm{empty}(::(Y, X_2)), [X_1]_{\mathrm{H}} \neq [Y]_{\mathrm{H}}, \phi, \end{array}$$

where ϕ abbreviates the conjunction $[X_1]_{\mathrm{H}} \neq \mathrm{pers}(\mathrm{bob}, 25), [X_1]_{\mathrm{H}} \neq \mathrm{pers}(\mathrm{alice}, 15)$. Finally, a resolution step with the clause $\mathrm{empty}([\,]) \Leftarrow$ for the first of these two clauses results in the clause

$$\mathrm{valid}(X_1) \Leftarrow \mathrm{is_vote}(X_1), \phi$$

expressing that a vote is valid only if the voter is neither Alice nor Bob.

4.1. *Termination*

Let us first convince ourselves that when starting from any finite set $\mathcal{C}$ of $\mathcal{H}_1$-clauses, after a finite number of rule applications, a set $\mathcal{S}$ is obtained, where each further application of a rule results in a clause which is already abstractly subsumed by a subset of clauses in $\mathcal{S}$. For that, we proceed in two steps: First, we show that the set of cores encountered during $\mathcal{H}_1$-normalization is finite; and second, we show that for each triple (h, A, X) only finitely many clauses c are encountered with $\mathsf{core}(c) = (h, A, X)$.

Let $\bar{X}$ denote the set of all variables occurring in any clause from $\mathcal{C}$ and $k \geqslant 0$ the maximal arity of a predicate occurring in $\mathcal{C}$. Concerning the set of variables not occurring in the head of the non-normal clause, the application of the resolution rule may result in a subset, while the splitting rule strictly shrinks this set. Finally, the propagation rule results in normal clauses only, implying that all variables in the resulting clauses occur in the respective heads. Therefore, the set of variables occurring in any newly generated clause is always a subset of $\{X_1, \ldots, X_k\} \cup \bar{X}$. Since the number of predicates and constructors is finite, there are only finitely many distinct heads of clauses. Likewise, the number of literals occurring in pre-conditions is bounded since new literals $p(t)$ in bodies are only created for *subterms* t of terms already occurring in the original set $\mathcal{C}$ of clauses. Therefore, the set of cores of clauses occurring during the normalization of a finite set $\mathcal{C}$ of $\mathcal{H}_1$-clauses, is finite.

We recall that a clause c is *abstractly subsumed* by clauses $c_1, \ldots, c_r$ iff the core of c equals the core of c_j for each j, and $\mathsf{constr}(c) \Rightarrow^{\sharp} (\mathsf{constr}(c1) \vee \ldots \vee \mathsf{constr}(c_r))$. For each occurring core (h, A, X), the normalization procedure generates a sequence

$$\phi_0 \vee \phi_1 \vee \phi_2 \vee \ldots$$

of constraints ϕ_i of clauses generated via the application of the three normalization rules. Based on Theorem 3 from Müller-Olm *et al.* (2005), we have shown in Seidl and Reuß (2011) the following lemma for arbitrary disjunctive sequences of term disequalities.

Lemma 4.1. For every sequence $\psi_i, i \geqslant 1$, of finite conjunctions of term disequalities over a finite set of variables, there exists some $m \geqslant 1$ such that $\bigvee_{i=1}^{m} \psi_i$ is implied by ψ_j for every $j \geqslant 1$.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

In our case, the occurring constraints ϕ_i in general will be arbitrary positive Boolean combinations of hom-disequalities and not necessarily just conjunctions. Nonetheless, we have the following:

Lemma 4.2. For every sequence $\phi_i, i \geqslant 1$, of positive Boolean combinations of term disequalities over a finite set of variables, there exists some $m \geqslant 1$ such that $\bigvee_{i=1}^{m} \phi_i$ is implied by ϕ_j for every $j \geqslant 1$.

Proof. Each constraint ϕ_i can be represented by a disjunction $\psi_{i1} \vee \ldots \vee \psi_{ir_i}$ where $\psi_{i1}, \ldots, \psi_{ir_i}$ are all conjunctions of term disequalities. Applying Lemma 4.1 to the sequence $\psi_{ij}, 1 \leqslant i, 1 \leqslant j \leqslant r_i$, we find some m so that $\bigvee_{i=1}^{m} \phi_i$ is implied by ψ_{ij} for every i and j. Thus, the disjunction is also implied by every disjunction $\psi_{j1} \vee \ldots \vee \psi_{jr_j} = \phi_j$. □

We observe that the assertion of Lemma 4.2 remains true for hom-disequalities as well as for general hom-term disequalities if we replace implication with abstract implication.

Corollary 4.1. For every sequence $\phi_i, i \geqslant 1$, of positive Boolean combinations of general hom-term disequalities over a finite set of hom-paths, there exists some $m \geqslant 1$ such that $\bigvee_{i=1}^{m} \phi_i$ is abstractly implied by ϕ_j for every $j \geqslant 1$.

Proof. Let $[X_{i_1}]_{\mathrm{H}}.\pi_1, \ldots, [X_{i_r}]_{\mathrm{H}}.\pi_r$ be the sequence of distinct hom-paths occurring in $\phi_i, i \geqslant 1$. Let E denote the set of indices l where $\pi_l = \epsilon$. Consider a fixed $I \subseteq \{1, \ldots, r\}$ with $E \subseteq I$. Let $\bar{\phi}_{i,I}, i \geqslant 1$, denote the corresponding sequence of positive Boolean combinations of ordinary term disequalities which is obtained from $\phi_i, i \geqslant 1$, by replacing each atomic disequality which mentions $[X_{i_l}]_{\mathrm{H}}.\pi_l$ for some $l \notin I$, with *true*, while in all other disequalities, for all $l \in I$, $[X_{i_l}]_{\mathrm{H}}.\pi_l$ is replaced with Y_l. The resulting sequence consists of term disequalities only. By Lemma 4.2, there is some $m_I \geqslant 1$ such that $\bigvee_{i=1}^{m_I} \bar{\phi}_{i,I}$ is implied by $\bar{\phi}_{j,I}$ for every $j \geqslant 1$. Let m denote the maximum of the indices m_I, $E \subseteq I \subseteq \{1, \ldots, r\}$. Then for every $j \geqslant 1$, for all $I \subseteq \{1, \ldots, r\}$ with $E \subseteq I$, $\bar{\phi}_{j,I}$ implies $\bigvee_{i=1}^{m_I} \bar{\phi}_{i,I}$ and therefore also $\bigvee_{i=1}^{m} \bar{\phi}_{i,I}$, and the assertion of the lemma follows. □

Example 4.3. The disjunction

$$[X_1]_{\mathrm{H}} \neq a \vee [X_1]_{\mathrm{H}} \neq h([X_2]_{\mathrm{H}})$$

abstractly is implied by true and thus also abstractly implied by any conjunction of hom-disequalities. The disjunction

$$([X_1]_{\mathrm{H}} \neq a \wedge [X_1]_{\mathrm{H}} \neq h([X_2]_{\mathrm{H}})) \vee [X_2]_{\mathrm{H}} \neq h([X_1]_{\mathrm{H}})$$

does is not abstractly implied by true, as the disjunction

$$(Y_1 \neq a \wedge Y_1 \neq h(Y_2)) \vee Y_2 \neq h(Y_1)$$

is, e.g., not satisfied by the substitution $\theta = \{Y_1 \mapsto a, Y_2 \mapsto h(a)\}$.

After disjunction with the disequality $[X_1]_{\mathrm{H}} \neq b$, though, we obtain

$$([X_1]_{\mathrm{H}} \neq a \wedge [X_1]_{\mathrm{H}} \neq h([X_2]_{\mathrm{H}})) \vee [X_2]_{\mathrm{H}} \neq h([X_1]_{\mathrm{H}}) \vee [X_1]_{\mathrm{H}} \neq b$$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

is equivalent to the conjunction

$$\begin{array}{rl} & ([X_2]_{\mathrm{H}} \neq h([X_1]_{\mathrm{H}}) \vee [X_1]_{\mathrm{H}} \neq b \vee [X_1]_{\mathrm{H}} \neq a), \\ \wedge & ([X_2]_{\mathrm{H}} \neq h([X_1]_{\mathrm{H}}) \vee [X_1]_{\mathrm{H}} \neq b \vee [X_1]_{\mathrm{H}} \neq h([X_2]_{\mathrm{H}})), \end{array}$$

and therefore abstractly implied by true. Hence, this disjunction is abstractly implied by any further positive Boolean combination of hom-disequalities.

In order to prove that the normalization procedure terminates, assume for a contradiction that this were not the case, i.e., an infinite sequence of clauses $c_i, i \geqslant 1$, is generated by application of the normalization rules. Since there are only finitely many possible cores $(h, A, \mathcal{X})$ of clauses, there is an infinite sub-sequence $c_{i_k}, k \geqslant 1$, of clauses which all agree in their cores. Let $\phi_k, k \geqslant 1$, denote the sequence of their respective constraints. But then, by Corollary 4.1, there is some $m \geqslant 1$ such that ϕ_k abstractly implies $\phi_1 \vee \ldots \vee \phi_m$ for each $k \geqslant 1$. In particular, $c_{i_{m+1}}$ is abstractly subsumed by a subset of clauses from $\mathcal{C} \cup \{c_{i_1}, \ldots, c_{i_m}\}$ – in contradiction to our assumption. Therefore, no such infinite sequence of clauses $c_i, i \geqslant 1$, is possible – implying that normalization necessarily terminates.

4.2. *Correctness*

Now that we know, that normalization terminates, we can prove correctness of the normalization procedure by generalizing the proofs for non-constrained $\mathcal{H}_1$-clauses as well as for $\mathcal{H}_1$-clauses with term disequalities of Seidl and Reuß (2011). We remark that the main difference to the proof for plain term disequalities is that normalization (and thus also the proof) does now no longer rely on implication between constraints but on *abstract* implication. We have the following:

Theorem 4.1. Let $\mathcal{C}$ denote a finite set of $\mathcal{H}_1$-clauses with hom-disequalities. Then an equivalent finite set of normal $\mathcal{H}_1$-clauses with hom-disequalities can effectively be constructed.

More precisely, let $\mathcal{S}$ denote the set of clauses which results from $\mathcal{C}$ by exhaustively applying the rules *Resolution*, *Splitting* and *Propagation*. Let $\mathcal{N}$ denote the subset of all normal clauses in $\mathcal{S}$. Then for every predicate p occurring in $\mathcal{C}$,

$$[\![p]\!]_{\mathcal{N}} = [\![p]\!]_{\mathcal{S}} = [\![p]\!]_{\mathcal{C}} \tag{1}$$

holds.

Proof. Since $\mathcal{N} \subseteq \mathcal{S}$ as well as $\mathcal{C} \subseteq \mathcal{S}$, $[\![p]\!]_{\mathcal{N}} \subseteq [\![p]\!]_{\mathcal{S}}$ and also $[\![p]\!]_{\mathcal{C}} \subseteq [\![p]\!]_{\mathcal{S}}$ for all predicates p. In order to prove that the least models of $\mathcal{S}$ and $\mathcal{C}$ coincide, consider an arbitrary set $\mathcal{C}'$ of $\mathcal{H}_1$-clauses with hom-disequality constraints, and assume that a clause c is added by one of the normalization rules. We show that the least models of $\mathcal{C}'$ and $\mathcal{C}' \cup \{c\}$ coincide. Assume that c is of the form $p(t_0) \Leftarrow \alpha, \phi$ for some sequence α of literals and some constraint ϕ.

We prove that each fact which is provable after addition, also is provable without application of c. Assume that this were not the case. Consider a fact $p(t) \notin \mathcal{M}_{\mathcal{C}'}$ which can be obtained after addition of c, where, from all such facts, $p(t)$ is obtained by means

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

of a resolution tree of minimal size. Since the resolution tree for the fact $p(t)$ is of minimal size, the clause c occurs in the root of the resolution tree. This means that there is a ground substitution θ' so that $p(t) = p(t_0\theta')$. Furthermore, $\theta' \models \phi$, and each literal in $\alpha\theta'$ is already contained in $\mathcal{M}_{\mathcal{C}'}$. We perform a case distinction on the rule by which c has been produced.

Resolution: There exist a clause $p(t_0) \Leftarrow \alpha_1, q(f(t_1,\ldots,t_k)), \alpha_2, \phi_1$ and a normal clause $q(f(X_1,\ldots,X_k)) \Leftarrow \beta, \phi_2$ in $\mathcal{C}'$ so that the sequence of literals and the constraint of c are given by $\alpha = \alpha_1, \beta\theta, \alpha_2$ and $\phi = \phi_1 \wedge \phi_2\theta$, where $\theta = \{X_1 \mapsto t_1,\ldots,X_k \mapsto t_k\}$. In particular, this means that $\theta' \models \phi_2\theta$ and thus, $\bar{\theta} = \{X_1 \mapsto t_1\theta',\ldots,X_k \mapsto t_k\theta'\} \models \phi_2$. Since also all literals in $(\beta\theta)\theta'$ are facts from $\mathcal{M}_{\mathcal{C}'}$, we conclude that the normal clause from $\mathcal{C}'$ allows to obtain the fact $q(f(t_1,\ldots,t_k))\theta'$ which therefore also is contained in $\mathcal{M}_{\mathcal{C}'}$. Since also $\theta' \models \phi_1$ and all literals in $(\alpha_1, \alpha_2)\theta'$ are in $\mathcal{M}_{\mathcal{C}'}$, we conclude by means of the first clause in the resolution, that $p(t_0\theta')$ also must be contained in $\mathcal{M}_{\mathcal{C}'}$ – in contradiction to our assumption.

Splitting: Assume that $\mathcal{C}'$ contains a simple clause $p(t_0) \Leftarrow \alpha', \phi'$. Assume that ϕ' has been brought into an equivalent form where it has satisfiable disequalities of the form $[X_i]_{\mathrm{H}} \neq s$ only. Assume further that α', ϕ' contains a variable X_j which is not mentioned in the term t_0. As the clause is simple, α' can be re-arranged into a sequence $\alpha'', q_1(X_j),\ldots,q_r(X_j)$, where α'' does not contain X_j.
Assume that ϕ' contains the disequalities $[X_{i_k}]_{\mathrm{H}} \neq s_k, k = 1,\ldots,n$ which mention X_j. Consider the language $L = [\![q_1]\!]_{\mathcal{N}'} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}'}$, where $\mathcal{N}'$ is the subset of normal clauses in $\mathcal{C}'$. Then the sequence of literals of the newly added clause c is given by $\alpha = \alpha''$. For the definition of the constraint of c, we distinguish two cases.

Case 1. HL contains at least $n+1$ elements. According to the splitting rule then the constraint ϕ of c is is obtained from ϕ' by replacing in ϕ' each disequality mentioning X_j with *true*.
We know that each literal in $\alpha''\theta'$ is contained in $\mathcal{M}_{\mathcal{C}'}$ and that $\theta \models \phi$. By assumption, there are terms $u_1,\ldots,u_{n+1}$ so that for each $k = 1,\ldots,n+1$ and each $i = 1,\ldots,r$, $u_k \in [\![q_i]\!]_{\mathcal{N}'}$ and $\mathrm{H}u_k$ are all pairwise distinct. Consider for $k = 1,\ldots,n+1$, the substitution θ_k which maps X_j to u_k and otherwise agrees with θ'. Then for each $k = 1,\ldots,n+1$, each literal in $\alpha'\theta_k$ is in $\mathcal{M}_{\mathcal{C}'}$. We claim furthermore, that there exists some $k = 1,\ldots,n+1$ so that θ_k satisfies all disequalities occurring in ϕ' and therefore also ϕ' itself.
In order to prove the claim, assume for a contradiction, that for each $k = 1,\ldots,n+1$, $\theta_k \not\models ([X_{i_l}]_{\mathrm{H}} \neq s_l)$ for some l, i.e., $\mathrm{H}(X_{i_l}\theta_k) = s_l\theta_k$. By the pigeonhole principle, there exists some l so that for $k \neq k'$, both $\theta_k \not\models ([X_{i_l}]_{\mathrm{H}} \neq s_l)$ and $\theta_{k'} \not\models ([X_{i_l}]_{\mathrm{H}} \neq s_l)$. First, assume that $i_l \neq j$. Then, $s_l\theta_k = \mathrm{H}(X_{i_l}\theta') = s_l\theta_{k'}$. Since s_l must contain an occurrence of $[X_j]_{\mathrm{H}}$, this is impossible. Now assume that $i_l = j$. Recall that then s_l cannot equal $[X_j]_{\mathrm{H}}$ as well, as then the disequality $[X_j]_{\mathrm{H}} \neq s_l$ would be unsatisfiable. But then X_j cannot occur at all on the right-hand side as the equality $[X_j]_{\mathrm{H}} = s_l$ is assumed to be satisfiable. But if s_l does not contain $[X_j]_{\mathrm{H}}$, then $\mathrm{H}u_k = s_l\theta' = \mathrm{H}u_{k'}$ – which is impossible.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

It follows that for some k, $\theta_k \models [X_{i_l}]_{\mathrm{H}} \neq s_l$ for all $l = 1, \ldots, n$. We claim that for this k, $\theta_k \models \phi'$.

In order to prove this second claim, we proceed by induction on the subformulas ψ of ϕ'. For each subformula ψ, let $\bar{\psi}$ denote the corresponding subformula of ϕ where each occurring disequality in ψ which mentions X_j, is replaced with *true*. We claim that $\theta' \models \bar{\psi}$ iff $\theta_k \models \psi$. This claim certainly holds for each atomic disequality, and if it holds for two subformulas ψ_1, ψ_2, then it also holds for their conjunction and disjunction. In summary, we therefore conclude that $\theta_k \models \phi'$. But then the simple clause from $\mathcal{C}'$ suffices to obtain the fact $p(t) = p(t_0\theta_k)$. Hence, $p(t)$ is already contained in $\mathcal{M}_{\mathcal{C}'}$ – in contradiction to our assumption.

Case 2. $\mathrm{H}(L) = \{s_1, \ldots, s_{n'}\}$, where $n' \leqslant n$. Then the constraint ϕ of c is given by $\phi = \phi'[s_1/[X_j]_{\mathrm{H}}] \vee \ldots \vee \phi'[s_{n'}/[X_j]_{\mathrm{H}}]$. By assumption, $\theta' \models \phi$. Therefore, $\theta' \models \phi'[s_k/[X_j]_{\mathrm{H}}]$ for some $1 \leqslant k \leqslant n'$. Let u denote a term in L so that $\mathrm{H}u = s_k$. We thus can define a ground substitution θ which maps X_j to u and agrees with θ' otherwise. Then all literals in $\alpha'\theta$ are contained in $\mathcal{M}_{\mathcal{C}'}$, and also $\theta \models \phi'$. Therefore, the simple clause from $\mathcal{C}'$ allows to obtain the fact $p(t)$. Accordingly, $p(t) \in \mathcal{M}_{\mathcal{C}'}$ – in contradiction to our assumption.

Propagation: Assume that there is a simple clause $p(X_1) \Leftarrow q_1(X_1), \ldots, q_m(X_1), \phi'$ in $\mathcal{C}'$ where ϕ' contains no variable other than X_1. Assume further that for each $i = 1, \ldots, m$, $\mathcal{C}'$ contains a normal clause $q_i(t_0) \Leftarrow \beta_i, \phi_i$ (sharing the same term t_0 from the head of c) so that the sequence of literals and the constraint of c is given by $\alpha = \beta_1, \ldots, \beta_m$ and $\phi = \phi'[t_0/X_1] \wedge \phi_1 \wedge \ldots \wedge \phi_m$. Thus, for each $i = 1, \ldots, m$, $\theta' \models \phi_i$, and each literal in $\beta_i\theta'$ is contained in $\mathcal{M}_{\mathcal{C}'}$. Accordingly, by the corresponding normal clause in $\mathcal{C}'$, we find that also $q_i(t_0\theta')$ are contained in $\mathcal{M}_{\mathcal{C}'}$. Since $\theta' \models \phi'[t_0/X_1]$, $\bar{\theta} \models \phi'$ where $\bar{\theta} = \{X_1 \mapsto t_0\theta'\}$. The ground substitution $\bar{\theta}$ applied to the given simple clause from $\mathcal{C}'$ therefore shows that also the fact $p(t) = p(X_1\bar{\theta})$ is contained in $\mathcal{M}_{\mathcal{C}'}$.

All in all, we obtain that $\mathcal{M}_{\mathcal{S}}$ and $\mathcal{M}_{\mathcal{C}}$ must coincide. It remains to prove that $\mathcal{M}_{\mathcal{S}} = \mathcal{M}_{\mathcal{N}}$ holds where $\mathcal{N}$ is the set of normal clauses contained in $\mathcal{S}$. In the following, let us call a resolution tree R *normal* if it contains normal clauses only. For a clause c_0, we define its *potential* as the function $\mathrm{pot}_{c_0} : \mathbb{N}_0 \to \mathbb{N}_0$, where $\mathrm{pot}_{c_0}(0)$ is defined as the number of variables which occur in the pre-condition of c_0 but which do *not* occur in the head of c_0, and for $i > 0$, $\mathrm{pot}_{c_0}(i)$ is defined as the number of literals $q(t)$ in the pre-condition of c_0 where t has exactly i occurrences of symbols from Σ. In other words, considered as a tree, t contains i non-variable *nodes*. We compare the potentials of two clauses c, c' as follows. $\mathrm{pot}_c > \mathrm{pot}_{c'}$ iff there is some $i \geqslant 0$ such that $\mathrm{pot}_c(i) \neq \mathrm{pot}_{c'}(i)$, and $\mathrm{pot}_c(i) > \mathrm{pot}_{c'}(i)$ for the maximal such i.

By induction on the structure of a resolution tree R, we prove that another resolution tree R' of the same fact $p(t)$ exists which uses normal clauses only, i.e., is normal. Thus, assume that R is of the form $\langle c, \theta \rangle (R_1, \ldots, R_m)$. By inductive hypothesis, there are normal resolution trees $R'_1, \ldots, R'_m$ for the same facts as $R_1, \ldots, R_m$, respectively. If the clause c is normal as well, we construct $R' = \langle c, \theta \rangle (R'_1, \ldots, R'_m)$ and are done. Therefore, assume that c is not normal. We perform a case distinction on the clause c. Assume that c is of the form $p(t_0) \Leftarrow \alpha, \phi$ so that $t = t_0\theta$.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Case 1 (Resolution). The sequence of literals α is of the form $\alpha = \alpha_1, q(f(t_1,\ldots,t_k)), \alpha_2$ for some predicate q and constructor f of arity $k \geqslant 0$. In particular, the fact $q(f(t_1,\ldots,t_k)\theta')$ is obtained by means of a normal resolution tree $R'_j = \langle c', \theta' \rangle (R''_1,\ldots,R''_r)$ for some j. Let $q(f(X_1,\ldots,X_k)) \Leftarrow \beta, \phi'$ be the clause c' at the root of R'_j. As c' is normal, the set of variables occurring in c' is given by $\{X_1,\ldots,X_k\}$. Therefore, $\theta' = \{X_1 \mapsto (t_1\theta),\ldots,X_k \mapsto (t_k\theta)\}$. Let $\theta_1 = \{X_1 \mapsto t_1,\ldots,X_k \mapsto t_k\}$. Then, $\theta' = \theta \circ \theta_1$. Since $\theta' \models \phi'$, it follows that $\theta \models \phi'\theta_1$. Moreover, the resolution tree R'_j provides us with resolution trees for the facts in $\beta\theta' = (\beta\theta_1)\theta$. Let c_1 denote the clause obtained from c and c' via the resolution rule. In particular, $\mathsf{constr}(c_1) = \phi \wedge \phi'\theta_1$. This means that $\mathcal{S}$ has a clause c'' with $\mathsf{core}(c'') = \mathsf{core}(c_1)$ and $\mathsf{constr}(c'') = \phi''$ which is satisfied by θ. Therefore, we can construct a resolution tree for $p(t)$ by replacing clause c at the root with the clause c'', by replacing R_i with R'_i for $i \neq j$, and by replacing the subtree R_j with the sequence of resolution trees $R''_1,\ldots,R''_r$. By that, we obtain a resolution tree for $p(t)$ where the only non-normal clause may be the clause c'' at the root. We remark, however, that $\mathrm{pot}_{c''} < \mathrm{pot}_c$ because the literal $q(f(t_1,\ldots,t_k))$ is not contained in the pre-condition of c_1 any more, while c_1 has additional literals of the form $q'(t_{j'})$ only, i.e., literals for proper subterms of $f(t_1,\ldots,t_k)$.

Case 2 (Splitting). The sequence of literals α only contains literals of the form $q(X_i)$. Assume w.l.o.g., that all disequalities occurring in ϕ are satisfiable and of the form $[X_i]_{\mathrm{H}} \neq t'$ for some variable X_i and some hom-term t'.

First, we claim that the clause c in the root of R can be chosen in such a way that there are no variables which occur in $\mathsf{constr}(c) = \phi$ only.

In order to see this, we proceed by induction on the number m of variables only occurring in ϕ. If $m = 0$, the claim is vacuously true. So assume that $m > 0$. Let X_j denote one variable that occurs in ϕ but nowhere else in c. Then the splitting rule provides us with a clause c_1 of the form $p(t_0) \Leftarrow \alpha, \phi'$, where ϕ' is obtained from ϕ by replacing each disequality mentioning X_j with *true*. Let θ' denote the restriction of θ to the variables occurring in c_1. Then, $\theta' \models \phi'$. Moreover, since $\mathcal{S}$ contains a subset of clauses which abstractly subsume c_1, $\mathcal{S}$ contains a clause c' with $\mathsf{core}(c_1) = \mathsf{core}(c')$ and $\theta' \models \mathsf{constr}(c')$. Therefore, we can replace the root label of R with $\langle c', \theta' \rangle$ to obtain a resolution tree for $p(t)$ with a clause c' which has a smaller set of variables only occurring in the constraint. We remark that $\mathrm{pot}_{c'} < \mathrm{pot}_c$ because the variable X_j does not occur in c' any more.

Therefore, now assume that for all variables X_i occurring in ϕ, there is also some literal in α which contains X_i, and that the variable X_j does not occur in the head of c. W.l.o.g., assume that α is of the form $\alpha = \alpha_1, q_1(X_j),\ldots,q_r(X_j), \alpha_2$ for some $r > 0$ where the sequence α_1, α_2 does not contain X_j. Let $u = \theta(X_j)$. For $k = 1,\ldots,r$, the resolution tree R then contains a subtree R_k which is a resolution tree for $q_k(u)$. Therefore, u is contained in the language $L = [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$.

Let $[X_{i_1}]_{\mathrm{H}} \neq t_1,\ldots,[X_{i_n}]_{\mathrm{H}} \neq t_n$ denote the disequalities in ϕ which mention the variable X_j. According to the definition of the splitting rule, we distinguish two subcases.

Case 2.1. HL contains at least $n+1$ elements. Let c_1 denote the clause obtained by the splitting rule for c and variable X_j. Then c_1 is given by $p(t_0) \Leftarrow \alpha_1, \alpha_2, \phi'$, where ϕ' is obtained from ϕ by replacing all disequalities $[X_{i_l}]_{\mathrm{H}} \neq t_l$ with *true*. Let θ' denote the restriction of the substitution θ to the variables occurring in c_1. Since $\mathcal{S}$ contains a subset

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

of clauses which abstractly subsume c_1, $\mathcal{S}$ has a clause c' so that $\mathsf{core}(c_1) = \mathsf{core}(c')$ and $\theta' \models \mathsf{constr}(c')$. Thus, we can construct a resolution tree R' for $p(t)$ which has $\langle c', \theta' \rangle$ as the root where the sequence of child trees is given by the normal trees R'_j for the child trees R_j of the root in R which refer to literals not mentioning X_j. While c' need not be a normal clause, again $\mathrm{pot}_{c'} < \mathrm{pot}_c$.

Case 2.2. $\mathrm{H}L = \{s_1 \ldots, s_{n'}\}$ for some $n' \leqslant n$. In particular, as $u \in L$, $\mathrm{H}u = s_{l'}$ for some $1 \leqslant l' \leqslant n'$. The splitting rule gives us a clause c_1 of the form $p(t_0) \Leftarrow \alpha_1, \alpha_2, \phi'$, where $\phi' = \phi[s_1/[X_j]_{\mathrm{H}}] \vee \ldots \vee \phi[s_{n'}/[X_j]_{\mathrm{H}}]$. Let θ' denote the restriction of θ to the variables occurring in c_1. By construction, $t = t_0\theta'$ and $\theta' \models \phi[s_{l'}/[X_j]_{\mathrm{H}}]$ and thus also $\theta' \models \phi'$. Again, $\mathcal{S}$ thus must contain a clause c' with $\mathsf{core}(c_1) = \mathsf{core}(c')$, where $\theta' \models \mathsf{constr}(c')$. Therefore, we can construct a resolution tree R' for $p(t)$ where the root label is given by $\langle c', \theta' \rangle$ and the sequence of child subtrees is given by the sequence of trees R'_j corresponding to child subtrees R_j from R for the literals different from $q_1(X_j), \ldots, q_r(X_j)$. As with Cases 1 and 2.1, all clauses in R' with the exception of c' are necessarily normal clauses, and again $\mathrm{pot}_{c'} < \mathrm{pot}_c$ for the clause c' at the root.

Case 3 (Propagation). The term t_0 in the head equals X_1, $\alpha = q_1(X_1), \ldots, q_r(X_1)$, and the only variable possibly occurring in ϕ is X_1. Then the resolution tree R for the ground fact $p(t)$ is of the form $\langle c, \theta \rangle (R_1, \ldots, R_r)$, where $\theta = \{X_1 \mapsto t\}$ and R_j is a resolution tree for $q_j(t)$, $j = 1, \ldots, r$. Assume that the term t is of the form $t = f(t_1, \ldots, t_k)$, and let θ' denote the substitution $\{X_1 \mapsto t_1, \ldots, X_k \mapsto t_k\}$.

First, consider the case where $r = 0$. Then the resolution tree R consists of a single node only. The propagation rule results in a clause c_1 given by $p(f(X_1, \ldots, X_k)) \Leftarrow \phi[f(X_1, \ldots, X_k)/X_1]$, and $\mathcal{S}$ has a clause c' with $\mathsf{core}(c_1) = \mathsf{core}(c')$ and $\theta' \models \mathsf{constr}(c')$. In particular, c' is normal and $\langle c', \theta' \rangle$ is a normal resolution tree for $p(t)$.

Now consider the case where $r > 0$. By construction, every R'_j is normal. Thus for each j, the root of R'_j is given by $\langle c_j, \theta' \rangle$ for a normal clause c_j of the form $q_j(f(X_1, \ldots, X_k)) \Leftarrow \alpha_j, \phi_j$, where $\theta' \models \phi_j$. Moreover, since $\theta \models \phi$, also $\theta' \models \phi[f(X_1, \ldots, X_k)/X_1]$ holds. Now consider the clause c' obtained via the propagation rule for c and $c_1, \ldots, c_r$, and a clause c'' in $\mathcal{S}$ so that $\mathsf{core}(c') = \mathsf{core}(c'')$ and $\theta' \models \mathsf{constr}(c'')$. Then, we construct a resolution tree R' for $p(t)$ with root label $\langle c'', \theta' \rangle$ and a sequence of subtrees at the root which is constructed as the concatenation of the sequence of subtrees at the roots in $R'_1, \ldots, R'_r$. Accordingly, R' is normal.

While Case 3 directly results in a normal resolution tree for $p(t)$, in Cases 1 and 2, we were able to construct resolution trees which either are normal or whose single non-normal clause is at the root. Moreover, the clause at the root of the new resolution tree has a strictly lower potential than the clause c at the root of the given resolution tree R. In Cases 1 and 2, the potential of the new root clause is the constant 0 function only if all variables occurring in the clause also occur in its head and all literals in the pre-condition are of the form $q(X_j)$. If additionally the head is of the form $p(f(X_1, \ldots, X_k))$, i.e., contains a constructor f, then the clause is a normal clause. If on the other hand, the head is of the form $p(X_1)$, then the new root clause has the form $p(X_1) \Leftarrow q_1(X_1), \ldots, q_r(X_1), \phi$, where the only variable possibly occurring in ϕ is X_1. In particular, the potential of the new root clause is 0 only if the clause is normal or Case 3 applies to the clause. Since $\mathrm{pot}_c(i) \geqslant 0$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

trivially holds for all clauses c and numbers i (and $\mathrm{pot}_c(i) > 0$ holds for only finitely many numbers i), we therefore may repeat the case distinction for the new resolution tree R' until the obtained resolution tree for $p(t)$ is normal. □

In summary, we have shown that for each finite set $\mathcal{C}$ of $\mathcal{H}_1$-clauses with hom-disequalities, a finite set $\mathcal{N}$ of normal $\mathcal{H}_1$-clauses with hom-disequalities can be effectively constructed which is equivalent to $\mathcal{C}$. For any predicate q, then emptiness of the set $[\![q]\!]_{\mathcal{N}}$ is decidable by Corollary 4.3. Therefore, we obtain the following:

Theorem 4.2. For a finite set $\mathcal{C}$ of $\mathcal{H}_1$-clauses with hom-disequalities and a predicate q, it is decidable whether $[\![q]\!]_{\mathcal{C}} \neq \varnothing$ holds.

Only a single building block is missing in our argument. For the application of the splitting rule, we have assumed that there is an effective procedure to decide for a given finite set $\mathcal{N}$ of normal clauses, predicates $q_1, \ldots, q_r$ and some $n \geqslant 1$ whether or not $\mathrm{H}([\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}})$ has at least n elements or, if this is not the case, to enumerate the elements therein. Clearly, instead of considering intersections $[\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$, we as well may just consider a single predicate (representing the conjunction $q_1 \wedge \ldots \wedge q_r$) and automaton clauses only.

4.3. *Bounding the number of terms*

In this subsection, we first establish that for every finite set of automaton clauses $\mathcal{A}$ with hom-disequalities, a finite set of *generalized* automaton clauses $\mathcal{A}_{\mathrm{H}}$ can be effectively constructed which uses *term disequalities* only, such that $\mathrm{H}[\![p]\!]_{\mathcal{A}} = [\![p]\!]_{\mathcal{A}_{\mathrm{H}}}$ for all predicates p occurring in $\mathcal{A}$. Second, we show that it is decidable for $\mathcal{A}_{\mathrm{H}}$, every predicate p, and number n, whether $|[\![p]\!]_{\mathcal{A}_{\mathrm{H}}}| < n$ – and thus, whether $|\mathrm{H}[\![p]\!]_{\mathcal{A}}| < n$. Finally, we wrap up the constructions into Corollary 4.3 which we relied upon when applying the splitting rule.

In this subsection, we consider finite sets of clauses of the form:

$$p(t_0) \Leftarrow p_1(X_1), \ldots, p_k(X_k), \phi \qquad (k \geqslant 0),$$

where $p, p_1, \ldots, p_k$ are unary predicates, t_0 is a term whose variables are contained in $\{X_1, \ldots, X_k\}$ and ϕ is a positive Boolean combination of term disequalities which may also only mention variables from $\{X_1, \ldots, X_k\}$. These have been called *generalized* automaton clauses in Section 2.4. The set of variables occurring in the head of a generalized automaton clause c is denoted by $\mathrm{hv}(c)$, while its complement with respect to $\{X_1, \ldots, X_k\}$, i.e., the set of variables occurring only in the body of c, is denoted $\mathrm{bv}(c)$. The number of atomic disequality constraints $t_1 \neq t_2$ in a clause c is denoted $\mathrm{dc}(c)$.

According to the following lemma, every finite set of automaton clauses with disequalities modulo a given tree homomorphism H can be transformed into a finite set of generalized automaton clauses with term disequalities for the images of terms.

Lemma 4.3. Let $\mathcal{A}$ be a finite set of automaton clauses with general hom-term disequalities w.r.t. a tree homomorphism H. Then a finite set $\mathcal{A}_{\mathrm{H}}$ of generalized automaton clauses with general term disequalities can be constructed such that for every predicate

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

p, $[\![p]\!]_{\mathcal{A}_\mathrm{H}} = \mathrm{H}[\![p]\!]_{\mathcal{A}}$. If the clauses in $\mathcal{A}$ use hom-disequalities only, then the clauses in $\mathcal{A}_\mathrm{H}$ use term disequalities only.

Proof. Let $\theta = \{[X_i]_\mathrm{H} \mapsto X_i \mid i = 1,\ldots,k\}$ denotes the substitution that transforms hom-variables occurring in constraints to ordinary variables. For every clause $c \in \mathcal{A}$ of the form

$$p(f(X_1,\ldots,X_k)) \Leftarrow p_1(X_1),\ldots,p_k(X_k),\phi,$$

the new set of clauses has the generalized automaton clause c' of the form

$$p(\mathrm{H}(f)\theta) \Leftarrow p(X_1),\ldots,p_k(X_k),(\phi\theta).$$

According to this construction, $\mathcal{A}_\mathrm{H}$ has term disequality constraints only, whenever $\mathcal{A}$ has hom-term disequalities only. Let us prove by induction on $i \geqslant 0$ that for all predicates q, $[\![q]\!]^i_{\mathcal{A}_\mathrm{H}} = \mathrm{H}[\![q]\!]^i_{\mathcal{A}}$. For the basis, we observe that $[\![q]\!]^0_{\mathcal{A}_\mathrm{H}} = \mathrm{H}[\![q]\!]^0_{\mathcal{A}} = \varnothing$ for all q. Let us proceed with the induction step. So assume for a fixed $i \geqslant 0$ that for all predicates q, $[\![q]\!]^i_{\mathcal{A}_\mathrm{H}} = \mathrm{H}[\![q]\!]^i_{\mathcal{A}}$.

Let us first prove that $t \in [\![p]\!]^{i+1}_{\mathcal{A}}$ implies $\mathrm{H}t \in [\![p]\!]^{i+1}_{\mathcal{A}_\mathrm{H}}$ for arbitrary ground terms t and predicates p. So assume that $t = f(t_1,\ldots,t_k) \in [\![p]\!]^{i+1}_{\mathcal{A}}$. Then there is a clause $c \in \mathcal{A}$ of the form $p(f(X_1,\ldots,X_k)) \Leftarrow p_1(X_1),\ldots,p_k(X_k),\phi$, such that for the ground substitution $\theta' = \{X_j \mapsto t_j \mid 1 \leqslant j \leqslant k\}$, $\theta' \models \phi$ and $t_j \in [\![p_j]\!]^i_{\mathcal{A}}$ for all $1 \leqslant j \leqslant k$. Using the induction hypothesis, the latter implies $\mathrm{H}t_j \in [\![p_j]\!]^i_{\mathcal{A}_\mathrm{H}}$ for all $1 \leqslant j \leqslant k$. Let $\theta_\mathrm{H} = \{[X_j]_\mathrm{H} \mapsto \mathrm{H}t_j \mid 1 \leqslant j \leqslant k\}$. Then, $\phi\theta'$ is identical to $\phi\theta_\mathrm{H}$ i.e., $\theta' \models \phi$ implies $\theta_\mathrm{H} \models \phi$. Let $\theta'_\mathrm{H} = \{X_j \mapsto \mathrm{H}t_j \mid 1 \leqslant j \leqslant k\}$. Then, $\theta_\mathrm{H} = \theta'_\mathrm{H} \circ \theta$. Hence, $\mathrm{H}(t) = \mathrm{H}(f(t_1,\ldots,t_k)) = \mathrm{H}(f)(\mathrm{H}t_1,\ldots,\mathrm{H}t_k) = \mathrm{H}(f)\theta_\mathrm{H} = \mathrm{H}(f(X_1,\ldots,X_k))\theta_\mathrm{H} \in [\![p]\!]^{i+1}_{\mathcal{A}_\mathrm{H}}$ through application of the clause $c' \in \mathcal{A}_\mathrm{H}$ of the form $p(\mathrm{H}(f)\theta) \Leftarrow p(X_1),\ldots,p_k(X_k),(\phi\theta)$, using the ground substitution θ'_H.

Now consider the reverse direction. Assume that there is a clause $c' \in \mathcal{A}_\mathrm{H}$ of the form $p(\mathrm{H}(f)\theta) \Leftarrow p(X_1),\ldots,p_k(X_k),(\phi\theta)$, such that some ground term t' is in $[\![p]\!]^{i+1}_{\mathcal{A}_\mathrm{H}}$ through application of c'. Let $\theta'_\mathrm{H} = \{X_j \mapsto t'_j \mid 1 \leqslant j \leqslant k\}$ be the applied ground substitution. This means that $t'_j \in [\![p_j]\!]^i_{\mathcal{A}_\mathrm{H}}$ for all $1 \leqslant j \leqslant k$. For each $1 \leqslant j \leqslant k$, by induction hypothesis, $t'_j \in \mathrm{H}[\![p_j]\!]^i_{\mathcal{A}}$, i.e., each t'_j is in $\mathrm{H}(\mathcal{T}_\Sigma)$. Therefore, $t' = (\mathrm{H}(f)\theta)\theta'_\mathrm{H}$ is also in $\mathrm{H}(\mathcal{T}_\Sigma)$. Moreover, there exists a ground term $t = f(t_1,\ldots,t_k)$ (all t_j in $\mathcal{T}_\Sigma$) such that $t' = \mathrm{H}(t)$. Using the induction hypothesis, we can even choose $t_j \in [\![p_j]\!]^i_{\mathcal{A}}$ (with $t'_j = \mathrm{H}t_j$) for all $1 \leqslant j \leqslant k$. Then, $\theta'_\mathrm{H} = \{X_j \mapsto \mathrm{H}t_j \mid 1 \leqslant j \leqslant k\}$. Since $\theta'_\mathrm{H} \models \phi\theta$, it follows that $\theta_\mathrm{H} \models \phi$ for $\theta_\mathrm{H} = \{[X_j]_\mathrm{H} \mapsto \mathrm{H}t_j \mid 1 \leqslant j \leqslant k\}$. The latter, however, is equivalent to $\theta' \models \phi$ for $\theta' = \{X_j \mapsto t_j \mid 1 \leqslant j \leqslant k\}$. We conclude that $t \in [\![p]\!]^{i+1}_{\mathcal{A}}$ using the clause $c \in \mathcal{A}$ which causes c' to be in $\mathcal{A}_\mathrm{H}$, and the ground substitution θ'. In summary, we have shown that for each $t' \in [\![p]\!]^{i+1}_{\mathcal{A}_\mathrm{H}}$, there exists a ground term $t \in [\![p]\!]^{i+1}_{\mathcal{A}}$ with $\mathrm{H}t = t'$. This completes the proof. □

Assume now that we are given a finite set $\mathcal{A}$ of generalized automaton clauses with term disequalities only, some predicate p occurring in $\mathcal{A}$, and some number $n \geqslant 1$. We are interested in the following problem:

— Decide whether $[\![p]\!]_{\mathcal{A}}$ has at least n terms.
— If there are only $m < n$ distinct terms in $[\![p]\!]_{\mathcal{A}}$, compute these m terms.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Here, we proceed along the lines of efficiently deciding n-finiteness of automata with term disequalities in Seidl and Reuß (2011). That algorithm, though, must be generalized, as now heads are no longer just single constructor applications and not all variables occurring in pre-conditions of a clause necessarily also occur in the head. For that, we establish an upper bound on the depth of resolution trees for deciding n-finiteness of $\mathcal{A}$'s predicates. This argument is based on a lemma which shows that each constraint ϕ of a clause c consisting of $m = \mathrm{dc}(c)$ disequalities can 'filter out' no more than m terms of any predicate in the pre-condition of the clause.

Lemma 4.4. Let $\mathcal{A}$ be a finite set of generalized automaton clauses with term disequalities, $c \in \mathcal{A}$ a clause of the form $p(t_0) \Leftarrow p_1(X_1), \ldots, p_k(X_k), \phi$. Let $d \geqslant 0$ and θ a ground substitution with $\theta \models \phi$ and $X_j\theta \in [\![p_j]\!]^d_{\mathcal{A}}$ for all $j \in \{1, \ldots, k\}$. Then, $|[\![p]\!]^{d+1}_{\mathcal{A}}| \geqslant \max\{|[\![p_j]\!]^d_{\mathcal{A}}| - \mathrm{dc}(c) \mid X_j \in \mathrm{hv}(c)\}$.

Proof. Let $X_j \in \mathrm{hv}(c)$. Since ϕ can be equivalently rewritten into disjunctive normal form, there is a conjunction $\psi = C_1 \wedge \ldots \wedge C_m$ of atomic constraints $C_1, \ldots, C_m$ with $m \leqslant \mathrm{dc}(c)$ so that $\psi \Rightarrow \phi$ and $\theta \models \psi$. W.l.o.g., assume that only the atomic constraints $C_1, \ldots, C_l$, $l \leqslant m$, in ψ mention X_j. As $\theta \models \psi$, each of the atomic constraints $C_i, i = 1, \ldots, l$, is satisfiable. By the pigeonhole principle, there are at most l terms $t \in [\![p_j]\!]^d_{\mathcal{A}}$ so that $\theta \oplus \{X_j \mapsto t\} \not\models (C_1 \wedge \ldots \wedge C_l)$. Accordingly, there are at least $|[\![p_j]\!]^d_{\mathcal{A}}| - l \geqslant |[\![p_j]\!]^d_{\mathcal{A}}| - m$ different trees t in $[\![p_j]\!]^d_{\mathcal{A}}$ which satisfy all C_i, $1 \leqslant i \leqslant l$ and thus $\theta \oplus \{X_j \mapsto t\} \models \phi$. Since $X_j \in \mathrm{hv}(c)$, each such t can be used in combination with the trees $X_i\theta, i \neq j$, to produce one distinct tree for $[\![q]\!]^{d+1}_{\mathcal{A}}$. □

The main theorem now establishes the bound on the depth of resolution trees needed to decide n-finiteness. This bound only depends on the number n, the number N of predicates occurring in $\mathcal{A}$, and the total number of disequalities in $\mathcal{A}$.

Theorem 4.3. Let $\mathcal{A}$ be a finite set of generalized automaton clauses with term disequality constraints. Assume that $\mathcal{A}$ uses N predicates. Let $d = \sum_{c \in \mathcal{A}} \mathrm{dc}(c)$. Then for all predicates p and every $n > 0$, $|[\![p]\!]_{\mathcal{A}}| < n$ if and only if $|[\![p]\!]^{N(d+n)}_{\mathcal{A}}| < n$.

Proof. Let P denote the set of predicates which occur in $\mathcal{A}$. For $i \geqslant 1$, let $\min[i] = \min\{|[\![p]\!]^{i-1}_{\mathcal{A}}| \mid p \in P, |[\![p]\!]^i_{\mathcal{A}}| > |[\![p]\!]^{i-1}_{\mathcal{A}}|\}$ with the convention that $\min \varnothing = \infty$. Direction '$\Rightarrow$' is trivial. Let $r = \max\{i_0 \geqslant 1 \mid \forall i \leqslant i_0.\ \min[i] < d+n\}$. For all $1 \leqslant i \leqslant r$, $\min[i] < d+n$ holds. We claim that $r \leqslant N(d+n)$.

For a contradiction, assume this were not the case. By the pigeonhole principle, assuming $|[\![p]\!]^{N(d+n)}_{\mathcal{A}}| < d+n$ for every $p \in P$, there exist $i_1 < i_2 \leqslant N(d+n) < r$ so that $\min[i_1] = \min[i_2]$ and the minimum in both cases is attained for the same predicate p. Since the number of elements in the set $[\![p]\!]^{i_2}_{\mathcal{A}}$ exceeds the number of elements in $[\![p]\!]^{i_1}_{\mathcal{A}}$, this is not possible.

Assume now for a proof of direction '$\Leftarrow$' by contradiction that there is a predicate p with $|[\![p]\!]_{\mathcal{A}}| \geqslant n$ but $|[\![p]\!]^{N(d+n)}_{\mathcal{A}}| < n$. Since $r \leqslant N(d+n)$, this implies that $|[\![p]\!]^r_{\mathcal{A}}| < n$.

For $i \geqslant 1$ and $q \in P$, let $\mathrm{Prod}[q, i]$ denote the set of clauses which occur in the root of a resolution tree of depth i for a ground fact $q(t)$ with $t \in [\![q]\!]^i_{\mathcal{A}} \setminus [\![q]\!]^{i-1}_{\mathcal{A}}$. Let i_1 denote the least index i so that $|[\![p]\!]^{i-1}_{\mathcal{A}}| < n$, but $|[\![p]\!]^i_{\mathcal{A}}| \geqslant n$ holds. By assumption, $i_1 > r$, and

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

$\mathrm{Prod}[p, i_1] \neq \varnothing$. For i starting from i_1 down to $r+1$, we inductively define clauses $c_i \in \mathcal{A}$ and sets of predicates $P_i \subseteq P$ as follows. We define c_{i_1} as any clause in $\mathrm{Prod}[p, i_1]$, and set P_{i_1} to $\varnothing$. For $i_1 > i \geqslant r+1$, assume that c_{i+1} is given by

$$q(t_0) \Leftarrow p_1(X_1), \ldots, p_l(X_l), \phi.$$

Then we set $P_i = P_{i+1} \cup \{q\}$, and choose c_i as any clause in $\mathrm{Prod}[q', i]$ where $q'(X_j)$ occurs in the body of the clause c_{i+1} for some variable X_j so that the following extra properties are met:

1. $X_j \in \mathrm{bv}(c_{i+1})$ implies $|[\![q']\!]^{i-1}_{\mathcal{A}}| \leqslant \mathrm{dc}(c_{i+1})$.
2. $X_j \in \mathrm{hv}(c_{i+1})$ implies $|[\![q']\!]^{i-1}_{\mathcal{A}}| \leqslant \mathrm{dc}(c_{i+1}) + |[\![q]\!]^{i}_{\mathcal{A}}|$.

In the following, we first show that for each $i_1 \geqslant i \geqslant r+1$ such a clause c_i exists. $c_{i_1} \in Prod[p, i_1]$ holds by assumption. This implies $\exists q \in P.\ c_{i_1} \in Prod[q, i_1]$. We thus may assume by induction hypothesis that there is a predicate $q \in P$ such that $c_{i+1} \in Prod[q, i+1]$. Let c_{i+1} be of the form $q(t_0) \Leftarrow p_1(X_1), \ldots, p_l(X_l), \phi$. Then there exists a ground substitution θ with $\theta \models \phi$, $X_m\theta \in [\![p_m]\!]^i_{\mathcal{A}}$ for all $1 \leqslant m \leqslant l$, and $t_0\theta \in [\![q]\!]^{i+1}_{\mathcal{A}} \setminus [\![q]\!]^i_{\mathcal{A}}$. Let

$$\mathcal{C}' = \bigcup \{Prod[p_{l'}, i] \mid X_{l'}\theta \in [\![p_{l'}]\!]^i_{\mathcal{A}} \setminus [\![p_{l'}]\!]^{i-1}_{\mathcal{A}}, 1 \leqslant l' \leqslant l\}.$$

Since $t_0\theta \notin [\![q]\!]^i_{\mathcal{A}}$ but $\theta \models \phi$, there exists at least one $m \in \{1, \ldots, l\}$ with $X_m\theta \in [\![p_m]\!]^i_{\mathcal{A}} \setminus [\![p_m]\!]^{i-1}_{\mathcal{A}}$. Hence, $\mathcal{C}' \neq \varnothing$.

Now assume for a proof of well definedness of c_i by contradiction that for all $c \in \mathcal{C}'$ of the form $p_{l'}(t'_0) \Leftarrow b$ (for some body b), either $X_{l'} \in \mathrm{bv}(c_{i+1})$ and $|[\![p_{l'}]\!]^{i-1}_{\mathcal{A}}| > \mathrm{dc}(c_{i+1})$, or $X_{l'} \in \mathrm{hv}(c_{i+1})$ and $|[\![p_{l'}]\!]^{i-1}_{\mathcal{A}}| > \mathrm{dc}(c_{i+1}) + |[\![q]\!]^i_{\mathcal{A}}|$. We distinguish two cases.

Case 1. t_0 is a ground term. Then for all $1 \leqslant l' \leqslant l$, $X_{l'} \in \mathrm{bv}(c_{i+1})$, and therefore either $\mathcal{C}'$ does not contain any clause with head predicate $p_{l'}$, or $|[\![p_{l'}]\!]^{i-1}_{\mathcal{A}}| > \mathrm{dc}(c_{i+1})$ holds. In both cases, we find terms $t_{l'} \in [\![p_{l'}]\!]^{i-1}_{\mathcal{A}}$ so that $\theta' \models \phi$ for $\theta' = \{X_{l'} \mapsto t_{l'} \mid 1 \leqslant l' \leqslant l\}$. Therefore, $t_0\theta$ is already contained in $[\![q]\!]^i_{\mathcal{A}}$ – contradiction.

Case 2. t_0 is not a ground term. Then there is at least one non-auxiliary variable $X_m \in \mathrm{hv}(c_{i+1})$, $1 \leqslant m \leqslant l$. Let $X_m \in \mathrm{hv}(c_{i+1})$ such that C' contains a clause with head predicate p_m. (If no such variable existed then we could conclude as in Case 1.) Then, by assumption, $|[\![p_m]\!]^{i-1}_{\mathcal{A}}| > \mathrm{dc}(c_{i+1}) + |[\![q]\!]^i_{\mathcal{A}}|$. If there exists a ground substitution θ', where $q(t_0\theta')$ has a resolution tree of depth $\leqslant i$ with the clause c_{i+1} at its root, then, by Lemma 4.4, $|[\![q]\!]^i_{\mathcal{A}}| \geqslant |[\![p_m]\!]^{i-1}_{\mathcal{A}}| - \mathrm{dc}(c_{i+1}) > |[\![q]\!]^i_{\mathcal{A}}|$ holds – contradiction. It remains to show that such a substitution θ' exists. We construct θ' as follows. Let ψ denote a conjunction of at most $\mathrm{dc}(c_{i+1})$ atomic disequalities so that $\psi \Rightarrow \phi$ and $\theta \models \psi$. The constraint ψ may be obtained from a DNF representation of ϕ. For $m' \in \{1, \ldots, l\}$, let $\psi_{m'}$ denote the conjunction of disequalities of ψ in which $X_{m'}$ occurs. Recall that if $X_{m'}\theta \in [\![p_{m'}]\!]^i_{\mathcal{A}} \setminus [\![p_{m'}]\!]^{i-1}_{\mathcal{A}}$ then by our assumption, $\mathrm{Prod}[p_{m'}, i] \neq \varnothing$ and $|[\![p_{m'}]\!]^{i-1}_{\mathcal{A}}| > \mathrm{dc}(c_{i+1})$ (be $X_{m'}$ a variable occurring in the head of c_{i+1} or not). In particular, $|[\![p_{m'}]\!]^{i-1}_{\mathcal{A}}| > 0$. Otherwise, $X_{m'}\theta \in [\![p_{m'}]\!]^{i-1}_{\mathcal{A}}$. In the second case, we define $\theta'(X_{m'})$ as $X_{m'}\theta$ while in the first case, by the pigeonhole principle, we can choose for $\theta'(X_{m'})$ elements in $[\![p_{m'}]\!]^{i-1}_{\mathcal{A}}$ such that $\theta' \models \psi_{m'}$ for each m'. We conclude that $\theta' \models \psi$, and therefore also $\theta' \models \phi$. Hence, the clauses c_i and therefore also the sets P_i exist.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Assume now that we have constructed a sequence $c_{i_1}, \ldots, c_{r+1}$. For a clause c, let $\mathrm{hp}(c)$ denote the predicate in the head literal of c. For $P' \subseteq P$, let $C(P') = \{c \in \mathcal{A} \mid \mathrm{hp}(c) \in P'\}$ denote the set of clauses of $\mathcal{A}$ whose head predicate is in P'. We claim that

$$(3) \qquad \forall i \in \{i_1, i_1 - 1, \ldots, r+1\}. |[\![\mathrm{hp}(c_i)]\!]^{i-1}_{\mathcal{A}}| < n + \sum_{c \in C(P_i)} \mathrm{dc}(c).$$

Since by construction, $|[\![\mathrm{hp}(c_i)]\!]^{i-1}_{\mathcal{A}}| < |[\![\mathrm{hp}(c_i)]\!]^{i}_{\mathcal{A}}|$, also $\min[i] < n + \sum_{c \in C(P_i)} \mathrm{dc}(c)$ holds. Therefore, also $\min[r+1] < n + \sum_{c \in C(P_{r+1})} \mathrm{dc}(c) \leqslant n + d$, while by definition of r, $\min[r+1] \geqslant n + d$ – contradiction. Thus, the claim implies the statement of the theorem.

Claim (3) is proven by induction on i ranging from i_1 down to $r+1$. By definition of i_1, $|[\![\mathrm{hp}(c_{i_1})]\!]^{i_1-1}_{\mathcal{A}}| = |[\![p]\!]^{i_1-1}_{\mathcal{A}}| < n = n + \sum_{c \in C(P_{i_1})} \mathrm{dc}(c)$, since $P_{i_1} = \varnothing$. For the induction step, we assume that we are given some $i \in \{r+1, \ldots, i_1\}$, and the induction hypothesis holds for all $i' \in \{i+1, \ldots, i_1 - 1\}$. Thus, in particular, $|[\![\mathrm{hp}(c_{i+1})]\!]^{i}_{\mathcal{A}}| < n + \sum_{c \in C(P_{i+1})} \mathrm{dc}(c)$ holds. Assume that the head predicate of c_{i+1} is q. By definition, c_i is a clause with some head predicate q' such that $q'(X_j)$ is a literal in the body of c_{i+1} for some variable X_j and $c_i \in \mathrm{Prod}[q', i]$. Furthermore, if $X_j \in \mathrm{bv}(c_{i+1})$ then $|[\![q']\!]^{i-1}_{\mathcal{A}}| \leqslant \mathrm{dc}(c_{i+1})$, and if $X_j \in \mathrm{hv}(c_{i+1})$, then $|[\![q']\!]^{i-1}_{\mathcal{A}}| \leqslant \mathrm{dc}(c_{i+1}) + |[\![q]\!]^{i}_{\mathcal{A}}|$. First, consider the case that $X_j \in \mathrm{bv}(c_{i+1})$. then we have $P_i = P_{i+1} \cup \{q\}$. Therefore, $\mathrm{dc}(c_{i+1}) \leqslant \sum_{c \in C(P_i)} \mathrm{dc}(c)$. Since $n > 0$, it follows that $|[\![\mathrm{hp}(c_i)]\!]^{i-1}_{\mathcal{A}}| = |[\![q']\!]^{i-1}_{\mathcal{A}}| \leqslant \mathrm{dc}(c_{i+1}) < n + \sum_{c \in C(P_i)} \mathrm{dc}(c)$. Now consider the case that $X_j \in \mathrm{hv}(c_{i+1})$. From $|[\![q']\!]^{i-1}_{\mathcal{A}}| \leqslant |[\![q]\!]^{i}_{\mathcal{A}}| + \mathrm{dc}(c_{i+1})$ and the induction hypothesis, it follows that $|[\![q']\!]^{i-1}_{\mathcal{A}}| < n + \sum_{c \in C(P_{i+1})} \mathrm{dc}(c) + \mathrm{dc}(c_{i+1})$.

We distinguish the following two cases. First, assume that $q \notin P_{i+1}$. Then, $\sum_{c \in C(P_{i+1})} \mathrm{dc}(c) + \mathrm{dc}(c_{i+1}) \leqslant \sum_{c \in C(P_{i+1})} \mathrm{dc}(c) + \sum_{c \in C(\{q\})} \mathrm{dc}(c) = \sum_{c \in C(P_i)} \mathrm{dc}(c)$, and the claim follows.

Second, assume that $q \in P_{i+1}$, i.e., $P_{i+1} = P_i$. By definition of the sets P_l ($r+1 \leqslant l \leqslant i_1$), this implies that there is a largest m with $i_1 \geqslant m > i+1$, $q = \mathrm{hp}(c_m)$ and $c_m \in \mathrm{Prod}[q, m]$. Since the sets P_l are monotonously increasing, $P_{m-1} \subseteq P_i$, and $\sum_{c \in C(P_{m-1})} \mathrm{dc}(c) \leqslant \sum_{c \in C(P_i)} \mathrm{dc}(c)$. We have $P_{m-1} = P_m \cup \{q\}$ where by maximality of m, $q \notin P_m$. As the induction hypothesis also holds for m, it follows that $|[\![q]\!]^{i+1}_{\mathcal{A}}| \leqslant |[\![q]\!]^{m-1}_{\mathcal{A}}| = |[\![\mathrm{hp}(c_m)]\!]^{m-1}_{\mathcal{A}}| < n + \sum_{c \in C(P_m)} \mathrm{dc}(c)$. By Lemma 4.4, $|[\![q']\!]^{i}_{\mathcal{A}}| \leqslant |[\![q]\!]^{i+1}_{\mathcal{A}}| + \mathrm{dc}(c_{i+1})$. Therefore, $|[\![q']\!]^{i}_{\mathcal{A}}| < n + \sum_{c \in C(P_m)} \mathrm{dc}(c) + \mathrm{dc}(c_{i+1})$. By maximality of m, we thus have $|[\![q']\!]^{i}_{\mathcal{A}}| < n + \sum_{c \in C(P_{m-1})} \mathrm{dc}(c)$. Thus, $|[\![\mathrm{hp}(c_i)]\!]^{i-1}_{\mathcal{A}}| = |[\![q']\!]^{i-1}_{\mathcal{A}}| \leqslant |[\![q']\!]^{i}_{\mathcal{A}}| < n + \sum_{c \in C(P_{m-1})} \mathrm{dc}(c) \leqslant n + \sum_{c \in C(P_i)} \mathrm{dc}(c)$ – and the claim follows. □

By Theorem 4.3, only a bounded number of resolution trees must be considered in order to determine whether or not $[\![p]\!]_{\mathcal{A}}$ has less than n elements, and if so, these resolution trees also allow to construct these. Therefore, we have the following:

Corollary 4.2. Let $\mathcal{A}$ be a finite set of generalized automaton clauses with term disequalities. Then for all predicates p and numbers $n > 0$, it can be effectively decided whether $|[\![p]\!]_{\mathcal{A}}| < n$. Moreover, if $|[\![p]\!]_{\mathcal{A}}| < n$, then the set $[\![p]\!]_{\mathcal{A}}$ can be effectively computed.

In summary, we conclude the following:

Corollary 4.3. Let $\mathcal{N}$ be a finite set of normal clauses with hom-disequalities. Then for a sequence of predicates $q_1, \ldots, q_r$ and some $n > 0$, it can be effectively decided whether for

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

$L = [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$, $\mathrm{H}(L)$ has less than n elements. Moreover, if this is the case, then this set can be effectively computed.

Proof. By Lemma 2.8, a finite set $\mathcal{A}$ of automaton clauses can be constructed using the same kind of constraints as $\mathcal{N}$ together with a predicate $[\{q_1, \ldots, q_r\}]$ so that $[\![[\{q_1, \ldots, q_r\}]]\!]_{\mathcal{A}} = [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}} = L$. By Lemma 4.3, a finite set $\mathcal{A}_{\mathrm{H}}$ of generalized automaton clauses with term disequalities only, can be constructed so that $[\![[\{q_1, \ldots, q_r\}]]\!]_{\mathcal{A}_{\mathrm{H}}} = \mathrm{H}([\![[\{q_1, \ldots, q_r\}]]\!]_{\mathcal{A}}) = \mathrm{H}(L)$. Finally, by Corollary 4.2, it can be decided whether or not $\mathrm{H}(L) = [\![\{[q_1, \ldots, q_r]\}]\!]_{\mathcal{A}_{\mathrm{H}}}$ contains less than n elements, and if so the elements in this set can be enumerated explicitly. □

We remark that as a special instance of Corollary 4.3, we obtain that emptiness of $[\![q]\!]_{\mathcal{N}}$ for a predicate q and a finite set $\mathcal{N}$ of normal clauses with hom-disequality constraints is decidable (just set $r = 1$ and $n = 1$).

5. $\mathcal{H}_1$-clauses with general hom-term disequalities

In this section, we extend the results from the last section to $\mathcal{H}_1$-clauses with hom-path disequalities, i.e., disequalities between paths of homomorphic images, or, more generally, general hom-term disequalities.

In Seidl and Reuß (2012), we have succeeded in normalizing finite sets of $\mathcal{H}_1$-clauses with path disequalities. Naturally, the question arises whether finite sets of $\mathcal{H}_1$-clauses still can be normalized if path disequalities modulo a homomorphism are allowed.

Here, we give an affirmative answer to that question. We build on work of Godoy *et al.* (2010) where it is shown that path disequalities can be added to automata that accept homomorphic images of regular languages, and emptiness is shown to be decidable for such automata. Recall that in Section 3, Corollary 3.1, we have translated this decision procedure into a decision procedure for finite sets of generalized automaton clauses with general term disequalities without auxiliary variables.

In the following, we proceed as in Section 4. The normalization procedure repeatedly applies rules *Resolution*, *Splitting* and *Propagation* until the set of clauses becomes saturated, i.e., if each application of a rule only adds clauses c which are *abstractly subsumed* by a subset of already available clauses. After saturation, the subset $\mathcal{N}$ of normal clauses is semantically equivalent to the whole set of clauses. Technically, the rules *Resolution* and *Propagation* are literally identical to the corresponding rules in the last section – only that now the constraints consist of *general* hom-term disequalities (instead of hom-disequalities only). The only rule which must be changed is the *splitting* rule. Assume that the current set of clauses is $\mathcal{C}$ where the current subset of normal clauses in $\mathcal{C}$ is given by $\mathcal{N}$.

Splitting: Assume that $\mathcal{C}$ contains a simple clause $p(t) \Leftarrow \alpha, \phi$ W.l.o.g., let us assume that ϕ has satisfiable hom-path disequalities only. Assume that α, ϕ contains a variable X_j which is not mentioned in the term t. As the clause is simple, α can be re-arranged into a sequence $\alpha', q_1(X_j), \ldots, q_r(X_j)$, where α' does not contain X_j. Consider the language $L = [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$. Assume that we are given finitely many ground terms

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

$t_1, \ldots, t_n \in \mathrm{H}L$ with $\phi' = \phi[t_1/[X_j]_\mathrm{H}] \vee \ldots \phi[t_n/[X_j]_\mathrm{H}]$, and assume that for all ground substitutions θ, $\theta \models \phi'$ iff $\theta \oplus \{X_j \mapsto u\} \models \phi$ for some $u \in L$. Then the clause

$$p(t) \Leftarrow \alpha', \phi'$$

is added.

In Section 5.2, we will explain that the required sequence $t_1, \ldots, t_n$ of ground terms can always be constructed (see Lemma 5.3).

For a proof of termination of the resulting normalization procedure, we proceed along the lines of the last section. No rule introduces fresh predicates or variables. Also, the set of term arguments occurring in literals remains finite. Therefore, the set of cores of clauses remains finite. The key issue for proving termination therefore is to convince ourselves that the number of hom-paths in constraints introduced by application of rules remains bounded. In the case of all rules, new atomic disequalities may only be created by applying a substitution to some variables of a disequality $t \neq t'$. By such a substitution, the length of paths is never increased. Therefore, the maximal length of an occurring path is bounded by the maximal length of paths in the initial set of clauses.

Example 5.1. Let $\mathrm{H} = \{a \mapsto a, g \mapsto g([X_1]_\mathrm{H}), f \mapsto g([X_1]_\mathrm{H})\}$, where $a, g, f \in \Sigma$ have arities 0, 1 and 2, respectively. Consider the following general hom-term disequality:

$$g([X_1]_\mathrm{H}.(g, 1)) \neq g(a).$$

When applying the ground substitution $\theta = \{X_1 \mapsto f(a, a)\}$ to the given disequality, we obtain

$$g((\mathrm{H}(X_1\theta)).(g, 1)) \neq g(a),$$

where $\mathrm{H}(X_1\theta) = \mathrm{H}(f(a, a)) = g(a)$. The left-hand side of the disequality thus is given by $g(\mathrm{H}(X_1\theta)).(g, 1)) = g(a)$. Hence, the path $(g, 1)$ has completely disappeared. In this case, the disequality is equivalent to $g(a) \neq g(a)$, i.e., is equivalent to *false*.

We conclude that the set of hom-paths occurring during normalization is necessarily finite. Therefore, the compactness argument for abstract implication between general term disequality constraints as provided by Lemma 4.1 can be applied to prove termination.

Likewise, the correctness argument from the last section can be almost literally be transferred. Altogether, we obtain the following:

Theorem 5.1. Let $\mathcal{C}$ denote a finite set of $\mathcal{H}_1$-clauses with general hom-term disequalities. Then an equivalent finite set of normal $\mathcal{H}_1$-clauses with general hom-term disequalities can be effectively constructed.

More precisely, let $\mathcal{S}$ denote the finite set of clauses which results from $\mathcal{C}$ by exhaustively applying the rules *Resolution*, *Splitting* and *Propagation*. Let $\mathcal{N}$ denote the subset of all normal clauses in $\mathcal{S}$. Then for every predicate p occurring in $\mathcal{C}$,

$$[\![p]\!]_\mathcal{N} = [\![p]\!]_\mathcal{S} = [\![p]\!]_\mathcal{C} \tag{2}$$

holds.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Let $\mathcal{N}$ denote a finite set of normal clauses, ϕ a general hom-term constraint, X_j a variable and $q_1, \ldots, q_r$ be predicates. Let $L = [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$. Our goal is to construct a sequence $t_1, \ldots, t_n$ of ground terms in HL so that for $\phi' = \phi[t_1/[X_j]_{\mathrm{H}}] \vee \ldots \vee \phi[t_n/[X_j]_{\mathrm{H}}]$, for all substitutions θ, $\theta \models \phi'$ iff $\theta \oplus \{X_j \mapsto u\} \models \phi$ for some $u \in L$.

For that, we extend the quantifier elimination algorithm of Seidl and Reuß (2012) to the case of hom-paths (see Subsection 5.2). This algorithm, however, relies on a procedure for deciding emptiness for a predicate defined by a finite set of normal clauses with general hom-term disequalities. By Lemma 4.3, this emptiness problem can be reduced to the emptiness problem for a predicate defined by a finite set of generalized automaton clauses with general term disequalities. A decision procedure for the latter problem is provided in the next subsection.

5.1. *Generalized automaton clauses with general term disequalities*

In this subsection, we show that emptiness is decidable for predicates defined by finite sets of generalized automaton clauses with general term disequalities. Recall that generalized automaton clauses may have arbitrary terms in their heads and their bodies may use auxiliary variables which do not occur in the respective heads. As tree homomorphisms may delete subtrees, the construction of Lemma 4.3 can result in auxiliary variables that do not occur in the head.

Example 5.2. Consider the tree homomorphism:

$$\mathrm{H} = \{a \mapsto b, b \mapsto b, g(X_1) \mapsto g([X_1]_{\mathrm{H}}), f(X_1, X_2) \mapsto f([X_1]_{\mathrm{H}}, f([X_1]_{\mathrm{H}}, a))\}.$$

Then the set of automaton clauses:

$$\begin{array}{lcl} p(a) & \Leftarrow & \\ p(f(X_1, X_2)) & \Leftarrow & p(X_1), p(X_2), [X_1]_{\mathrm{H}}.(f,2) \neq f([X_2]_{\mathrm{H}}.(g,1), [X_2]_{\mathrm{H}}) \end{array}$$

is transformed into the following set of generalized automaton clauses:

$$\begin{array}{lcl} p(b) & \Leftarrow & \\ p(f(X_1, f(X_1, a))) & \Leftarrow & p(X_1), p(X_2), X_1.(f,2) \neq f(X_2.(g,1), X_2) \end{array}$$

Variable X_2 no longer occurs in the head of the second clause, while variable X_1 occurs more than once.

In the case of term disequalities only, we have seen that emptiness (or, more generally, n-finiteness) can be directly decided – even in presence of auxiliary variables not occurring in the head (see Section 4.3). In presence of paths in disequalities, however, an argument based on the cardinalities of sets $[\![q]\!]^i_{\mathcal{A}}$ no longer suffices: there could be infinitely many terms which still agree in a particular subterm. Still, there is at least one transformation which allows us to remove auxiliary variables.

Lemma 5.1. Let $\mathcal{A}$ be a finite set of generalized automaton clauses with general term disequalities which does not use auxiliary variables. Let p be a predicate, and Y be a variable. For every finite conjunction of path disequalities ψ, a finite set $\mathcal{T} \subseteq [\![p]\!]_{\mathcal{A}}$ of ground terms can be effectively constructed such that with respect to $\mathcal{A}$, the disjunction

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

$\phi = \bigvee\{\psi[t/Y] \mid t \in \mathcal{T}\}$ is equivalent to the expression $\exists\, Y.(p(Y) \wedge \psi)$, i.e., for every ground substitution θ of the variables occurring in ϕ, $\theta \models \phi$ holds iff $\theta \models \psi[t/Y]$ holds for some ground term $t \in [\![p]\!]_{\mathcal{A}}$.

Proof. The construction is similar to the quantifier elimination algorithm introduced in Seidl and Reuß (2012). Here, we generalize the algorithm from Seidl and Reuß (2012) from automaton clauses to generalized automaton clauses by replacing the emptiness test for tree automata with path disequalities of Comon and Jacquemard (1994) with the emptiness test from Corollary 3.1.

W.l.o.g., we assume that the predicate p only occurs in the heads of clauses in $\mathcal{A}$. If this is not the case, then we add to $\mathcal{A}$ for every clause of the form $p(t_0) \Leftarrow \alpha, \phi'$ the clause $p'(t_0) \Leftarrow \alpha, \phi'$ for a fresh predicate p'. This results in $[\![p']\!]_{\mathcal{A}} = [\![p]\!]_{\mathcal{A}}$. Therefore, we may then replace every literal $p(X_j)$ occurring in the body of a clause from $\mathcal{A}$ by $p'(X_j)$.

First, we modify $\mathcal{A}$ so that only those terms of $[\![p]\!]_{\mathcal{A}}$ are accepted by p which satisfy each path disequality $Y.\pi \neq Y.\pi'$ occurring in ψ. W.l.o.g., let $\psi = Y.\pi_1 \neq Y.\pi'_1, \ldots, Y.\pi_m \neq Y.\pi'_m \wedge \psi'$, where ψ' does not contain any disequality of the form $Y.\pi \neq Y.\pi'$. We construct a finite set $\mathcal{A}'$ of generalized automaton clauses which is obtained from $\mathcal{A}$ by replacing every clause of the form $p(t_0) \Leftarrow \alpha, \phi'$ in $\mathcal{A}$, with the clause $p(t_0) \Leftarrow \alpha, \phi' \wedge \phi''$, where $\phi'' = t_0.\pi_1 \neq t_0.\pi'_1 \wedge \ldots \wedge t_0.\pi_m \neq t_0.\pi'_m$. By construction, all variables occurring in the body of a clause of $\mathcal{A}'$ also occur in the head. Moreover, $[\![p]\!]_{\mathcal{A}'} = \{t \in [\![p]\!]_{\mathcal{A}} \mid t.\pi_j \neq t.\pi'_j \text{ for all } j = 1,\ldots,m\}$. Then, for every ground substitution θ of the variables in ψ different from Y and every ground term $t \in \mathcal{T}_\Sigma$, $t \in [\![p]\!]_{\mathcal{A}}$ and $\theta \models \psi[t/Y]$ iff $t \in [\![p]\!]_{\mathcal{A}'}$ and $\theta \models \psi[t/Y]$. Therefore, in the following, we w.l.o.g., assume that the variable Y does not occur on both sides within the same disequality in ψ.

Our goal is to construct a finite set $\mathcal{T}$ of ground terms in $[\![p]\!]_{\mathcal{A}}$ such that for every ground substitution θ not mentioning Y, $\theta \models \psi[t/Y]$ for some $t \in [\![p]\!]_{\mathcal{A}}$ implies that $\theta \models \psi[t'/Y]$ for some $t' \in \mathcal{T}$. Assume that $\psi = Y.\pi_1 \neq u_1 \wedge \ldots \wedge Y.\pi_l \neq u_l \wedge \psi'$ where Y neither occurs in any of the u_i nor in ψ'. Then the set $\mathcal{T}$ is generated by the following algorithm. If $[\![p]\!]_{\mathcal{A}} = \varnothing$ (which is decidable by Corollary 3.1), then $\mathcal{T}$ is set to $\varnothing$ as well. Otherwise, an arbitrary ground term $s_0 \in [\![p]\!]_{\mathcal{A}}$ is selected and the set $\mathcal{T}$ is initialized with s_0.

Subsequently, the algorithm iterates through all permutations σ of $\{1,\ldots,l\}$. Now consider a permutation σ. Then the algorithm iterates through $i = 1,\ldots,l$. At each iteration, a ground term $s_{\sigma,i}$ is constructed and added to $\mathcal{T}$. Assume that for $j = 1,\ldots,i-1$, ground terms $s_{\sigma,j}$ already have been constructed by the algorithm. Then define

$$\psi_{\sigma,i} = Y.\pi_{\sigma(1)} \neq s_0.\pi_{\sigma(1)} \wedge Y.\pi_{\sigma(2)} \neq s_{\sigma,1}.\pi_{\sigma(2)} \wedge \ldots \wedge Y.\pi_{\sigma(i)} \neq s_{\sigma,i-1}.\pi_{\sigma(i)},$$

and let $\mathcal{A}_{\sigma,i}$ denote the set of clauses which is obtained from $\mathcal{A}$ by replacing each clause of the form $p(t_0) \Leftarrow \alpha, \phi'$ with $p(t_0) \Leftarrow \alpha, \psi_{\sigma,i}[t_0/Y] \wedge \phi'$. Thus,

$$[\![p]\!]_{\mathcal{A}_{\sigma,i}} = \{t \in [\![p]\!]_{\mathcal{A}} \mid t.\pi_{\sigma(1)} \neq s_0.\pi_{\sigma(1)} \wedge\ t.\pi_{\sigma(2)} \neq s_{\sigma,1}.\pi_{\sigma(2)} \wedge\ \ldots \wedge t.\pi_{\sigma(i)} \neq s_{\sigma,i-1}.\pi_{\sigma(i)}\}.$$

If $[\![p]\!]_{\mathcal{A}_{\sigma,i}}$ is empty, then the algorithm proceeds with the next permutation – or terminates, if all permutations have already been considered. Otherwise, a ground term $s_{\sigma,i} \in [\![p]\!]_{\mathcal{A}_{\sigma,i}}$ is selected and added to the set $\mathcal{T}$. Assume that after termination of the algorithm,

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

$\mathcal{T} = \{t_1, \ldots, t_r\}$. Let $\bar{\psi} = \psi[t_1/Y] \vee \ldots \vee \psi[t_r/Y]$. Now consider an arbitrary ground substitution θ defined for all variables different from Y occurring in ψ, and consider a ground term $t \in [\![p]\!]_{\mathcal{A}}$ so that $\theta \models \psi[t/Y]$. Let again $s_0 \in [\![p]\!]_{\mathcal{A}}$ denote the start term of the algorithm. If $\theta \models \psi[s_0/Y]$, then we have found a term t' in $\mathcal{T}$ such that $\theta \models \psi[t'/Y]$. Now assume that $\theta \not\models \psi[s_0/Y]$. Then we claim that there is a permutation σ of $\{1, \ldots, l\}$ and some index $1 \leqslant i \leqslant l$ such that the algorithm has constructed a ground term $s_{\sigma,i}$ with $\theta \models \psi[s_{\sigma,i}/Y]$ – again implying that there is a term t' in $\mathcal{T}$ such that $\theta \models \psi[t'/Y]$.

We construct the permutation σ and the index i as follows. Since $\theta \not\models \psi[s_0/Y]$, there is a disequality $Y.\pi_{k_1} \neq u_{k_1}$ in ψ where $\theta \not\models s_0.\pi_{k_1} \neq u_{k_1}$. This means that both $s_0.\pi_{k_1}$ and $u_{k_1}\theta$ are defined, and $s_0.\pi_{k_1} = u_{k_1}\theta$. Accordingly, we choose $\sigma(1)$ as k_1. In particular, this means that all further terms $s_{\sigma,i}$ output by the algorithm, $\theta \models s_{\sigma,i}.\pi_{\sigma(1)} \neq u_{\sigma(1)}$ holds. After each round i, one further entry k_i is selected such that for every permutation σ with $\sigma(j) = k_j$ for $j = 1, \ldots, i$, the corresponding disequality is guaranteed to be satisfied by θ after substitution of Y with $s_{\sigma,i}$. Assume that in this way, $k_1, \ldots, k_{i-1}$ and ground terms $s_{\sigma,1}, \ldots, s_{\sigma,i-1}$ have been constructed. Assume further that $\theta \not\models \psi[s_{\sigma,i-1}/Y]$ (because otherwise, we are done). This means that there is some index $1 \leqslant k_i \leqslant l$ so that $\theta \not\models s_{\sigma,i-1}.\pi_{k_i} \neq (u_{k_i}\theta)$. By construction, $t \in [\![p]\!]_{\mathcal{A}_{\sigma,i}}$ for each permutation σ with $\sigma(j) = k_j, j = 1, \ldots, i$. Therefore, for each such σ, $[\![p]\!]_{\mathcal{A}_{\sigma,i}}$ is non-empty, and the algorithm will construct another term $s_{\sigma,i}$. If i reaches l, then each disequality $Y.\pi_j \neq u_j$ has been considered. This means that $\theta \models \psi[s_{\sigma,l}/Y]$ for the permutation σ defined by $\sigma(j) = k_j, j = 1, \ldots, l$, and we are done. □

Although the algorithm in principle may construct an exponential number of terms, this need not always be the case.

Example 5.3. For $n \geqslant 1$, consider the conjunction of path disequalities

$$\phi \quad = \quad (X_2 \neq X_1) \wedge (X_2.(g,1) \neq X_1) \wedge \ldots \wedge (X_2.\underbrace{(g,1)\ldots(g,1)}_{n-1 \text{ times}} \neq X_1)$$

and let $\mathcal{A}$ denote the set generalized automaton clauses consisting of

$$\begin{array}{rcl} p'(a) & \Leftarrow & \\ p'(g(X_1)) & \Leftarrow & p'(X_1) \\ p(X_1) & \Leftarrow & p'(X_1) \end{array}$$

We remark that the predicate p only occurs in the heads of clauses. In order to construct for predicate p, constraint ϕ and variable X_2 the set $\mathcal{T}$, we start with some $s_0 \in [\![p]\!]_{\mathcal{A}}$, e.g., $s_0 = a$. There are n occurrences of path expressions involving X_2. In principle, the algorithm iterates over all permutations of n elements. Let σ denote the permutation which considers the occurrences of X_2 in ϕ from left-to-right. Then the algorithm constructs $\psi_{\sigma,1}$ as $X_2 \neq a$ that excludes the term a from $[\![p]\!]_{\mathcal{A}}$. Thus, the modified clause for p in $\mathcal{A}_{\sigma,1}$ is given by

$$p(X_1) \quad \Leftarrow \quad p'(X_1), X_1 \neq a.$$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Since $[\![p]\!]_{\mathcal{A}_{\sigma,1}}$ is non-empty, we find another term $g(a)$. Now we consider the second disequality $X_2.(g,1) \neq X_1$. Accordingly,

$$\begin{aligned} \psi_{\sigma,2} &= (X_2 \neq a) \wedge (X_2.(g,1) \neq g(a).(g,1)), \\ &= (X_2 \neq a) \wedge (X_2.(g,1) \neq a), \end{aligned}$$

where the clause for p is given by

$$p(X_1) \Leftarrow p'(X_1), (X_1 \neq a) \wedge (X_1.(g,1) \neq a).$$

Again, $[\![p]\!]_{\mathcal{A}_{\sigma,2}}$ is non-empty. Repeating this procedure for $i = 3, \ldots, n$, may altogether result in a set $\mathcal{T} = \{g^i(a) \mid 0 \leqslant i \leqslant n\}$. We may convince ourselves that the terms in this set suffice also for other permutations $\sigma' \neq \sigma$.

Lemma 5.1 can be applied to remove auxiliary variables from generalized automaton clauses.

Lemma 5.2. Let $\mathcal{A}$ denote a finite set of generalized automaton clauses with general term disequalities. Then a finite set $\mathcal{A}'$ of generalized automaton clauses with general term disequalities can be effectively constructed which does not use auxiliary variables, such that for each predicate p, $[\![p]\!]_{\mathcal{A}} = [\![p]\!]_{\mathcal{A}'}$.

Proof. W.l.o.g., we assume that for every variable X_j occurring in a constraint of a clause c in $\mathcal{A}$, there is also a literal $p(X_j)$ in the body of c for some predicate p. In order to construct the set $\mathcal{A}'$, we apply a saturation procedure that successively splits auxiliary variables of clauses inside the set $\mathcal{A}$ by means of the construction from Lemma 5.1, until all clauses with auxiliary variables in $\mathcal{A}$ are abstractly subsumed by the subset of clauses without auxiliary variables. We exhaustively add implied clauses according to the following rule. Assume that $\mathcal{C}_1$ is the current set of generalized automaton clauses and $\mathcal{A}_1 \subseteq \mathcal{C}_1$ is the current subset of clauses in $\mathcal{C}_1$ with *no* auxiliary variables. Assume that $\mathcal{C}_1$ has a clause

$$q(t_0) \Leftarrow \alpha, p(X_j), \beta, \phi,$$

where X_j does not occur in t_0. By Lemma 2.5 (using the identity homomorphism), ϕ is equivalent to a positive Boolean combination of path disequalities only. Let ϕ be equivalent to a disjunction $\psi_1 \vee \ldots \vee \psi_r$ of conjunctions $\psi_k (1 \leqslant k \leqslant r)$ of path disequalities only. Let $t_1, \ldots, t_l$ denote a sequence of ground terms in $[\![p]\!]_{\mathcal{A}_1}$ such that for every $k = 1, \ldots, r$, there is a subsequence $t_{k_1}, \ldots, t_{k_{l_k}}$ with $\psi'_k = \psi_k[t_{k_1}/X_j] \vee \ldots \vee \psi_k[t_{k_{l_k}}/X_j]$, so that it holds for every ground substitution θ of variables different from X_j that $\theta \oplus \{X_j \mapsto t\} \models \psi_k$ for some $t \in [\![p]\!]_{\mathcal{A}_1}$ iff $\theta \models \psi'_k$. Such a sequence of ground terms can be constructed by Lemma 5.1. Then the clause

$$q(t_0) \Leftarrow \alpha, \beta, \phi'$$

is added to the set $\mathcal{C}_1$, where $\phi' = \phi[t_1/X_j] \vee \ldots \vee \phi[t_l/X_j]$.

We remark that the number of cores of clauses added by the saturation procedure remains finite. Moreover, also the number of path expressions occurring in constraints is bounded. Therefore, by Corollary 4.1, the saturation procedure terminates. Let $\bar{\mathcal{A}}$ denote the saturated set obtained from $\mathcal{A}$, and let $\mathcal{A}'$ denote the subset of clauses in $\bar{\mathcal{A}}$ without auxiliary variables. We claim that $[\![p]\!]_{\mathcal{A}'} = [\![p]\!]_{\bar{\mathcal{A}}} = [\![p]\!]_{\mathcal{A}}$ for all predicates p.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

First, one verifies for every ground term $t \in [\![p]\!]_{\bar{\mathcal{A}}}$ that for every resolution tree for $p(t)$ with clauses from $\bar{\mathcal{A}}$, a resolution tree for $p(t)$ with clauses from $\mathcal{A}'$ can be constructed. This proof is by structural induction on resolution trees. As $\mathcal{A}' \subseteq \bar{\mathcal{A}}$ and therefore $[\![p]\!]_{\mathcal{A}'} \subseteq [\![p]\!]_{\bar{\mathcal{A}}}$, the first equality of the claim follows. For the second equality, one proves that for every finite set $\mathcal{C}_1$ of generalized automaton clauses, and every clause c' possibly added to $\mathcal{C}_1$ by the splitting rule, the least models $\mathcal{M}_{\mathcal{C}_1}$ and $\mathcal{M}_{\mathcal{C}_1 \cup \{c'\}}$ coincide. For that, we assume for a contradiction, that there is a ground fact $q(t)$ in $\mathcal{M}_{\mathcal{C}_1 \cup \{c'\}} \setminus \mathcal{M}_{\mathcal{C}_1}$ so that there is a resolution tree for $q(t)$ with a minimal number of nodes. As this resolution tree is minimal, the clause c' occurs in the root only. Assume that the clause c' is of the form:

$$q(t_0) \Leftarrow \alpha, \beta, \phi'$$

with $\phi' = \phi[t_1/X_j] \vee \ldots \vee \phi[t_r/X_j]$ for some clause $c \in \mathcal{C}_1$ of the form $q(t_0) \Leftarrow \alpha, p(X_j), \beta, \phi$. In particular, the ground terms $t_1, \ldots, t_r$ all are contained in $[\![p]\!]_{\mathcal{A}_1}$ if $\mathcal{A}_1$ is the subset of generalized automaton clauses of $\mathcal{C}_1$ without auxiliary variables. Assume that the root node of the resolution tree for $q(t)$ is $\langle c', \theta \rangle$. As $\theta \models \phi'$, we have that $\theta \models \phi[t_i/X_j]$ for some i. As $t_i \in [\![p]\!]_{\mathcal{A}_1} \subseteq [\![p]\!]_{\mathcal{C}_1}$, there is a resolution tree for $p(t_i)$ using clauses from $\mathcal{C}_1$ only. This resolution tree can be used to construct a resolution tree for $q(t)$ whose root node is given by $\langle c, \theta \oplus \{X_j \mapsto t_i\} \rangle$ and which uses clauses from $\mathcal{C}_1$ only. Therefore, $q(t) \in \mathcal{M}_{\mathcal{C}_1}$ – in contradiction to our assumption. □

As for every finite set $\mathcal{A}$ of generalized automaton clauses with general term disequalities and every predicate p, an equivalent finite set without auxiliary variables can be constructed by Lemma 5.2, and this set, in turn, is effectively equivalent to a Godoy automaton, we conclude that it can be decided whether $[\![p]\!]_{\mathcal{A}} = \varnothing$. Recalling Lemma 4.3, we have in summary:

Theorem 5.2. For a predicate q, it is decidable whether $[\![q]\!]_{\mathcal{A}} \neq \varnothing$, whenever $\mathcal{A}$ is a finite set of

1. generalized automaton clauses with general term disequalities; or
2. automaton clauses with general hom-term disequalities.

5.2. *Elimination of auxiliary variables*

In the following, let H denote a fixed tree homomorphism. Let $\mathcal{C}'$ denote the current set of clauses attained during the normalization of a finite set of $\mathcal{H}_1$-clauses with general hom-term disequalities, where $\mathcal{N}'$ is the subset of normal clauses in $\mathcal{C}'$. Consider a simple clause c of the form $h \Leftarrow \alpha, \psi$ in $\mathcal{C}'$ where $\alpha = \alpha', q_1(X_j), \ldots, q_r(X_j)$ for a variable X_j neither occurring in α' nor in h. Recall that, in order to apply splitting, we require a finite sequence $t_1, \ldots, t_l$ of ground terms such that for all ground substitutions θ of the variables in c different from X_j, $\theta \models \phi[t/X_j]$ for some $t \in [\![q_1]\!]_{\mathcal{N}'} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}'}$ iff $\theta \models \phi[t_i/X_j]$ for some $1 \leqslant i \leqslant l$. The next lemma shows that such a sequence $t_1, \ldots, t_l$ can indeed be constructed by means of the algorithm from Lemma 5.1.

Lemma 5.3. Let $\mathcal{N}$ denote a finite set of normal clauses with general hom-term disequalities, $q_1, \ldots, q_r$ a sequence of predicates, ϕ a positive Boolean combination of general

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

hom-term disequalities, and X_j a variable. Then a finite set $\mathcal{T}$ of ground terms can be constructed so that for every ground substitution θ of the variables in ϕ excluding X_j, $\theta \models \phi[t/X_j]$ for some $t \in [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$ iff $\theta \models \phi[t'/X_j]$ for some $t' \in \mathcal{T}$.

Proof. By Lemma 2.8, a finite set $\mathcal{A}$ of automaton clauses can be constructed using the same kind of constraints as $\mathcal{N}$ together with a predicate $[\{q_1, \ldots, q_r\}]$ so that $[\![[\{q_1, \ldots, q_r\}]]\!]_{\mathcal{A}} = [\![q_1]\!]_{\mathcal{N}} \cap \ldots \cap [\![q_r]\!]_{\mathcal{N}}$. By Lemma 4.3, a finite set $\mathcal{A}_{\mathrm{H}}$ of generalized automaton clauses with general term disequalities only, can be constructed so that $[\![[\{q_1, \ldots, q_r\}]\!]_{\mathcal{A}_{\mathrm{H}}} = \mathrm{H}([\![[\{q_1, \ldots, q_r\}]]\!]_{\mathcal{A}})$. By Lemma 5.2, we may w.l.o.g., assume that the clauses in $\mathcal{A}_{\mathrm{H}}$ do not have auxiliary variables. Let ϕ' denote the positive Boolean combination of general term disequalities which is obtained from ϕ by replacing every hom-path $[X_i]_{\mathrm{H}}.\pi$ with $X_i.\pi$. By Lemma 2.6, ϕ' can be equivalently written as $\phi' = \psi_1 \vee \ldots \vee \psi_l$ for conjunctions $\psi_1, \ldots, \psi_l$ of path disequalities. For each $i = 1, \ldots, l$, Lemma 5.1, applied to $\mathcal{A}_{\mathrm{H}}$, the predicate $[\{q_1, \ldots, q_r\}]$, the conjunction ψ_i and the variable X_j, provides us with a finite set $\mathcal{T}_i$ so that for all ground substitutions θ of variables different from X_j, $\theta \models \psi_i[t/X_j]$ for some $t \in [\![[\{q_1, \ldots, q_r\}]]\!]_{\mathcal{A}_{\mathrm{H}}}$ iff $\theta \models \psi_i[t'/X_j]$ for some $t' \in \mathcal{T}_i$. Thus, we set $\mathcal{T} = \mathcal{T}_1 \cup \ldots \cup \mathcal{T}_l$. Since $\theta \models \psi_i[t'/X_j]$ for some i and some $t' \in \mathcal{T}_i$ implies that $\theta \models \phi'[t'/X_j]$ for some $t' \in \mathcal{T}$, the assertion follows. □

In summary, we thus have shown that finite sets of $\mathcal{H}_1$-clauses with general hom-term disequalities can be effectively normalized to an equivalent set of normal clauses. For these in turn, a corresponding set of automaton clauses can be constructed, still using general hom-term disequalities. By Theorem 5.2, we therefore obtain the following:

Theorem 5.3. For a finite set $\mathcal{C}$ of $\mathcal{H}_1$-clauses with general hom-term disequalities and a predicate p, it is decidable whether or not $[\![p]\!]_{\mathcal{C}} \neq \varnothing$ holds.

6. Conclusion

We have shown that finite sets of Horn clauses of the class $\mathcal{H}_1$ extended with disequalities of hom-paths can be normalized into an equivalent finite set of automaton clauses with disequalities. As these sets of automaton clauses correspond to known classes of tree automata for which emptiness is decidable, we obtained new decision results for constrained Horn clauses. We also compared the expressiveness of finite sets of automaton clauses with various kinds of disequality constraints. Using general hom-term disequalities turned out to be more expressive than either using path disequalities or hom-disequalities as considered in Seidl and Reuß (2012) and Reuß and Seidl (2012), respectively.

Hom-path disequalities allow us to compare, to a certain extent, subtrees for semantic properties. It remains to explore in how far this ability allows to construct enhanced analyses of term-manipulating programs. In the present paper, we were concerned with decidability only. It would be interesting to determine the precise complexities of the suggested algorithms and, perhaps, find more efficient procedures, in particular for splitting in presence of path or hom-path disequalities.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Appendix A. Proofs

Proof of Lemma 3.4. Assume for a proof by contradiction that there is a finite set $\mathcal{A}$ of automaton clauses with hom-disequalities which accepts T through some predicate p.

Concerning the alphabet Σ of occurring symbols in $\mathcal{A}$, we may assume w.l.o.g. that $\Sigma = \{a, f\}$ for the symbols a, f used to define T. In the following, we show that this is no restriction on expressiveness, First, let $\mathcal{A}$ consist of clauses of the forms $q(a) \Leftarrow$ and $q(f(X_1, X_2)) \Leftarrow q_1(X_1), q_2(X_2), \phi$ only. This is w.l.o.g. since clauses with symbols other than a, f in the heads cannot occur in resolution trees for elements of T, and preconditions ϕ of clauses of the form $q(a) \Leftarrow \phi$ are either vacuously true or vacuously false as they do not contain any variables. We observe that every ground substitution θ that is applied to the clauses of $\mathcal{A}$ maps the variables X_i to ground terms $t \in \mathcal{T}_{\{a,f\}}$. Concerning application of H in the constraints, we have $([X_i]_\mathrm{H})\theta = \mathrm{H}(X_i\theta)$, i.e., H is applied only to terms over $\{a, f\}$. Now, let $\Sigma = \{a, f\}$, and assume that H is defined for the symbols of some alphabet $\Sigma' = \Sigma \dot\cup \{s_1, \ldots, s_n\}$, containing $n \geqslant 0$ symbols s_i different from a, f. We define a coding $c : \mathcal{T}_{\Sigma'}(\{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}\}) \to \mathcal{T}_\Sigma(\{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}\})$ such that for every pair of two hom-terms $t, t' \in \mathcal{T}_{\Sigma'}(\{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}\})$ and ground substitution θ for X_1, X_2, $\theta \models t = t'$ w.r.t. H if and only if $\theta \models c(t) = c(t')$ w.r.t. H′ where the tree homomorphism H′ is defined as $c \circ \mathrm{H}$. Let $t'_0, \ldots, t'_n$ be $n+1$ distinct elements of $\mathcal{T}_\Sigma$. For a sequence of $k \geqslant 0$ terms $r_1, \ldots, r_k$, we define $d(r_1, \ldots, r_k)$ recursively as $d() = a$ and $d(r_1, \ldots, r_k) = f(r_1, d(r_2, \ldots, r_k))$. We define $c(a) = a$ and $c([X_j]_\mathrm{H}) = [X_j]_\mathrm{H}$ for $j \in \{1, 2\}$. In order to simplify the notation, let s_0 denote the symbol f. Then for $s_i \in \Sigma' \setminus \{a\}$ of arity k and k terms $t_1, \ldots, t_k \in \mathcal{T}_{\Sigma'}(\{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}\})$, $c(s_i(t_1, \ldots, t_k))$ is recursively defined as $f(t'_i, d(c(t_1), \ldots, c(t_k)))$.

We prove by contradiction that c^{-1} is a function. For $t \in \mathcal{T}_{\Sigma'}(\{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}\})$ and $t' \in \{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}, a\}$, we have $c(t) = t'$ iff $t = t'$. Thus, $c^{-1}(t) = t'$ iff $t = t'$. Now, let s'_1 be a term of smallest depth such that there exists a term $t \in \mathcal{T}_\Sigma(\{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}\}) \setminus \{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}, a\}$ such that there exists a term $s'_2 \neq s'_1$ with $s'_1, s'_2 \in \{s \mid c(s) = t\}$. Let $s'_1 = s_i(t_1, \ldots, t_k)$ ($0 \leqslant i \leqslant n$ – again with the understanding that f is represented by s_0). Then, $t = f(t'_i, d(c(t_1), \ldots, c(t_k)))$. Since $t.(f, 1) = t'_i$ is a unique result of applying c to a term whose topmost symbol is s_i, it follows that also s'_2 has the form $s_i(t''_1, \ldots, t''_m)$, where $m = k$ is the arity of s_i. Thus, $t = f(t'_i, d(c(t''_1), \ldots, c(t''_k)))$. For $k \geqslant 0$, it can be proved by induction on the number of arguments that $d(a_1, \ldots, a_k) = d(b_1, \ldots, b_k)$ iff $a_i = b_i$ for all $i \in \{1, \ldots, k\}$. For all $j \in \{1, \ldots, k\}$, we therefore have $c(t_j) = c(t''_j)$. Let $S_j = \{s \mid c(s) = c(t_j)\}$. Then for all $1 \leqslant j \leqslant k$, we have $t_j, t''_j \in S_j$. Since s'_1 is of smallest depth, it follows that $|S_j| = 1$ for all those sets S_j. We conclude that $t_j = t''_j$ for all $1 \leqslant j \leqslant k$ and therefore $s'_1 = s'_2$ – contradiction.

Since c defines a bijection from $\mathcal{T}_{\Sigma'}$ to the set $\{t \mid \exists s \in \mathcal{T}_{\Sigma'} \text{ s.t. } c(s) = t\}$, we have for $l, r \in \mathcal{T}_{\Sigma'}$ that $l = r$ iff $c(l) = c(r)$. Accordingly, for $\mathrm{H}' = c \circ \mathrm{H}$, we have $\mathrm{H}'l = \mathrm{H}'r$ iff $\mathrm{H}l = \mathrm{H}r$. Let $\theta = \{X_1 \mapsto t_1, X_2 \mapsto t_2\}$ be a ground substitution for the variables X_1 and X_2. Then $\theta \models l = r$ iff $l[\mathrm{H}t_1/[X_1]_\mathrm{H}, \mathrm{H}t_2/[X_2]_\mathrm{H}] = r[\mathrm{H}t_1/[X_1]_\mathrm{H}, \mathrm{H}t_2/[X_2]_\mathrm{H}]$. Let $t \in \{l, r\}$. By c's recursive definition, $c(t)[\mathrm{H}'t_1/[X_1]_\mathrm{H}, \mathrm{H}'t_2/[X_2]_\mathrm{H}] = c(t)[c(\mathrm{H}(t_1))/[X_1]_\mathrm{H}, c(\mathrm{H}(t_2))/[X_2]_\mathrm{H}] = c(t[\mathrm{H}t_1/[X_1]_\mathrm{H}, \mathrm{H}t_2/[X_2]_\mathrm{H}])$. We conclude that $\theta \models l = r$ w.r.t. H iff $\theta \models c(l) = c(r)$ w.r.t. H′. Recall that for hom-disequalities $l \neq r$ (and ground substitutions θ), by definition, $\theta \models l \neq r$ iff $\neg(\theta \models l = r)$. Consider now the set $\mathcal{A}'$ which contains exactly the clauses

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

from $\mathcal{A}$ but with every hom-disequality $l \neq r$ replaced by $c(l) \neq c(r)$. Then, $\mathcal{A}'$ w.r.t. H′ contains only symbols from $\Sigma = \{a, f\}$ and is equivalent to $\mathcal{A}$ w.r.t. H.

Recalling Lemma 2.6, we may w.l.o.g., assume that $\mathcal{A}$ has conjunctive constraints only. By the construction provided in Seidl and Neumann (1999), Godoy *et al.* (2010), and Seidl and Reuß (2012), a finite set $\mathcal{B}$ of automaton clauses can be obtained from $\mathcal{A}$ with complementary constraints, i.e., conjunctions of hom-*equalities* only, such that $\mathcal{B}$ defines a predicate $\overline{p}$ which accepts the complement of T, i.e.,

$$\begin{aligned}[\![\overline{p}]\!]_{\mathcal{B}} &= \overline{T} \\ &= \mathcal{T}_\Sigma \setminus T \\ &= \{a\} \cup \{f(t, f(t,s)) \mid s,t \in \mathcal{T}_\Sigma\}.\end{aligned}$$

The set $\mathcal{B}$ is obtained in two steps. First, a finite set $\mathcal{N}$ of normal clauses with hom-equality constraints is constructed such that for every predicate q of $\mathcal{A}$, $\mathcal{N}$ has a predicate $\overline{q}$ with $[\![\overline{q}]\!]_{\mathcal{N}} = \mathcal{T}_\Sigma \setminus [\![q]\!]_{\mathcal{A}}$. Assume that $q(f(X_1,\ldots,X_k)) \Leftarrow l_{i1},\ldots,l_{ir_i}$ for $i = 1,\ldots,m$ are the clauses for q and constructor f where each l_{ij} either is a literal of the form $p'(X_s)$ or a single disequality. The predicate $\overline{q}$ for constructor f then is defined by the set of all clauses

$$\overline{q}(f(X_1,\ldots,X_k)) \Leftarrow \bigwedge_{1 \leqslant i \leqslant m} \bar{l}_{ij_i},$$

where for each i, $1 \leqslant j_i \leqslant r_i$. For a literal l_{ij_i} of the form $p'(X_s)$, $\bar{l}_{ij_i}$ is given by $\overline{p}'(X_s)$, and if l_{ij_i} equals a disequality $t_1 \neq t_2$, $\bar{l}_{ij_i}$ is given by $t_1 = t_2$. By this construction, the resulting clauses are normal clauses where the constraints are conjunctions of hom-equalities. As in Lemma 2.8, a finite set $\mathcal{B}$ of automaton clauses with hom-equalities can be computed such that for all predicates $\overline{q}$ of $\mathcal{N}$, $[\![\overline{q}]\!]_{\mathcal{N}} = [\![\overline{q}]\!]_{\mathcal{B}}$.
W.l.o.g., let us assume that each single equality occurring in $\mathcal{B}$ is of the form $[X_i]_{\mathrm{H}} = r$ for some hom-term r. Let us furthermore assume w.l.o.g. that $[X_i]_{\mathrm{H}}$ does not occur in r for otherwise, the equality would be either vacuously true (if r is $[X_i]_{\mathrm{H}}$) or vacuously false (if r is different from $[X_i]_{\mathrm{H}}$).

We now make a case distinction on the effects of H, consisting of four cases. The goal for the first three cases is to show the existence of an equivalent finite set of automaton clauses with term equalities which is then proved wrong. In the fourth case, we directly deduce a contradiction.

Case 1: Both $[X_1]_{\mathrm{H}}$ and $[X_2]_{\mathrm{H}}$ occur in $\mathrm{H}f$. Then, H maps each term of the form $f(t_1,t_2)$ again to a term of the form $f(s_1,s_2)$ whose size is equal or bigger.

We observe that H defines the following bijection between ground terms $\mathcal{T}_\Sigma$ and their homomorphic images $\mathrm{H}(\mathcal{T}_\Sigma)$. Since we only have the constructors f and a, ground terms can be uniquely represented by the set of unlabelled paths to the ground term a that are defined for them. Let Π_1, Π_2 and Π_a denote the set of unlabelled paths in $\mathrm{H}f$ to $[X_1]_{\mathrm{H}}, [X_2]_{\mathrm{H}}$, and the ground term a, respectively, and let Π'_a denote the set of unlabelled paths in $\mathrm{H}a$ to a. For each maximal (i.e., reaching a) path $i_1.\ldots.i_n$ of a ground term t, exactly the following maximal paths exist in $\mathrm{H}t$:

$$\begin{array}{lll} (1) & \pi_1.\ldots.\pi_n.\pi, & \pi_j \in \Pi_{i_j}, \pi \in \Pi'_a, \\ (2) & \pi_1.\ldots.\pi_k.\pi', & \pi_j \in \Pi_{i_j}, \pi' \in \Pi_a, 0 \leqslant k < n \qquad (\text{if } n > 0). \end{array}$$

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

We formally prove our observation by structural induction. For the base case $t = a$, ϵ is the only path to a. Since $n = 0$, only paths of form (1) need to be considered. In $\mathrm{H}t = \mathrm{H}a$, we have exactly the paths of Π'_a – by definition of Π'_a. For the induction step, we consider $\mathrm{H}t$ for a ground term $t = f(t_1, t_2)$. In $\mathrm{H}t$, all maximal paths π have a prefix π' which is a maximal path in $\mathrm{H}f$. We distinguish two cases.

First, if $(\mathrm{H}f).\pi' = a$, then we also have $(\mathrm{H}t).\pi' = a$ and, by definition, $\pi' \in \Pi_a$. Furthermore, $\pi' = \pi$ because every path $\pi'.\pi_r$ with $\pi_r \neq \epsilon$ is undefined for $\mathrm{H}t$. This first case covers exactly the paths of form (2) for $k = 0$. For $k > 0$ and the paths of form (1), the second case applies:

Second, let $(\mathrm{H}f).\pi' = [X_i]_\mathrm{H}$ for $i \in \{1, 2\}$. Then, π is of the form $\pi_i.\pi''$ where $\pi_i \in \Pi_i$ and π'' is a maximal path in $\mathrm{H}t_i$. Now consider a maximal path $i_0.i_1.\ldots.i_n$ in t with $i_0 = i$. The maximal paths of $\mathrm{H}t_i$ according to the induction hypothesis are given by (1) and (2). Concerning (1), if we shift the indexes 0 to n of the path $i_0.i_1.\ldots.i_n$ under consideration by 1, we end up with indexes 1 to $n + 1$ and the same conditions for all π_j. Concerning (2), the same index shift yields the conditions $1 \leqslant k < n + 1$. The paths of (2) with $k = 0$ were already collected by the first case in our case distinction. Altogether, we therefore get exactly the paths of (1) and (2) for $n + 1$ instead of n.

Consequently, two ground terms are equal if and only if their homomorphic images are equal. Furthermore, there is the following recursive membershiptest for $\mathrm{H}(\mathcal{T}_\Sigma)$. Let t be a ground term. If $t \neq \mathrm{H}a$, then $t \in \mathrm{H}(\mathcal{T}_\Sigma)$ if and only if

(1) $t.\pi = a$ for all $\pi \in \Pi_a$,
(2) $t.\pi \in \mathrm{H}(\mathcal{T}_\Sigma)$ for all $\pi \in \Pi_1 \cup \Pi_2$ and
(3) $t.\pi = t.\pi'$ for all $\pi, \pi' \in \Pi_1$ as well as for all $\pi, \pi' \in \Pi_2$.

In the following, we describe how to replace hom-equalities $[X_i]_\mathrm{H} = r$ by equalities of ordinary terms. For notational convenience, let us assume that $i = 1$, i.e., X_i is X_1. The construction for the case $i = 2$ is analogous. Recall that we furthermore assume that $[X_1]_\mathrm{H}$ does not occur in r.

Let $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2$ be two paths in $\mathrm{H}f$ to $[X_1]_\mathrm{H}$ and $[X_2]_\mathrm{H}$, respectively. We define the mapping $\mathrm{pre}_\mathrm{H} : \mathcal{T}_\Sigma(\{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}\}) \mapsto \mathcal{T}_\Sigma(\{X_1, X_2\})$ by

$$\mathrm{pre}_\mathrm{H}(t) = \begin{cases} X_i & \text{if } t = [X_i]_\mathrm{H} \\ a & \text{if } t = \mathrm{H}a \\ f(\mathrm{pre}_\mathrm{H}(t.\pi_1), \mathrm{pre}_\mathrm{H}(t.\pi_2)) & \text{if } t.\pi \text{ is defined for } \textbf{all } \pi \in \Pi_1 \cup \Pi_2 \cup \Pi_a \\ & \quad \text{and } t.\pi = a \text{ for all } \pi \in \Pi_a \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Note that when applied to a term $t \in \mathrm{H}(\mathcal{T}_\Sigma)$, $\mathrm{pre}_\mathrm{H}(t)$ is independent of the choice of π_1 and π_2 because for $i \in \{1, 2\}$, $t.\pi' = t.\pi''$ holds for all $\pi', \pi'' \in \Pi_i$. Hence, pre_H maps homomorphic images to their respective pre-images.

Our goal is to use ordinary term equalities of the form $X_i = \mathrm{pre}_\mathrm{H}(r)$ in order to simulate hom-equalities $[X_i]_\mathrm{H} = r$. An important step in this direction is to establish that for all ground substitutions $\theta = \{X_1 \mapsto t_1, X_2 \mapsto t_2\}$ with $t_1, t_2 \neq a$ and every hom-equality $l = r$ with $l\theta, r\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$, $\theta \models l = r$ if and only if $\theta \models \mathrm{pre}_\mathrm{H}(l) = \mathrm{pre}_\mathrm{H}(r)$. For

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

that, we first prove a property of *complex* homomorphic images, i.e., elements of $\mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$.

Let $s \neq \mathrm{H}a$ be a term of $\mathrm{H}(\mathcal{T}_\Sigma)$, i.e., $s = \mathrm{H}(f(t_1, t_2))$ for some $t_1, t_2 \in \mathcal{T}_\Sigma$. Then for all $\pi \neq \epsilon$ for which there is a path $\pi' \neq \epsilon$ with $\pi.\pi' \in \Pi_1 \cup \Pi_2 \cup \Pi_a$, it holds that either $s.\pi = \mathrm{H}a$ or $s.\pi \notin \mathrm{H}(\mathcal{T}_\Sigma)$. Intuitively, if a complex homomorphic image s properly contains another complex homomorphic image s' at some path π (i.e., $s.\pi = s'$), then π has a prefix π' with $(\mathrm{H}f).\pi' = [X_i]_\mathrm{H}$, $i \in \{1, 2\}$.

This assertion is proved by induction on the depth of the term s. Let us therefore assume by induction hypothesis that the assertion holds for every proper subterm s' of $s \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$. In the remainder of the proof, we deduce a contradiction from the assumption that the assertion does not also hold for s. Let π be an arbitrary path violating the assertion, i.e., $\pi \neq \epsilon$, there is some $\pi' \neq \epsilon$ with $\pi.\pi' \in \Pi_1 \cup \Pi_2 \cup \Pi_a$, and $s.\pi \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$.

By induction hypothesis, every non-empty path in $s.\pi$ to a complex homomorphic image must have a prefix from the set $\Pi_1 \cup \Pi_2$. Let $\Pi_f = \{\pi \mid (\mathrm{H}f).\pi = \mathrm{H}a\}$. Since $s.\pi \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$, we have $s.\pi.\pi_s \in \mathrm{H}(\mathcal{T}_\Sigma)$ for every path $\pi_s \in \Pi_1 \cup \Pi_2$. Thus, in $s.\pi$, the set of non-empty paths to elements of $\mathrm{H}(\mathcal{T}_\Sigma)$ of minimal lengths (i.e., there is no proper prefix with the same property) are given exactly by $\Pi_f \cup \Pi_1 \cup \Pi_2$. Let $\Pi = \Pi_f \cup \Pi_1 \cup \Pi_2$. Let $|\pi|$ denote the length of a path π as inductively defined by $|\epsilon| = 0$ and $i.\pi_r = 1 + |\pi_r|$ for $i \in \mathbb{N}$. Let $\pi_{\max}$ be a longest path of Π. Now consider the path $\pi.\pi_{\max}$. Recall that π is a proper prefix of the path $\pi.\pi' \in \Pi$. We have $s.\pi.\pi_{\max} \in \mathrm{H}(\mathcal{T}_\Sigma)$ and therefore $s.\pi_p \neq \mathrm{H}a$ for every proper prefix π_p of $\pi.\pi_{\max}$. Recall that $\pi_{\max}$ is a longest path of the set $\Pi \supseteq (\Pi_1 \cup \Pi_2)$. Therefore, $|\pi_{\max}| \geqslant |\pi''|$ for all $\pi'' \in \Pi_1 \cup \Pi_2$. We furthermore have $s.\pi'' \in \mathrm{H}(\mathcal{T}_\Sigma)$ for all $\pi'' \in \Pi_1 \cup \Pi_2$. Since $|\pi.\pi_{\max}| > |\pi_{\max}|$, there must be a proper prefix $\pi_p = \pi.\pi_u \in \Pi$ of $\pi.\pi_{\max}$ with $s.\pi_p \in \mathrm{H}(\mathcal{T}_\Sigma)$ and $\pi_u \neq \epsilon$ (because π is a proper prefix of some element of Π which implies that no path of Π can be a prefix of π). However, π_u is a proper prefix of $\pi_{\max} \in \Pi$, and $(s.\pi).\pi_u \in \mathrm{H}(\mathcal{T}_\Sigma)$. Therefore, $\pi_{\max}$ is not of minimal length, implying that $\pi_{\max} \notin \Pi$ – contradiction.

Now consider a hom-equality $l = r$. Let θ be a ground substitution with $X_i\theta \neq a$ for all $i \in \{1, 2\}$ (i.e., $([X_i]_\mathrm{H})\theta \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$), and let $\{l\theta, r\theta\} \subseteq \mathrm{H}(\mathcal{T}_\Sigma)$. Then we claim that

$$\theta \models (l = r) \quad \text{iff} \quad \theta \models (\mathrm{pre}_\mathrm{H}(l) = \mathrm{pre}_\mathrm{H}(r))$$

because $l\theta = \mathrm{H}(\mathrm{pre}_\mathrm{H}(l)\theta)$ and $r\theta = \mathrm{H}(\mathrm{pre}_\mathrm{H}(r)\theta)$. Assume for a proof by contradiction that t is a hom-term of smallest depth where the latter does not hold even though $t\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$, i.e., $t\theta \neq \mathrm{H}(\mathrm{pre}_\mathrm{H}(t)\theta)$. We have $(\mathrm{H}a)\theta = \mathrm{H}a = \mathrm{H}(\mathrm{pre}_\mathrm{H}(\mathrm{H}a)\theta)$ and $([X_i]_\mathrm{H})\theta = \mathrm{H}(X_i\theta) = \mathrm{H}(\mathrm{pre}_\mathrm{H}([X_i]_\mathrm{H})\theta)$. Thus, $t \notin \{\mathrm{H}a, [X_1]_\mathrm{H}, [X_2]_\mathrm{H}\}$. It follows that $t\theta \neq \mathrm{H}a$ because $\mathrm{H}a$ must be a proper subterm of $t\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$. Let π be a proper non-empty prefix of a path in $\Pi_1 \cup \Pi_2 \cup \Pi_a$. As we have already proved, $t\theta \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$ implies $(t\theta).\pi \notin \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$. Since $X_i\theta \neq a$ for every $i \in \{1, 2\}$, the latter implies $t.\pi \notin \{[X_1]_\mathrm{H}, [X_2]_\mathrm{H}\}$, and because $t\theta \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$, all paths $\pi' \in \Pi_1 \cup \Pi_2 \cup \Pi_a$ are defined in t. Since t is of smallest depth, we have $(t.\pi)\theta = \mathrm{H}(\mathrm{pre}_\mathrm{H}(t.\pi)\theta)$ for all $\pi \neq \epsilon$ where $t.\pi$ is defined. By definition of pre_H, we have $\mathrm{H}(\mathrm{pre}_\mathrm{H}(t)\theta) = \mathrm{H}(f(\mathrm{pre}_\mathrm{H}(t.\pi_1), \mathrm{pre}_\mathrm{H}(t.\pi_2))\theta) = \mathrm{H}(f(\mathrm{pre}_\mathrm{H}(t.\pi_1)\theta, \mathrm{pre}_\mathrm{H}(t.\pi_2)\theta)) = (\mathrm{H}f)\theta'$ for the ground substitution $\theta' = \{[X_1]_\mathrm{H} \mapsto \mathrm{H}(\mathrm{pre}_\mathrm{H}(t.\pi_1)\theta), [X_2]_\mathrm{H} \mapsto \mathrm{H}(\mathrm{pre}_\mathrm{H}(t.\pi_2)\theta)\}$. By t's

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

minimal depth, $\theta' = \{[X_1]_{\mathrm{H}} \mapsto (t.\pi_1)\theta, [X_2]_{\mathrm{H}} \mapsto (t.\pi_2)\theta\}$. Since $t\theta \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$, it holds for all $i \in \{1,2\}$ that for all $\pi \in \Pi_i$, $(t.\pi)\theta = (t.\pi_i)\theta$. Therefore, $(\mathrm{H}f)\theta' = t\theta$ – contradiction.

For the given hom-term r (more specifically, for arbitrary hom-terms r which are ground terms or whose sole hom-variable is $[X_2]_{\mathrm{H}}$), we define a formula ϕ_r as a conjunction of term equalitiessuch that for every ground substitution $\theta = \{X_2 \mapsto t\}$ with $t \neq a$, $\theta \models \phi_r$ if and only if $r\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$. The definition of ϕ_r is recursive as follows:

— If r is of the form $[X_2]_{\mathrm{H}}$ or $\mathrm{H}a$, then $r\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$ for all ground substitutions θ. Therefore, ϕ_r is empty, i.e., equivalent to true.
— If $r.\pi$ is defined for all $\pi \in \Pi_1 \cup \Pi_2 \cup \Pi_a$ and $r.\pi = a$ for all $\pi \in \Pi_a$, then ϕ_r consists of

$$\mathrm{pre}_{\mathrm{H}}(r.\pi') = \mathrm{pre}_{\mathrm{H}}(r.\pi'')$$

for all $\pi', \pi'' \in \Pi$ for all $\Pi \in \{\Pi_1, \Pi_2\}$, in conjunction with the equalities of $\phi_{r.\pi}$ for all $\pi \in \Pi_1 \cup \Pi_2$.
— Otherwise, ϕ_r consists of $a = f(a,a)$, i.e., is equivalent to false.

In order to prove that ϕ_r has the specified property, we use structural induction on r. For the base cases, $\phi_{[X_2]_{\mathrm{H}}}$ is equivalent to true, and ϕ_a is true iff $\mathrm{H}a = a$. Hence, the property holds for $r = [X_2]_{\mathrm{H}}$ as well as for $r = a$. Let us therefore now assume that the property holds for all formulas $\phi_{r.\pi}$ where $\pi \neq \epsilon$ and $r.\pi$ are defined. First, assume that r is a ground term t'. Then, $r\theta = t'$ for every ground substitution θ, and the first two items provide a straightforward recursive check whether $t' \in \mathrm{H}(\mathcal{T}_\Sigma)$. In the second item, it is checked that the paths specified by application of H to f exist in t' and the subterm equalities induced by application of H to arbitrary ground terms also hold. Furthermore, it is ensured that $t'.\pi \in \mathrm{H}(\mathcal{T}_\Sigma)$ for all $\pi \in \Pi_1 \cup \Pi_2$.

Now consider the case that r is not ground, i.e., it contains the hom-variable $[X_2]_{\mathrm{H}}$. As already considered in one of the base cases, if r equals $[X_2]_{\mathrm{H}}$, then ϕ_r is true and $[X_2]_{\mathrm{H}}\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$ holds for every ground substitution θ. Assume now that r is different from $[X_2]_{\mathrm{H}}$. Let $\theta = \{X_2 \mapsto t\}$ be a ground substitution with $t \neq a$. Assume that $r \notin \{\mathrm{H}a, [X_2]_{\mathrm{H}}\}$ but $r\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$. Then we have $r\theta \neq \mathrm{H}a$ since by assumption, θ maps X_2 to a term of the form $f(t_1,t_2)$ so that $\mathrm{H}a$ is a proper subterm of $[X_2]_{\mathrm{H}}\theta = \mathrm{H}(X_2\theta) = \mathrm{H}(f(t_1,t_2))$ and thus also of $r\theta$. Therefore, all paths of $\Pi_1 \cup \Pi_2 \cup \Pi_a$ are defined in $r\theta$.

If all paths of $\Pi_1 \cup \Pi_2 \cup \Pi_a$ are already defined in r, then the property of ϕ_r is directly reflected in the second item as it applies the $\mathrm{H}(\mathcal{T}_\Sigma)$-membership test for ground terms on page 54 to the topmost constructors while the recursive part (items (2) and (3) of the membership test) is met by recursive application of ϕ_r to the paths $r.\pi$ (membership test item (2)) and by the pre-image equalities (item (3) of membership test). The latter make use of the fact that $\theta \models (r.\pi' = r.\pi'')$ iff $\theta \models (\mathrm{pre}_{\mathrm{H}}(r.\pi') = \mathrm{pre}_{\mathrm{H}}(r.\pi''))$ provided that $r.\pi'\theta$ and $r.\pi''\theta$ both belong to $\in \mathrm{H}(\mathcal{T}_\Sigma)$ (which, by induction hypothesis, is the case iff $\theta \models (\phi_{r.\pi'} \wedge \phi_{r.\pi''})$) and $X_2\theta \neq a$ (which holds by assumption).

It remains to show that $r\theta \notin \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$ if any of the paths of $\Pi_1 \cup \Pi_2 \cup \Pi_a$ is undefined in r. If $r\theta \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$, then a path $\pi \in \Pi_1 \cup \Pi_2 \cup \Pi_a$ can only be undefined for r if $r.\pi' = [X_2]_{\mathrm{H}}$ (since we assume $[X_2]_{\mathrm{H}}$ to be the only variable in r) for a proper prefix π' of π. But then $(r\theta).\pi'$ must be a complex homomorphic image itself since we assume X_2 to be mapped to a ground term $t \neq a$. The case $\pi' = \epsilon$ (i.e., $r = [X_2]_{\mathrm{H}}$) has already

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

been considered. As we have already proved, there can be no complex homomorphic image at any proper non-empty prefix of a path from $\Pi_1 \cup \Pi_2 \cup \Pi_a$ in $r\theta$ if $r\theta \in \mathrm{H}(\mathcal{T}_\Sigma) \setminus \{\mathrm{H}a\}$. In summary, we conclude that the formula ϕ_r has the specified property.

W.l.o.g., let $\mathcal{B}$ have a predicate p_a with $[\![p_a]\!]_\mathcal{B} = \{a\}$. We now define a set $\mathcal{C}$ of automaton clauses with term equalities which is equivalent to $\mathcal{B}$. Each clause in $\mathcal{B}$ with the constructor a in the head has the form

$$p(a) \Leftarrow \phi.$$

For each such clause, if $a \in [\![p]\!]_\mathcal{B}$, then $\mathcal{C}$ contains the clause $p(a) \Leftarrow$. For each clause in $c \in \mathcal{B}$ with the constructor f in the head, there will be a corresponding clause c' in $\mathcal{C}$. If equalities are sorted by their left-hand sides, each such clause c has the form:

$$p_0(f(X_1, X_2)) \Leftarrow p_1(X_1), p_2(X_2), \phi,$$

where ϕ equals $[X_1]_\mathrm{H} = r_1, \ldots, [X_1]_\mathrm{H} = r_k, [X_2]_\mathrm{H} = r_{k+1}, \ldots, [X_2]_\mathrm{H} = r_m$.

For $i \in \{1, 2\}$, let $\theta_i = \{X_i \mapsto a\}$ and let $j = 3 - i$. Then the only hom-variable possibly occurring in $\phi\theta_i$ is $[X_j]_\mathrm{H}$. Thus, $\phi\theta_i$ is equivalent to a conjunction $\bigwedge_{i \in I} [X_j]_\mathrm{H} = g_i$ for ground terms g_i and a finite set of indexes I. If $\phi\theta_i$ is not equivalent to true or false, then there is a single ground term g with $g = g_i$ for all $i \in I$. Let

$$\phi_i = \begin{cases} X_i = a & \text{if } \phi\theta_i \text{ is equivalent to true} \\ X_i = a, X_j = \mathrm{pre}_\mathrm{H}(g) & \text{if } \phi\theta_i \text{ is equivalent to} [X_j]_\mathrm{H} = g \\ f(a, a) = a & \text{if } \phi\theta_i \text{ is equivalent to false .} \end{cases}$$

Then $\mathcal{C}$ contains the clause c':

$$p_0(f(X_1, X_2)) \quad \Leftarrow \quad p_1(X_1), p_2(X_2), \phi_1 \vee \phi_2 \vee \phi',$$

where ϕ' is defined as

$$X_1 = \mathrm{pre}_\mathrm{H}(r_1), \ldots, X_1 = \mathrm{pre}_\mathrm{H}(r_k), X_2 = \mathrm{pre}_\mathrm{H}(r_{k+1}), \ldots, X_2 = \mathrm{pre}_\mathrm{H}(r_m),\ \phi_r$$

with $\phi_r = \bigwedge_{1 \leqslant j \leqslant m} \phi_{r_j}$, where ϕ_{r_j} are the ϕ_r-formulas for hom-terms r with a sole variable or none, as defined on page 55 ff, i.e., for all ground substitutions θ with $\theta(X_l) \neq a$ for the variable X_l possibly occurring in r_j, $\theta \models \phi_{r_j}$ iff $r_j\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$.

We have to show that $[\![p_0]\!]_\mathcal{B} = [\![p_0]\!]_\mathcal{C}$ for every predicate p_0 of $\mathcal{B}$. Concerning the ground term a, we have $a \in [\![p_0]\!]_\mathcal{B}$ if and only if $a \in [\![p_0]\!]_\mathcal{C}$ by definition of the corresponding clauses in $\mathcal{C}$ with $p_0(a)$ in the head. Now consider elements $f(t_1, t_2) \in [\![p_0]\!]_\mathcal{B} \cup [\![p_0]\!]_\mathcal{C}$. Assume for a proof by contradiction that $[\![p_0]\!]_\mathcal{B} \setminus [\![p_0]\!]_\mathcal{C} \neq \emptyset$. Let $f(t_1, t_2) \in [\![p_0]\!]_\mathcal{B} \setminus [\![p_0]\!]_\mathcal{C}$ have a resolution tree of minimal size. Let $p_0(f(X_1, X_2)) \Leftarrow p_1(X_1), p_2(X_2), \phi$ where ϕ equals $[X_1]_\mathrm{H} = r_1, \ldots, [X_1]_\mathrm{H} = r_k, [X_2]_\mathrm{H} = r_{k+1}, \ldots, [X_2]_\mathrm{H} = r_m$ be the root clause c of a minimal resolution tree for the ground fact $p_0(f(t_1, t_2))$ in $\mathcal{B}$. Let $\phi_1 \vee \phi_2 \vee \phi'$ denote the constraint of the corresponding clause c' in $\mathcal{C}$ as defined above. Let $\theta = \{X_1 \mapsto t_1, X_2 \mapsto t_2\}$. We have $t_1 \in [\![p_1]\!]_\mathcal{B}$, $t_2 \in [\![p_2]\!]_\mathcal{B}$ and $\theta \models \phi$. By assumption on minimality of resolution tree, we also have $t_1 \in [\![p_1]\!]_\mathcal{C}$ and $t_2 \in [\![p_2]\!]_\mathcal{C}$.

First, assume that $t_1 \neq a$ and $t_2 \neq a$. Consider an arbitrary equality $X_i = \mathrm{pre}_\mathrm{H}(r_j)$ and the corresponding sub-formula ϕ_{r_j} of ϕ'. Since $\theta \models [X_i]_\mathrm{H} = r_j$, we have $r_j\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Since the sole variable of r_j is X_{3-i} and $X_{3-i}\theta \neq a$, we have $r_j\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$ iff $\theta \models \phi_{r_j}$, and it follows that $\theta \models \phi_{r_j}$. Furthermore, since $[X_i]_\mathrm{H}\theta, r_j\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$ and $X_i\theta \neq a$ for all $i \in \{1,2\}$, we have $\theta \models X_i = \mathrm{pre}_\mathrm{H}(r_j)$ iff $\theta \models [X_i]_\mathrm{H} = r_j$ – which is true. Therefore, $\theta \models \phi'$, and $f(t_1,t_2) \in [\![p_0]\!]_\mathcal{C}$ by application of the clause $p_0(f(X_1,X_2)) \Leftarrow p_1(X_1), p_2(X_2), \phi'$ from $\mathcal{C}$.

Now assume that $t_i = a$ for some $i \in \{1,2\}$. Let $\theta_i = \{X_i \mapsto a\}$. If $\phi\theta_i$ is equivalent to true, then ϕ_i is defined as $X_i = a$, i.e., $\theta \models \phi_i$ and $f(t_1,t_2) \in [\![p_0]\!]_\mathcal{C}$ by application of the clause $p_0(f(X_1,X_2)) \Leftarrow p_1(X_1), p_2(X_2), \phi_i$ from $\mathcal{C}$. Likewise, if $\phi\theta_i$ is equivalent to $[X_j]_\mathrm{H} = g$, then ϕ_i is defined as $X_i = a, X_j = \mathrm{pre}_\mathrm{H}(g)$. Now, $\theta \models \phi$ only if $\mathrm{H}t_j = g$, i.e., $t_j = \mathrm{H}^{-1}(g)$. Since g is a ground term, $\mathrm{H}^{-1}(g) = \mathrm{pre}_\mathrm{H}(g)$. Therefore, $\theta \models \phi_i$ and $f(t_1,t_2) \in [\![p_0]\!]_\mathcal{C}$ by application of the clause $p_0(f(X_1,X_2)) \Leftarrow p_1(X_1), p_2(X_2), \phi_i$ from $\mathcal{C}$. We conclude that $[\![p]\!]_\mathcal{B} \setminus [\![p]\!]_\mathcal{C} = \varnothing$ for all predicates p of $\mathcal{B}$.

Now assume for a proof by contradiction that for some predicate p_0, $[\![p_0]\!]_\mathcal{C} \setminus [\![p_0]\!]_\mathcal{B} \neq \varnothing$. Recall that for all predicates p, $a \in [\![p]\!]_\mathcal{C}$ iff $a \in [\![p]\!]_\mathcal{B}$. Let $f(t_1,t_2) \in [\![p_0]\!]_\mathcal{C} \setminus [\![p_0]\!]_\mathcal{B}$ be with a resolution tree of minimal size. Let $p_0(f(X_1,X_2)) \Leftarrow p_1(X_1), p_2(X_2), \phi_0$ be the root clause of a minimal resolution tree for the ground fact $p_0(f(t_1,t_2))$ in $\mathcal{C}$. Let $\theta = \{X_1 \mapsto t_1, X_2 \mapsto t_2\}$. We have $t_1 \in [\![p_1]\!]_\mathcal{C}$, $t_2 \in [\![p_2]\!]_\mathcal{C}$, and $\theta \models \phi_0$. By assumption on minimality of resolution tree, we also have $t_1 \in [\![p_1]\!]_\mathcal{B}$ and $t_2 \in [\![p_2]\!]_\mathcal{B}$. Let $p_0(f(X_1,X_2)) \Leftarrow p_1(X_1), p_2(X_2), \phi$ where ϕ equals $[X_1]_\mathrm{H} = r_1, \ldots, [X_1]_\mathrm{H} = r_k, [X_2]_\mathrm{H} = r_{k+1}, \ldots, [X_2]_\mathrm{H} = r_m$ be a clause in $\mathcal{B}$ whose existence causes the existence of the considered clause in $\mathcal{C}$.

According to the construction of $\mathcal{C}$, we have $\phi_0 = \phi_1 \vee \phi_2 \vee \phi'$ as defined above. We have to show that $\theta \models (\phi_1 \vee \phi_2 \vee \phi')$ implies $\theta \models \phi$. Then the original clause from $\mathcal{B}$ proves $f(t_1,t_2) \in [\![p_0]\!]_\mathcal{B}$, a contradiction to our assumption that $f(t_1,t_2) \notin [\![p_0]\!]_\mathcal{B}$. Since this assumption is obtained from the original assumption that $[\![p_0]\!]_\mathcal{C} \setminus [\![p_0]\!]_\mathcal{B} \neq \varnothing$, it follows that the original assumption must be wrong.

First, consider for a fixed $i \in \{1,2\}$ the constraint ϕ_i. Assume therefore that $\theta \models \phi_i$. It follows that $t_i = a$ because ϕ_i either contains the equality $X_i = a$ or is unsatisfiable. Let again $\theta_i = \{X_i \mapsto a\}$. If $\phi\theta_i$ is equivalent to true, then ϕ_i is defined as $X_i = a$. Then $\theta \models \phi$, since θ extends θ_i. If $\phi\theta_i$ is equivalent to $[X_j]_\mathrm{H} = g$, then ϕ_i is defined as $X_i = a, X_j = \mathrm{pre}_\mathrm{H}(g)$. Now, $\theta \models X_j = \mathrm{pre}_\mathrm{H}(g)$ implies $t_j = \mathrm{pre}_\mathrm{H}(g)$ – which in turn implies $\theta \models [X_j]_\mathrm{H} = g$. Therefore, $\theta \models \phi$ also holds.

Now consider the constraint ϕ'. Assume that $\theta \models \phi'$. Let $i \in \{1,2\}$ and let $k = 3 - i$. Consider an equality $X_i = \mathrm{pre}_\mathrm{H}(r_j)$ of ϕ' and the corresponding equalities of ϕ_{r_j} also contained in ϕ'. If $t_k \neq a$, then we have $\theta \models \phi_{r_j}$ if and only if $r_j\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$. For $r_j\theta \in \mathrm{H}(\mathcal{T}_\Sigma)$ and $t_k \neq a$, we have $\theta \models X_i = \mathrm{pre}_\mathrm{H}(r_j)$ iff $\theta \models [X_i]_\mathrm{H} = r_j$. For the corresponding equality $[X_i]_\mathrm{H} = r_j$ of ϕ, we therefore have $\theta \models [X_i]_\mathrm{H} = r_j$ if only $t_k \neq a$. We conclude that for $t_1, t_2 \neq a$, $\theta \models \phi$. Now assume that $t_i = a$. Then, $X_i = \mathrm{pre}_\mathrm{H}(r_j)$ can only be satisfied by θ if either r_j equals $\mathrm{H}a$ or r_j equals $[X_k]_\mathrm{H}$ with $[X_k]_\mathrm{H}\theta = \mathrm{H}a$ since, by definition of pre_H, a term equality $a = \mathrm{pre}_\mathrm{H}(r)$ where r is a hom-term can only be satisfied if r equals $\mathrm{H}a$ or r is a hom-variable. We have $[X_k]_\mathrm{H}\theta = \mathrm{H}a$ iff $X_k\theta = a$ iff $t_k = a$. If r_j equals $\mathrm{H}a$, then $\theta \models [X_i]_\mathrm{H} = r_j$. If r_j equals $[X_k]_\mathrm{H}$ and $t_k = a$, then $\theta \models [X_i]_\mathrm{H} = r_j$ since, by assumption, $t_i = a$. Therefore, in all cases, $\theta \models \phi$. We conclude that $[\![p]\!]_\mathcal{C} \setminus [\![p]\!]_\mathcal{B} = \varnothing$ for all predicates p of $\mathcal{C}$. To summarize, for $\mathcal{B}$, there exists a finite set $\mathcal{C}$ of automaton clauses with term equalities.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Case 2: Hf is of the form $[X_1]_{\mathrm{H}}$ or $[X_2]_{\mathrm{H}}$. Then, Ht = Ha for every ground term t (over the alphabet $\{f, a\}$). Thus, each hom-equality $[X_i]_{\mathrm{H}} = r$ is either vacuously true and therefore can be omitted or vacuously false implying that the whole clause can be omitted.

Case 3: Hf is a ground term g. Then each Boolean combination of hom-equalities gives rise to four cases, depending on whether $X_1\theta = a$ and $X_2\theta = a$ for the applied ground substitution θ. I.e., in $\mathcal{B}$, there are at most $2^4 = 16$ semantically different constraints. Furthermore, the information needed to evaluate a constraint can be recorded in the predicates. Applying the construction from Lemma 3.3 for the case of automaton clauses, we construct from $\mathcal{B}$ a set $\mathcal{B}'$ of automaton clauses with hom-equalities as the minimal set according to the following rules. For each clause $p(a) \Leftarrow \phi$ of $\mathcal{B}$, $\mathcal{B}'$ also contains that clause and additionally the clause $p_a(a) \Leftarrow \phi$. For each clause $p(f(X_1, X_2)) \Leftarrow q(X_1), r(X_2), \phi$ in $\mathcal{B}$, $\mathcal{B}'$ has all clauses

$$p'(f(X_1, X_2)) \Leftarrow q'(X_1), r'(X_2), \phi$$

for $p' \in \{p, p_f\}, q' \in \{q_f, q_a\}, r' \in \{r_f, r_a\}$. Then, $[\![p]\!]_{\mathcal{B}'} = [\![p]\!]_{\mathcal{B}}$, $[\![p_a]\!]_{\mathcal{B}'} = [\![p]\!]_{\mathcal{B}} \cap \{a\}$, and $[\![p_f]\!]_{\mathcal{B}'} = [\![p]\!]_{\mathcal{B}} \cap \{f(t_1, t_2) \mid t_1, t_2 \in \mathcal{T}_\Sigma\}$. In $\mathcal{B}'$, we can remove the hom-variables in the constraints as follows, ending up with ordinary term equalities without variables – which are either vacuously true or vacuously false. Let $p'(f(X_1, X_2)) \Leftarrow q_c(X_1), r_d(X_2), \phi$ with $c, d \in \{a, f\}$ be a clause in $\mathcal{B}'$ whose constraint ϕ contains a hom-variable X_i. Then we replace the clause by the clause $p'(f(X_1, X_2)) \Leftarrow q_c(X_1), r_d(X_2), \phi[\mathrm{H}c/X_1, \mathrm{H}d/X_2]$.

Thus, in these first three cases of a tree homomorphism, there exists a set $\mathcal{C}$ of automaton clauses with ordinary term equalities which is equivalent to $\mathcal{B}$. For such sets of automaton clauses, however, a proof has been reported in Seidl and Reuß (2012) that they cannot express the language $\overline{T}$. This proof is repeated here for self-containedness. W.l.o.g., we may again assume that $\mathcal{C}$ has conjunctive constraints only. Recall that $[\![\overline{p}]\!]_{\mathcal{C}} = [\![\overline{p}]\!]_{\mathcal{B}} = \overline{T}$, where $\overline{p}$ is the complementary predicate of p from $\mathcal{A}$ with $[\![p]\!]_{\mathcal{A}} = T$. Therefore, $\mathcal{C}$ must contain a clause c of the form

$$\overline{p}(f(X_1, X_2)) \Leftarrow q_1(X_1), q_2(X_2), \phi,$$

where ϕ is a finite conjunction of term equalities, such that there are distinct terms $t_1, t_2 \in [\![q_1]\!]_{\mathcal{C}}$ and for $i = 1, 2$ there are two distinct terms t_{i1}, t_{i2} such that $f(t_i, t_{ij}) \in [\![q_2]\!]_{\mathcal{C}}$ and $f(t_i, f(t_i, t_{ij})) \in [\![\overline{p}]\!]_{\mathcal{C}}$ through application of that clause. For $i = 1, 2$ and $j = 1, 2$, consider the substitution θ_{ij} defined by $\theta_{ij}(X_1) = t_i$ and $\theta_{ij}(X_2) = f(t_i, t_{ij})$. Then, in particular, $\theta_{ij} \models \phi$. In order to see that such terms t_i and t_{ij} exist, we first claim that for every term t, there must be some clause c_t by which for two distinct terms s_1, s_2, facts $\overline{p}(f(t, f(t, s_1)))$ and $\overline{p}(f(t, f(t, s_2)))$ can be obtained. Assume for a proof by contradiction that this claim were wrong. Then for some t and every clause c for predicate $\overline{p}$, there would be at most one term t_c such that a fact $\overline{p}(f(t, f(t, t_c)))$ can be obtained by application of c. Accordingly, for this t, the set $\{s \mid f(t, f(t, s)) \in \overline{T}\}$ would be finite – which is not the case. Consequently, our claim holds, i.e., for every t, we can find a clause c_t by which for two distinct terms s_1, s_2, facts $\overline{p}(f(t, f(t, s_1)))$ and $\overline{p}(f(t, f(t, s_2)))$ can be obtained. Since the number of terms is infinite while the number of clauses is finite, we conclude that there

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

must be two distinct terms t_1, t_2 for which the clauses c_{t_1} and c_{t_2} coincide – giving us the terms t_i and t_{ij} together with the clause c as desired.

Now recall that each finite conjunction of term equalities $s_i = s_j$ between arbitrary terms s_i, s_j can be expressed as a finite conjunction of term equalities of the form $Z = s$ for variables Z. W.l.o.g., let ϕ be of this form. Furthermore, recall that a term equality $Z = s$ where Z occurs in s either is trivially true or trivially false. In addition to equalities $X_1 = X_1$ and $X_2 = X_2$, the satisfiable constraint ϕ therefore can only contain equalities of one of the following forms:

a. $X_1 = g$ or $X_2 = g$, where g denotes a ground term.
b. $X_1 = s$, where X_2 occurs in s.
c. $X_2 = s$, where X_1 occurs in s but X_2 does not.

In the following, we show that an equality of any of these forms leads to a contradiction.

Case (a) ϕ contains an equality $X_r = g$ for some ground term g. First, assume that $r = 1$. We have either $t_1 \neq g$ or $t_2 \neq g$ – implying that $\theta_{ij} \not\models (X_r = g)$ for some i, j. Second, assume that $r = 2$. We have either $f(t_1, t_{11}) \neq g$ or $f(t_1, t_{12}) \neq g$ and hence also $\theta_{ij} \not\models (X_r = g)$ for some i, j.

Case (b) ϕ contains an equality $X_1 = s$, where X_2 occurs in s. If $\theta_{11} \models (X_1 = s)$, then $f(t_1, t_{11})$ would be a subterm of t_1 – which is impossible.

Case (c) Finally, assume that there is an equality $X_2 = s$ where X_1 occurs in s but X_2 does not. Consider the two substitutions θ_{11} and θ_{12}. If $\theta_{11} \models (X_2 = s)$, we conclude that $f(t_1, t_{11}) = \theta_{11}(X_2) = s[t_1/X_1]$. Then, $\theta_{12} \models (X_2 = s)$ implies that $s[t_1/X_1]$ also equals $f(t_1, t_{12})$ – from which it follows that $f(t_1, t_{11}) = f(t_1, t_{12})$. This, however, implies $t_{11} = t_{12}$ in contradiction to our assumption.

We conclude that the conjunction ϕ is equivalent to true. But then $[\![\overline{p}]\!]_C$ must also contain the term $f(t_1, f(t_2, t_{21}))$ – which is not contained in $\overline{T}$.

Let us now continue with the case distinction on properties of the tree homomorphism H.

Case 4: The hom-term $\mathrm{H}f$ equals $t = f(t'_1, t'_2)$ where t's sole variable is $[X_1]_{\mathrm{H}}$ (respectively $[X_2]_{\mathrm{H}}$). Then we define $d(t)$ to be the maximal $i \geqslant 0$ so that the path $(f,1)^i$ (respectively $(f,2)^i$) is defined for t. Then for all ground terms l, r, $\mathrm{H}l = \mathrm{H}r$ iff $d(l) = d(r)$. We now define an infinite set $S = \{s_1, s_2, \ldots\}$ where $s_1 = f(a,a)$ and for $i > 1$, s_i is inductively defined as follows. If the only hom-variable in $\mathrm{H}f$ is $[X_1]_{\mathrm{H}}$, then $s_i = f(s_{i-1}, a)$. Likewise, if the only hom-variable in $\mathrm{H}f$ is $[X_2]_{\mathrm{H}}$, then $s_i = f(a, s_{i-1})$. From this definition, it follows for any terms $s, t \in S$ with $s \neq t$ that $d(s) \neq d(t)$ and hence, $\mathrm{H}s \neq \mathrm{H}t$ holds. This set S now allows for an analogous Ramsey-type argument as for term equalities above. Thus, we deduce that $\mathcal{B}$ must contain a clause of the form

$$\overline{p}(f(X_1, X_2)) \Leftarrow q_1(X_1), q_2(X_2), \phi,$$

where ϕ is a finite conjunction of hom-equalities, such that there are distinct terms $t_1, t_2 \in [\![q_1]\!]_{\mathcal{B}} \cap S$ and for $i = 1, 2$ there are two distinct terms $t_{i1}, t_{i2} \in S$ such that $f(t_i, t_{ij}) \in [\![q_2]\!]_{\mathcal{B}}$ and $f(t_i, f(t_i, t_{ij})) \in [\![\overline{p}]\!]_{\mathcal{B}}$ through application of that clause. We observe that for a given ground substitution θ with $\theta(X_1) = g_1$ and $\theta(X_2) = g_2$, the evaluation of

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

ϕ depends solely on the values of $d(g_1)$ and $d(g_2)$. Recalling the definition of the set S, if $[X_1]_{\mathrm{H}}$ is the only hom-variable that occurs in $\mathrm{H}f$, then we have $d(f(t_i, t_{ij})) = d(t_i) + 1$, and otherwise, we have $d(f(t_i, t_{ij})) = d(t_{ij}) + 1$. Hence, the equalities of ϕ can depend on $d(t_i)$ and possibly on $d(t_{ij})$.

Case 4.1 There is some $c \geqslant 0$ such that ϕ is true only when $d(t_i) = c$, or when $d(t_{ij}) = c$. This contradicts the fact that $d(t_1) \neq d(t_2)$ and also $d(t_{11}) \neq d(t_{12})$, since the terms are chosen from S.

Case 4.2 There is some integer c such that $\theta \models \phi$ iff $d(t_i) = d(t_{ij}) + c$. This is again a contradiction to $d(t_{11}) \neq d(t_{12})$, since it implies that $f(t_1, f(t_1, t_{1j})) \in [\![\overline{p}]\!]_{\mathcal{B}}$ is not proved by application of the given clause for at least one $j \in \{1, 2\}$.

Case 4.3 The constraint ϕ is equivalent to true. Then the term $f(t_1, f(t_2, t_{22}))$ is contained in $[\![\overline{p}]\!]_{\mathcal{B}}$. But $f(t_1, f(t_2, t_{22}))$ is not in $\overline{T}$, since $t_1 \neq t_2$ – contradiction.

Thus, we have proven that $\overline{T}$ cannot be accepted by any finite set of automaton clauses with hom-equalities. Consequently, the original language T cannot be accepted by any finite set of automaton clauses with hom-disequalities. This completes the proof.

References

Baskar, A., Ramanujam, R. and Suresh, S.P. (2007). Knowledge-based modelling of voting protocols. In: *TARK*, Presses universitaires de Louvain, 62–71, ISBN 978-2-87463-077-4.

Blanchet, B. (2001). An efficient cryptographic protocol verifier based on prolog rules. In: *CSFW*, IEEE Computer Society, 82–96. ISBN 0-7695-1146-5.

Comon, H. and Jacquemard, F. (1994). Ground reducibility and automata with disequality constraints. In: *STACS*, Lecture Notes in Computer Science, vol. 775, Springer, 151–162.

Creus, C., Gascón, A., Godoy, G. and Ramos, L. (2012). The hom problem is exptime-complete. In: *LICS*, IEEE, 255–264. ISBN ISBN 978-1-4673-2263-8.

Frühwirth, T.W., Shapiro, E.Y., Vardi, M.Y. and Yardeni, E. (1991). Logic programs as types for logic programs. In: *LICS*, IEEE Computer Society, 314–328. ISBN 0-8186-2230-X.

Fujioka, A., Okamoto, T. and Ohta, K. (1993). A practical secret voting scheme for large scale elections. In: *AUSCRYPT*, Lecture Notes in Computer Science, vol. 718, Springer, 244–251.

Godoy, G., Giménez, O., Ramos, L. and Àlvarez, C. (2010). The hom problem is decidable. In: *STOC*, ACM, 485–494. ISBN 978-1-4503-0050-6.

Goubault-Larrecq, J. (2005). Deciding H1 by resolution. *Information Processing Letters* **95** (3) 401–408.

Goubault-Larrecq, J. and Parrennes, F. (2005). Cryptographic protocol analysis on real C code. In: *VMCAI*, Lecture Notes in Computer Science, vol. 3385, Springer, 363–379.

Müller-Olm, M., Rüthing, O. and Seidl, H. (2005) Checking Herbrand equalities and beyond. In: *VMCAI*, Lecture Notes in Computer Science, vol. 3385, Springer, 79–96.

Nielson, F., Nielson, H.R. and Seidl, H. (2002). Normalizable horn clauses, strongly recognizable relations, and Spi. In: *SAS*, Lecture Notes in Computer Science, vol. 2477, Springer, 20–35.

Reuß, A. and Seidl, H. (2010). Bottom-up tree automata with term constraints. In: *LPAR-17*, Lecture Notes in Computer Science, vol. 6397, Springer, 581–593.

Reuß, A. and Seidl, H. (2012). Crossing the syntactic barrier: Hom-disequalities for H1-clauses. In: *CIAA*, Lecture Notes in Computer Science, vol. 7381, Springer, 301–312.

Seidl, H. and Neumann, A. (1999). On guarding nested fixpoints. In: *CSL*, Lecture Notes in Computer Science, vol. 1683, Springer, 484–498.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press

Seidl, H. and Reuß, A. (2011). Extending H1-clauses with disequalities. *Information Processing Letters* **111** (20) 1007–1013.

Seidl, H. and Reuß, A. (2012). Extending H1-clauses with path disequalities. In: *FoSSaCS*, Lecture Notes in Computer Science, vol. 7213, Springer, 165–179.

Weidenbach, C. (1999). Towards an automatic analysis of security protocols in first-order logic. In: *CADE*, Lecture Notes in Computer Science, vol. 1632, Springer, 314–328.

https://doi.org/10.1017/S096012951700024X Published online by Cambridge University Press