Available online at www.sciencedirect.com
SCIENCE DIRECT®

Information Processing Letters 95 (2005) 401–408

Information Processing Letters

www.elsevier.com/locate/ipl

Deciding $\mathcal{H}_1$ by resolution [☆]

Jean Goubault-Larrecq [1]

LSV/UMR 8643, CNRS, ENS Cachan & INRIA Futurs project SECSI, 61, av. du président-Wilson, 94235 Cachan Cedex, France

Received 1 January 2005; received in revised form 23 March 2005; accepted 25 April 2005

Available online 4 June 2005

Communicated by D. Basin

Abstract

Nielson, Nielson and Seidl's class $\mathcal{H}_1$ is a decidable class of first-order Horn clause sets, describing strongly regular relations. We give another proof of decidability, and of the regularity of the defined languages, based on fairly standard automated deduction techniques.

Keywords: Automatic theorem proving; Formal languages

Nielson et al. [6] introduced the class $\mathcal{H}_1$ of first-order Horn clause sets to model reachability in the spi-calculus. They showed that $\mathcal{H}_1$ satisfiability is decidable and DEXPTIME-complete, and that $\mathcal{H}_1$ clause sets can be converted to equivalent tree automata in exponential time—so $\mathcal{H}_1$ defines regular tree languages. Further subclasses of $\mathcal{H}_1$, namely $\mathcal{H}_2$ and $\mathcal{H}_3$, have polynomial time complexity.

Our objective is to make it clear that fairly standard automated deduction techniques yield streamlined proofs of these facts, and in passing to introduce a slightly simpler definition of $\mathcal{H}_1$. We reprove most results of Nielson et al. [6] this way, sometimes correcting small inaccuracies.

$_\flat\mathcal{H}_1$ and $\mathcal{H}_1$. As usual, we fix a first-order signature, which we shall leave implicit. Terms are denoted s, t, u, v, …, predicate symbols P, Q, …, variables X, Y, Z, …. We assume there are finitely many predicate symbols. Horn clauses C are of the form $H \Leftarrow \mathcal{B}$ where the *head* H is either an atom or $\perp$, and the *body* $\mathcal{B}$ is a finite set $A_1, \dots, A_n$ of atoms. If $\mathcal{B}$ is empty ($n = 0$), then $C = H$ is a *fact*.

Definition 1. An $_\flat\mathcal{H}_1$ *clause* is any Horn clause $H \Leftarrow P_1(t_1), \dots, P_n(t_n)$, where all predicate symbols are unary, $t_1, \dots, t_n$ are arbitrary, and the head H is either

☆ Partially supported by the RNTL projects EVA and Prouvé, the ACI SI Rossignol, and the young ACI researchers “Sécurité informatique, proto. crypto. et détection d’intrusions”.

E-mail address: goubault@lsv.ens-cachan.fr (J. Goubault-Larrecq).

URL: http://www.lsv.ens-cachan.fr/~goubault/.

[1] Fax: +33 1 47 40 75 21.

0020-0190/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2005.04.007

$\perp$, of the form $P(X)$ or of the form $P(f(X_1, \ldots, X_k))$ where $X_1, \ldots, X_k$ are distinct.

This is fairly unrestricted: apart from the use of unary predicates, which is innocuous—encode k-ary relations $P(t_1, \ldots, t_k)$ as $P(c(t_1, \ldots, t_k))$—the only restrictions on ${}_\flat\mathcal{H}_1$ clauses are on heads. Let us restate Nielson et al.'s [6] definition. We depart from op.cit. only in that we only use unary predicates, to ease comparison. It should be clear that we do not lose any generality this way.

Definition 2. An $\mathcal{H}_1$ *clause* is any Horn clause $H \Leftarrow P_1(t_1), \ldots, P_n(t_n)$ where all predicate symbols are unary, $t_1, \ldots, t_n$ are arbitrary, and: (a) Its head H is *linear*, i.e., no variable occurs twice in H; and (b) any two variables X, Y that are connected in the body and occur in H must be siblings in H.

This requires auxiliary definitions: given a Horn clause as above, let $\cong$ ("is connected to") be the smallest equivalence relation such that $X \cong Y$ if both X and Y occur in the same $P_i(t_i)$ for some i, $1 \leqslant i \leqslant n$; X and Y are *siblings* in H iff there is a subterm $f(t_1, \ldots, t_n)$ of H such that $X = t_i$ and $Y = t_j$ for some i, j, $1 \leqslant i, j \leqslant n$. Every ${}_\flat\mathcal{H}_1$ clause is in $\mathcal{H}_1$; Proposition 4 below is a form of converse.

The main value of ${}_\flat\mathcal{H}_1$ to us is that it provides a straightforward descriptive typing discipline for Horn clause sets, à la Frühwirth et al. [3]. Define the rewrite relation $\rightsquigarrow$ on clause sets by (we use the letter $\mathcal{B}$ to denote finite sets of atoms, and $\mathcal{B}\{X := t\}$ to denote $\mathcal{B}$ with t substituted for X):

$$P\big(\mathcal{C}[t]\big) \Leftarrow \mathcal{B} \quad \rightsquigarrow \quad \begin{cases} P(\mathcal{C}[Z]) \Leftarrow \mathcal{B}, Q(Z) \\ \quad (Z \text{ fresh}) \\ Q(t) \Leftarrow \mathcal{B} \end{cases} \tag{1}$$

where t is not a variable, Q is a fresh predicate symbol, and $\mathcal{C}[]$ is a non-trivial one-hole context,

$$P\big(\mathcal{C}[X]\big) \Leftarrow \mathcal{B} \quad \rightsquigarrow \quad P\big(\mathcal{C}[Y]\big) \Leftarrow \mathcal{B}, \mathcal{B}\{X := Y\}, \tag{2}$$

where X occurs at least twice in $\mathcal{C}[X]$, Y is a fresh variable.

A one-hole context $\mathcal{C}[]$ is a term with a distinguished occurrence of the *hole* $[]$. $\mathcal{C}[u]$ is $\mathcal{C}[]$ with u in place of the hole. $\mathcal{C}[]$ is *non-trivial* iff $\mathcal{C}[] \neq []$. In (2), we require X to occur at least twice, with one occurrence distinguished by the hole in $\mathcal{C}[]$.

Proposition 3. *Starting from a clause set S_0, $\rightsquigarrow$ terminates in polynomially many steps. Any $\rightsquigarrow$-normal form of S_0 is an ${}_\flat\mathcal{H}_1$ clause set S_* that logically implies S_0.*

Proof. Let $|t|$ be defined by $|X| = 0$, $|f(t_1, \ldots, t_n)| = |t_1| + \cdots + |t_n| + 1$. Define the *defect* ∂t of t as $|t| - 1 = |t_1| + \cdots + |t_n|$ if $t = f(t_1, \ldots, t_n)$, 0 otherwise. The *defect* of an atom $P(t)$ is ∂t, that of $\perp$ is 0. The *defect* ∂C of a clause C is that of its head. The *defect* ∂S of a Horn clause set is $\sum_{C \in S} \partial C$. Clearly $\partial\mathcal{C}[t] = \partial\mathcal{C}[Z] + |t|$ when $\mathcal{C}[u]$ is a non-trivial one-hole context, so $\partial\mathcal{C}[Z] + \partial t = \partial\mathcal{C}[Z] + |t| - 1 < \partial\mathcal{C}[t]$ when t is not a variable; it follows that (1) makes ∂S decrease strictly, while (2) leaves it invariant. So in any rewrite sequence from S_0, (1) is applied at most ∂S_0 times, generating at most ∂S_0 additional clauses.

If X occurs in t, let the *excess* of X in t be the number of occurrences of X in t minus one, and εt be the sum of all excesses of variables in t. Note that $\varepsilon t = 0$ iff t is linear. Define εC as εt where C has head $P(t)$, 0 otherwise, and $\varepsilon S = \sum_{C \in S} \varepsilon C$. Then εS decreases strictly with (2). Moreover, in any rewrite from S_0 to S, εS is bounded by $\max_{C \in S} \varepsilon C$ times the number $\#S$ of clauses in S. The former is bounded by $\max_{C \in S_0} \varepsilon C$, since no rule creates any new clause with a higher excess, and the latter is bounded by $\#S_0 + \partial S_0$. So (2) can only be applied $\max_{C \in S_0} \varepsilon C \times (\#S_0 + \partial S_0)$ times. So $\rightsquigarrow$ terminates in polynomially many steps.

Any normal form S_* of S_0 for $\rightsquigarrow$ is clearly in ${}_\flat\mathcal{H}_1$. That S_* implies S_0 is because the left-hand side of each $\rightsquigarrow$ rule is implied by the right-hand side: in (1) the left-hand side is a resolvent (on Q) of the right-hand clauses, in (2) it is an instance. $\square$

Although $\rightsquigarrow$ terminates in polynomially many steps, it need not terminate in polynomial *time*: starting from a clause with k variables in the head, each occurring $q + 1$ times, rule (2) ends up producing a clause whose body contains q^k instances of $\mathcal{B}$. On sets of clauses with linear heads, e.g., $\mathcal{H}_1$ clause sets, (2) never applies, so $\rightsquigarrow$ does terminate in polynomial time.

For each satisfiable set S of Horn clauses, and each predicate symbol P, let $L_P(S)$ be the set of ground terms t such that $P(t)$ is in the least Herbrand model of S. $L_P(S)$ is the *language* recognized at *state* P. In the particular case that S consists only of *automaton clauses* $P(f(X_1, \ldots, X_n)) \Leftarrow P_1(X_1)$,

$\ldots, P_n(X_n)$ ($X_1, \ldots, X_n$ pairwise distinct), this coincides with the usual definition of the set of terms recognized at P; such clauses are just tree automaton transitions from $P_1, \ldots, P_n$ to P. Accordingly, we call a set of automaton clauses a (tree) *automaton*. This connection between tree automata and Horn clauses was pioneered by Frühwirth et al. [3]; there, $L_P(S)$ is called the *success set* for P. The point is that least Herbrand models naturally generalize $L_P(S)$ to sets S that are not just tree automata.

By Proposition 3, for each predicate P in S_0, $L_P(S_*) \supseteq L_P(S_0)$. I.e., S_* provides *upper approximants* for success sets (languages) defined by S_0—$L_P(S_*)$ is a *type* of P in S_0. When S_0 is a set of $\mathcal{H}_1$ clauses, Proposition 3 can be refined:

Proposition 4. *If S_0 is a set of $\mathcal{H}_1$ clauses, and $S_0 \leadsto^* S_*$, then S_0 and S_* are equivalent up to auxiliary predicates, that is, either both are unsatisfiable, or both are satisfiable and $L_P(S_0) = L_P(S_*)$ for every P occurring in S_0.*

Proposition 4 is subsumed by Nielson et al. [6, Proposition 2] (whose proof was omitted). There, a linear time complexity is claimed. This must be taken with a grain of salt. On Turing machines, the best we can claim is that ${}_\flat\mathcal{H}_1$ and $\mathcal{H}_1$ have equivalent expressive power, up to polynomial time reductions—since $\leadsto$ terminates in polynomial time in this case.

Proof. Starting from $\mathcal{H}_1$ clauses, $\leadsto$ only produces $\mathcal{H}_1$ clauses, and (2) never applies. We claim that if S is a set of $\mathcal{H}_1$ clauses and $S \leadsto S'$ (by (1)), then S logically implies S' *up to* auxiliary predicates, i.e., every (Herbrand) model I of S induces a (Herbrand) model I' of S' whose restriction to the predicates of S is I. This will then prove the proposition.

Let $P(\mathcal{C}[t]) \Leftarrow \mathcal{B}$ be the clause rewritten by (1) in S, generating the fresh predicate Q. Build I' by extending I with all ground atoms $Q(t\sigma)$, where σ is any grounding substitution such that $\mathcal{B}\sigma$ holds in I. We claim that $P(\mathcal{C}[Z]) \Leftarrow \mathcal{B}, Q(Z)$ is valid in I'. Otherwise, there would be a grounding substitution σ' such that $P(\mathcal{C}[Z]\sigma')$ is false in I' but all atoms in $\mathcal{B}\sigma'$ as well as $Q(u)$ are true in I', where $u = \sigma'(Z)$. By definition of I', u is of the form $t\sigma$, where $\mathcal{B}\sigma$ holds in I. We now merge σ and σ' as follows. By Definition 2(b), no variable in t is connected to any variable in $\mathcal{C}[\,]$: partition variables into $\mathcal{V}_1$, the set of variables connected (in $\mathcal{B}$) to some variable occurring in t, and its complement $\mathcal{V}_2$. Let σ'' map each $X \in \mathcal{V}_1$ to $\sigma(X)$, and each $X \in \mathcal{V}_2$ to $\sigma'(X)$. In particular, $\mathcal{C}[Z]\sigma'' = \mathcal{C}[Z]\sigma'$ since no variable occurring in $\mathcal{C}[Z]$ is connected to any variable of t. So $P(\mathcal{C}[Z]\sigma'')$ is false in I'. Since $Z\sigma'' = Z\sigma' = u = t\sigma = t\sigma''$, $P(\mathcal{C}[t]\sigma'')$ is false in I'. Also, all atoms of $\mathcal{B}\sigma''$ are either of the form $A\sigma$ or $A\sigma'$ with $A \in \mathcal{B}$, and are therefore true in I'. So σ'' falsifies $P(\mathcal{C}[t]) \Leftarrow \mathcal{B}$, contradiction. □

Deciding ${}_\flat\mathcal{H}_1$ by resolution. Ordered resolution with selection is a complete refinement of resolution, parameterized by a stable strict reduction ordering $\succ$ on atoms and a *selection function* mapping each clause to a subset of its negative literals [1]. In the case of Horn clauses, ordered resolution with selection can be stated thus: from the *main* premise $A \Leftarrow \mathcal{B}, A_1, \ldots, A_m$ (where $A_1, \ldots, A_m$, $m \geqslant 1$, is the set of selected atoms if any atom is selected at all, or $m = 1$ and A_1 is $\succ$-maximal in the whole clause if no atom is selected), and the *m side* premises $A'_i \Leftarrow \mathcal{B}'_i$, $1 \leqslant i \leqslant m$ (where no atom is selected and A'_i is $\succ$-maximal in each), infer the *resolvent* $A\sigma \Leftarrow \mathcal{B}\sigma, \mathcal{B}'_1\sigma, \ldots, \mathcal{B}'_m\sigma$ (where σ is the simultaneous most general unifier of A_1 with $A'_1, \ldots, A_m$ with A'_m). The resolvent is added to the current set of clauses. This is complete in the sense that S is unsatisfiable iff the empty clause $\perp$ can be deduced by finitely many instances of this rule from S. Completeness is retained also in the presence of redundancy elimination rules [1].

Call an *ε-block* any finite set of atoms of the form $P_1(X), \ldots, P_m(X)$ (with the same X, and $m \geqslant 0$); it is *non-empty* iff $m \geqslant 1$. We shall abbreviate ε-blocks $B(X)$ to make the variable X explicit. We say that $B(X)$ is a block *of* the clause $A \Leftarrow \mathcal{B}, B(X)$ iff X occurs neither in A nor in $\mathcal{B}$. A *deep* atom is any atom of the form $P(f(\ldots))$, i.e., not an atom of the form $P(X)$ or $\perp$.

We shall use an additional rule, which is a variant of the *splitting without backtracking* rule of Riazanov and Voronkov [7], namely the *ε-splitting* rule of Goubault-Larrecq et al. [4]: for each non-empty ε-block $B(X)$, create a fresh nullary predicate symbol q_B; now, if $B(X)$ is a non-empty block of $A \Leftarrow \mathcal{B}, B(X)$, then replace the latter by the two clauses $A \Leftarrow \mathcal{B}, q_B$ and $q_B \Leftarrow B(X)$. (Intuitively, the latter *defines* q_B to hold whenever the intersection of the languages of predi-

$$\frac{P(f(\overline{X}^k)) \Leftarrow B_1(X_1), \ldots, B_k(X_k) \qquad H \Leftarrow P(f(t_1, \ldots, t_k)), \mathcal{B}}{H \Leftarrow \mathcal{B}, B_1(t_1), \ldots, B_k(t_k)} \tag{3}$$

$$\frac{\overbrace{P_j(f(\overline{X}^k)) \Leftarrow B_{j1}(X_1), \ldots, B_{jk}(X_k)}^{1 \leqslant j \leqslant \ell, \ell \geqslant 1} \quad \overbrace{P_j(X)}^{\ell+1 \leqslant j \leqslant m} \qquad H \Leftarrow P_1(X), \ldots, P_m(X)}{H \Leftarrow B_{11}(X_1), \ldots, B_{\ell 1}(X_1), \ldots, B_{1k}(X_k), \ldots, B_{\ell k}(X_k)} \tag{4}$$

where $H = \perp$ or H is a splitting literal q

$$\frac{\overbrace{P_j(f(\overline{X}^k)) \Leftarrow B_{j1}(X_1), \ldots, B_{jk}(X_k)}^{1 \leqslant j \leqslant \ell, \ell \geqslant 1} \quad \overbrace{P_j(X)}^{\ell+1 \leqslant j \leqslant m} \qquad P(X) \Leftarrow P_1(X), \ldots, P_m(X)}{P(f(\overline{X}^k)) \Leftarrow B_{11}(X_1), \ldots, B_{\ell 1}(X_1), \ldots, B_{1k}(X_k), \ldots, B_{\ell k}(X_k)} \tag{5}$$

$$\frac{q \qquad H \Leftarrow \mathcal{B}, q}{H \Leftarrow \mathcal{B}} \tag{6}$$

$$\frac{P_1(X) \ \ldots \ P_m(X) \quad H \Leftarrow \mathcal{B}, P_1(t_1), \ldots, P_m(t_m)}{H \Leftarrow \mathcal{B}} \tag{7}$$

Fig. 1. Specializing resolution to ${}_\flat\mathcal{H}_1^{a,*}$ clauses.

cates in B is non-empty, and will be called a *defining clause*. The former allows one to conclude A from $\mathcal{B}$, as soon as the *splitting atom* q_B holds.) This replacement rule can be applied anytime without breaking completeness, provided A or $\mathcal{B}$ contains at least one atom of the form $P(t)$ (for any t), and the ordering $\succ$ is extended so that $P(t) \succ q_B$ for every unary predicate symbol P, every term t, and every splitting atom q_B. This is an easy consequence of Bachmair and Ganzinger's [1, Section 4.2.2] standard redundancy criterion.

To decide ${}_\flat\mathcal{H}_1$, let $P(s) \succ Q(t)$ iff s is a proper superterm of t, regardless of P and Q. Define the selection function as follows: if the clause body contains some splitting atom q_B, select one; otherwise, if it contains a deep atom, select one; otherwise, select all atoms in the body if the head is not deep; else none. For short, call *resolution* this special brand of ordered resolution with selection.

We now show that, starting from a set of ${}_\flat\mathcal{H}_1$ clauses, all clauses produced by resolution are of a special form, provided ε-splitting is used *eagerly* (i.e., as soon as possible). The latter is not required to decide ${}_\flat\mathcal{H}_1$, only to reach optimal complexity bounds. Write $\overline{X}^k$ for the sequence of pairwise distinct variables $X_1, \ldots, X_k$.

Proposition 5. *Let a be an upper bound on arities of function symbols. Call ${}_\flat\mathcal{H}_1^{a,*}$ the class of non-ε-splittable clauses of the form $H \Leftarrow P_1(t_1), \ldots, P_n(t_n)$, $q_{B_1}, \ldots, q_{B_m}$, where $B_1, \ldots, B_m$ are non-empty ε-blocks, $m \leqslant a$, and H is $\perp$, a splitting symbol q_B, or an atom $P(X)$ or $P(f(\overline{X}^k))$ (the latter two forms being as in Definition* 1*). Applying resolution to ${}_\flat\mathcal{H}_1^{a,*}$ yields clauses that are or that ε-split into clauses of ${}_\flat\mathcal{H}_1^{a,*}$. All resolution steps are of one of the forms shown in Fig.* 1.

Proof. Because of selection, the only clauses that can be used as side premises, and which cannot be ε-split further, are unit clauses q_B, or so-called *universal clauses* $P(X)$, or so-called *alternating automaton clauses* of the form

$$P\big(f(X_1, \ldots, X_k)\big) \Leftarrow B_1(X_1), \ldots, B_k(X_k), \tag{8}$$

where $B_1(X_1), \ldots, B_k(X_k)$ are possibly empty ε-blocks. (When each $B_i(X_i)$ contains exactly one atom, these are just automaton clauses. General ε-blocks encode intersection of languages inside transitions, whence the *alternating* qualifier.)

Assume some atom q_{B_i} is selected in the main premise. We can only resolve it with a unit clause q_{B_i} (if any), resulting in an ${}_\flat\mathcal{H}_1^{a,*}$ clause by Rule (6). Otherwise, if the main premise is of the form $H \Leftarrow \mathcal{B}, P(t)$, where $P(t)$ is deep and selected, say $t = f(t_1, \ldots, t_k)$, then we may resolve it either with a universal clause $P(X)$ (Rule (7)) or with a clause (8) (Rule (3)). In both cases, we get a clause $H \Leftarrow \mathcal{B}$ or $H \Leftarrow \mathcal{B}, B_1(t_1), \ldots, B(t_k)$, of the right form except that it may ε-split. The largest number of blocks of the resolvent is reached when t is of the form $f(\overline{X}^k)$, $X_1, \ldots, X_k$ are distinct variables occurring only in atoms of the form $P_{ij}(X_i)$ in the body, thus creating $k \leqslant a$ splitting atoms $q_{B'_i}$, $1 \leqslant i \leqslant k$, in one of the splitters, plus k defining clauses, so that the constraint $m \leqslant a$ is satisfied.

In the remaining cases, the main premise is of the form $H \Leftarrow \mathcal{B}$, and we have two subcases. Either H is deep and no atom of $\mathcal{B}$ is selected: then, since $H \Leftarrow \mathcal{B}$ cannot ε-split, all atoms of $\mathcal{B}$ are of the form $P(X)$ with X occurring in H, so none can be maximal: contradiction. Or H is not deep and all atoms of $\mathcal{B}$, none of which are deep or of the form q_B, are selected. Moreover, $\mathcal{B}$ is not empty. So the main premise is of the form $H \Leftarrow B_1(X_1), \ldots, B_n(X_n)$, where H is of one of the forms $\perp$, q_B, or $Q(X)$. By assumption this is not ε-splittable, so $n = 1$ (and $X_n = X$ if the head is $Q(X)$). That is, the main premise is of the form $H \Leftarrow B_1(X)$ where $B_1(X)$ is a non-empty block, and H is $\perp$, q_B or $Q(X)$ (with the same X). The side premises are universal clauses of the form $P_j(X)$ with $P \in B_1$, or alternating automaton clauses $P_j(f(\overline{X}^k)) \Leftarrow B_{j1}(X_1), \ldots, B_{jk}(X_k)$ with possibly different $P_j \in B_1$, but with the same f. Index them so that $1 \leqslant j \leqslant \ell$ in the latter and $\ell + 1 \leqslant j \leqslant m$ in the former: this is Rule (4) or Rule (5) in case $\ell \geqslant 1$, Rule (7) if $\ell = 0$. $\square$

This allows us to restate Nielson et al.'s [6] main theorem, in a form that makes the role of the maximal arity a more apparent:

Theorem 6. *Let ${}_\flat\mathcal{H}_1^a$, resp. $\mathcal{H}_1^a$ be the restriction of ${}_\flat\mathcal{H}_1$, resp. $\mathcal{H}_1$ to clause sets with function symbols of arity at most a. Satisfiability of ${}_\flat\mathcal{H}_1$, resp. $\mathcal{H}_1$ clause sets, as well as ${}_\flat\mathcal{H}_1^a$, resp. $\mathcal{H}_1^a$ clause sets for any fixed $a \geqslant 1$, is* DEXPTIME*-complete.*

Proof. We may assume without loss of generality that all clauses in the initial set are ε-split. Let n_p be the number of unary predicate symbols, n_f be the number of function symbols. So there are at most 2^{n_p} splitting atoms q_B. Since $m \leqslant a$ in ${}_\flat\mathcal{H}_1^{a,*}$ clauses, there are at most

$$\sum_{m=0}^{a} \binom{2^{n_p}}{m} \leqslant \sum_{m=0}^{a} \frac{2^{n_p m}}{m!} \leqslant \sum_{m=0}^{a} \frac{2^{n_p a}}{m!} < e2^{n_p a}$$

subsets $q_{B_1}, \ldots, q_{B_m}$. Look at the rules in Fig. 1. They produce either alternating automaton clauses (Rule (5)), of which there are at most $\underbrace{n_p n_f}_{\text{head}} \underbrace{2^{n_p a}}_{\text{body}}$; or clauses containing only subterms of terms occurring in the right premise (this is obvious for all rules except Rule (4); for the latter, remember that the conclusion ε-splits as a collection of one-variable clauses, and we may rename this variable as the variable X of the right premise). By induction on the length of the derivation, all derived clauses that are not alternating automaton clauses only contain subterms of the initial set of clauses. Let N be the number of such subterms, so we can derive at most

$$\underbrace{(1 + 2^{n_p} + n_p + n_p n_f)}_{\text{head } H} \underbrace{2^{n_p N}}_{P_i(t_i)} \underbrace{e2^{n_p a}}_{q_{B_j}}$$

non-alternating automata clauses. Hence resolution with eager ε-splitting only generates exponentially many clauses.

Conversely, emptiness of alternating two-way (word) automata S is DEXPTIME-complete [2, Corollary 4]; S is an ${}_\flat\mathcal{H}_1^1$ clause set, and we can test whether $L_P(S) = \emptyset$ by adding $\perp \Leftarrow P(X)$ and checking for satisfiability. $\square$

In contrast to Theorem 6, if $a = 0$, ${}_\flat\mathcal{H}_1^a$ and $\mathcal{H}_1^a$ can be decided in polynomial time, by instantiating all clauses over the finite Herbrand base.

An easy application is *Tison's Theorem* [5, Theorem 6]: given a regular tree language $L_P(S)$, it is decidable in exponential time whether $L_P(S)$ (where S is a set of automaton clauses) contains an instance of some given first-order term s—even when s is *not* linear. Indeed, this is equivalent to checking whether S plus the clause $\perp \Leftarrow P(s)$ is unsatisfiable. The results above establish that this is still decidable in exponential time, and no more, when S is presented as an alternating automaton, or even as a set of ${}_\flat\mathcal{H}_1$ or $\mathcal{H}_1$ clauses. Also, our technique does not require determinizing any automaton, which makes it potentially more efficient than the algorithm of Middeldorp [5]. Let us push the argument further:

Theorem 7. *It is decidable in exponential time whether, given n terms $t_1, \ldots, t_n$, possibly sharing free variables, and n tree languages $L_{P_1}(S_1), \ldots, L_{P_n}(S_n)$, whether there is a substitution σ such that $t_i\sigma$ is ground and in $L_{P_i}(S_i)$ for all i, $1 \leqslant i \leqslant n$. This holds whether tree languages are presented by tree automata, alternating automata, or satisfiable ${}_\flat\mathcal{H}_1$ or $\mathcal{H}_1$ clause sets.*

Indeed, without loss of generality, $S_1, \ldots, S_n$ are built using disjoint sets of predicates. Let S be $\bigcup_{i=1}^n S_i$. S plus the clause $\bot \Leftarrow P_1(t_1), \ldots, P_n(t_n)$ is unsatisfiable iff there is a grounding substitution σ such that $t_1\sigma \in L_{P_1}(S), \ldots, t_n\sigma \in L_{P_n}(S)$.

Compiling $_\flat\mathcal{H}_1$ to alternating automata, and to automata. Call sets of universal clauses and alternating automaton clauses *alternating automata*. This coincides with usual alternating tree automata, once final states are chosen, and provided universal clauses are replaced by the usual clauses defining universal languages.

Starting from an $_\flat\mathcal{H}_1$ clause set S_0, resolution with ε-splitting ends with a so-called *saturated* set S of clauses. It is well known that (at least without splitting) we can extract a model from the latter by taking all clauses from which no atom is selected. Let S^- be the set of universal and alternating automaton clauses in S (we do not keep unit clauses q_B). We can in fact prove directly that S^- and S have the same least Herbrand model. Take any ground atom $P(t)$. $P(t)$ is in the least Herbrand model of S iff S plus $\bot \Leftarrow P(t)$ is unsatisfiable. But since S is saturated, any derivation of $\bot$ from S plus $\bot \Leftarrow P(t)$ by resolution and eager ε-splitting must first resolve $\bot \Leftarrow P(t)$ with some clause with no selected literal from S, i.e., with some clause from S^-. The resolvent must then be of the form $\bot \Leftarrow P_1(t_1), \ldots, P_n(t_n)$ (a *negative ground* clause). An easy induction on the length of the refutation shows that any refutation from S plus a set of negative ground clauses can only resolve between the latter and clauses from S^-. Since no clause from $S \setminus S^-$ is ever used, $P(t)$ is in the least Herbrand model of S^-. Conversely, the least Herbrand model of S^- is trivially included in that of S. So:

Proposition 8. *Any satisfiable $_\flat\mathcal{H}_1$ clause set S_0 can be converted in exponential time to an equivalent alternating automaton S^- recognizing the same languages*: $L_P(S^-) = L_P(S_0)$ *for every predicate P.*

Alternating automata can now be converted to (non-deterministic) tree automata, using a variant of *non-ground splitting*. Our version of this rule reads as follows. For each ε-block $B(X)$ containing at least two atoms, create a fresh predicate symbol $[B]$—call such predicates the *intersection predicates*. Then we may replace any clause $H \Leftarrow \mathcal{B}, B(X)$ of which $B(X)$ is a block of at least two atoms, by the two clauses $H \Leftarrow \mathcal{B}, [B](X)$ and $[B](X) \Leftarrow B(X)$. Generalize this rule: replace any clause $H \Leftarrow \mathcal{B}, B(X), [B_1](X), \ldots, [B_n](X)$, where $(B, B_1, \ldots, B_n)$ contains no intersection predicate and at least two distinct atoms, and X does not occur in $H \Leftarrow \mathcal{B}$, by the two clauses $H \Leftarrow \mathcal{B}, [B, B_1, \ldots, B_n](X)$ and $[B, B_1, \ldots, B_n](X) \Leftarrow B(X), B_1(X), \ldots, B_n(X)$. Completeness is preserved if we apply this non-ground splitting rule at any time, using standard redundancy arguments [1, Section 4.2.2], provided that $\succ$ is extended so that $P(s) \succ [B](t)$ for any non-intersection predicate P, and $[B](s) \succ [B'](t)$ whenever $B(s)$ properly subsumes $B'(t)$. The arguments of Theorem 6 apply to resolution with eager ε-splitting and this form of non-ground splitting. Given a saturated set S, the set S^- then consists of universal clauses $P(X)$ and clauses of the form $P(f(\overline{X}^k)) \Leftarrow P_{i_1}(X_{i_1}), \ldots, P_{i_\ell}(X_{i_\ell})$ $(1 \leqslant i_1 < \cdots < i_\ell \leqslant k)$, which can clearly be transformed into equivalent automaton clauses in polynomial time. So:

Proposition 9. *Any satisfiable $_\flat\mathcal{H}_1$ clause set can be converted in exponential time to an equivalent tree automaton recognizing the same languages. In particular, all languages $L_P(S)$, S a set of $_\flat\mathcal{H}_1$, resp. $\mathcal{H}_1$ clauses, are effectively regular.*

The $_\flat\mathcal{H}_2^{i,a}$ subclass. For any $i, a \in \mathbb{N}$, let $_\flat\mathcal{H}_2^{i,a}$ consist of those $_\flat\mathcal{H}_1^a$ clauses such that any variable occurring in the head occurs at most once in the body, and any other variable occurs at most i times in the body. These invariants are preserved by resolution and eager ε-splitting. There are now at most $n_p n_f (n_p + 1)^a$ alternating automaton $_\flat\mathcal{H}_2^{i,a}$ clauses (and they are all just automaton clauses). Assume without loss of generality that $i \leqslant n_p$. Since we only need splitting atoms q_B with $|B| \leqslant i$, we need at most $\sum_{m=1}^{i} \binom{n_p}{m} \leqslant en_p^i$ of them. In particular, at most en_p^i defining clauses are generated, each of size $O(n_p^i)$. Finally, all cases of resolution of Fig. 1 now produce smaller and smaller clauses (for Rule (3), this is because $B_1, \ldots, B_k$ contain at most one atom each by definition of $_\flat\mathcal{H}_2^{i,a}$), except for Rule (5) which, as we have seen, generates an alternating automaton clause, and Rule (4), which only generates clauses $H \Leftarrow B_1(X)$ with H of the form $\bot$

or q_B and $|B_1| \leqslant i$, hence can only generate at most 2^i consecutive clauses of the same size; ε-splitting produces defining clauses, which are only polynomially many anyway, plus a smaller clause (counting the size of q_B as being equal to that of $B(X)$). Our version of non-ground splitting never applies on clauses that are already ε-split. So:

Proposition 10. *Let $i, a \in \mathbb{N}$ be fixed. Satisfiability and conversion of ${}_\flat\mathcal{H}_2^{i,a}$ clause sets to tree automata can be done in polynomial time.*

This is roughly Theorem 2 of Nielson et al. [6]. Fixing i and a is important: the complexity of ${}_\flat\mathcal{H}_2^{i,a}$ is *exponential*, not polynomial, in i and a. In fact, satisfiability of ${}_\flat\mathcal{H}_2^{a} = \bigcup_{i\in\mathbb{N}} {}_\flat\mathcal{H}_2^{i,a}$ clause sets is DEXPTIME-complete for any $a \geqslant 2$, since it easily encodes the emptiness problem for intersections of tree automata [8], using a goal $\perp \Leftarrow P_1(X), \ldots, P_i(X)$ to test whether $L_{P_1}(S) \cap \cdots \cap L_{P_i}(S) = \emptyset$.

Clearly, ${}_\flat\mathcal{H}_2 = \bigcup_{a\geqslant 0} {}_\flat\mathcal{H}_2^{a}$ is a subclass of $\mathcal{H}_2$, the subclass of $\mathcal{H}_1$ where $(*)$ every variable in the head occurs at most once in the body [6]. Conversely, every $\mathcal{H}_2$ clause set converts in polynomial time, using (1) to an ${}_\flat\mathcal{H}_2$ clause set, by Proposition 4 and noticing that Rule (1) preserves $(*)$. $\mathcal{H}_2$ and ${}_\flat\mathcal{H}_2$ are therefore equivalent up to polynomial time reductions.

Undecidability. May we add equality tests to ${}_\flat\mathcal{H}_1$ clause sets, much as with tree automata with equality tests between brothers? In principle, this is easy: drop the requirement that $X_1, \ldots, X_k$ be distinct in Definition 1, to obtain the new class ${}_\flat\mathcal{H}_1^{=}$. E.g., $P(f(X, X)) \Leftarrow Q(X)$ states that the terms recognized at P are of the form $f(s,t)$ with $s = t$, and t recognized at Q.

Theorem 11. *It is undecidable whether sets of ${}_\flat\mathcal{H}_1^{=}$ clauses are satisfiable.*

Proof. Encode Post's correspondence problem on two letters a, b. Given any word w, and any term t, define the term $w(t)$ by: $\varepsilon(t) = t$, $aw(t) = a(w(t))$, $bw(t) = b(w(t))$. Write the following clauses. First, $Q_c(c)$; $Q(a(X)) \Leftarrow Q_c(X)$; $Q(b(X)) \Leftarrow Q_c(X)$; $Q(a(X)) \Leftarrow Q(X)$; $Q(b(X)) \Leftarrow Q(X)$; so that Q recognizes all terms $w(c)$, $w \in \{a,b\}^+$. Then the only non-${}_\flat\mathcal{H}_1$ clause $P(f(X, X)) \Leftarrow Q(X)$; P recognizes all pairs of equal non-empty words. Then, $P'(X) \Leftarrow P(X)$, and for each Post pair (u, v), $P'(f(X, Y)) \Leftarrow P'(f(u(X), v(Y)))$. Post's correspondence problem has a solution iff these clauses, plus $\perp \Leftarrow P'(f(c, c))$, form an unsatisfiable set. Note that we need only one non-${}_\flat\mathcal{H}_1$ clause, and all others are in ${}_\flat\mathcal{H}_2^{0,2}$. □

Note added to the final version. At the same time that this paper was accepted, H. Seidl and I discovered that some of the results of this paper were already present in [9]. There, ${}_\flat\mathcal{H}_1$ clause sets are called "monadic Horn theories where all positive literals are linear and shallow". That ${}_\flat\mathcal{H}_1$ can be decided by sort resolution is the topic of [9, Lemma 4]. No complexity bound is given; the procedure of op.cit. runs in double exponential time, and is therefore not optimal. That the problem mentioned in Theorem 7 is decidable was also proved, in [9, Lemma 3], again without complexity bound. Finally, [9] gives another proof that ${}_\flat\mathcal{H}_1^{=}$ is undecidable (end of Section 3), by reduction from unifiability in arbitrary sort theories.

Acknowledgements

Thanks to H. Comon-Lundh, who urged me to publish these results, to H. Seidl for spotting a bug in a previous version of Proposition 4, to F. Jacquemard, who carefully reread a first version of this paper, and to the anonymous referees for pointing out some important bugs and inaccuracies.

References

[1] L. Bachmair, H. Ganzinger, Resolution theorem proving, in: J.A. Robinson, A. Voronkov (Eds.), Handbook of Automated Reasoning, vol. I, North-Holland, Amsterdam, 2001, pp. 19–99, Chapter 2.

[2] W. Charatonik, A. Podelski, J.-M. Talbot, Paths vs. trees in set-based program analysis, in: Proc. 27th Annual ACM Symp. Principles of Programming Languages, ACM Press, New York, 2000, pp. 330–338.

[3] T. Frühwirth, E. Shapiro, M.Y. Vardi, E. Yardeni, Logic programs as types for logic programs, in: Proc. 6th Symp. on Logic in Computer Science, IEEE Computer Society Press, New York, 1991, pp. 300–309.

[4] J. Goubault-Larrecq, M. Roger, K.N. Verma, Abstraction and resolution modulo AC: How to verify Diffie–Hellman-like protocols automatically, J. Logic Algebraic Programming (2004), submitted for publication.

[5] A. Middeldorp, Approximating dependency graphs using tree automata techniques, in: F. Massacci, R. Goré (Eds.), Proc. 1st Internat. Joint Conf. Automated Reasoning, in: Lecture Notes in Artif. Intell., vol. 2083, Springer-Verlag, Berlin, 2001, pp. 593–610.
[6] F. Nielson, H.R. Nielson, H. Seidl, Normalizable Horn clauses, strongly recognizable relations and Spi, in: Proc. 9th Static Analysis Symposium, in: Lecture Notes in Comput. Sci., vol. 2477, Springer-Verlag, Berlin, 2002, pp. 20–35.
[7] A. Riazanov, A. Voronkov, Splitting without backtracking, in: B. Nebel (Ed.), Proc. 17th Internat. Joint Conf. Artif. Intell., vol. 1, Morgan Kaufmann, Los Altos, CA, 2001, pp. 611–617.
[8] H. Seidl, Haskell overloading is DEXPTIME-complete, Inform. Process. Lett. 52 (2) (1994) 57–60.
[9] C. Weidenbach, Towards an automatic analysis of security protocols in first-order logic, in: Proc. 16th Internat. Conf. Automated Deduction, in: Lecture Notes in Artif. Intell., vol. 1632, Springer-Verlag, Berlin, 1999, pp. 314–328.