



**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**

**Technische Universität Dresden  
Institute for Theoretical Computer Science  
Chair for Automata Theory**

## **LTCS-Report**

# **Finite Herbrand Models for Restricted First-Order Clauses**

Stefan Borgwardt and Barbara Morawska

LTCS-Report 19-07

Postal Address:  
Lehrstuhl für Automatentheorie  
Institut für Theoretische Informatik  
TU Dresden  
01062 Dresden

<http://lat.inf.tu-dresden.de>

Visiting Address:  
Nöthnitzer Str. 46  
Dresden

## Abstract

We call a Herbrand model of a set of first-order clauses *finite*, if each of the predicates in the clauses is interpreted by a finite set of ground terms. We consider first-order clauses with the signature restricted to unary predicate and function symbols and one variable. Deciding the existence of a finite Herbrand model for a set of such clauses is known to be EXPTIME-hard even when clauses are restricted to an anti-Horn form. Here we present an EXPTIME algorithm to decide if a finite Herbrand model exists in the more general case of arbitrary clauses. Moreover, we describe a way to generate finite Herbrand models, if they exist. Since there can be infinitely many minimal finite Herbrand models, we propose a new notion of acyclic Herbrand models. If there is a finite Herbrand model for a set of restricted clauses, then there are finitely many (at most triple-exponentially many) acyclic Herbrand models. We show how to generate all of them.

## 1 Introduction

We consider the problem of existence of the so called *finite Herbrand model* for a set of restricted first-order clauses. This is a special case of the problem of satisfiability a set of the first order formulas in FOL which continues to be a great area of research in logic, computer science, automated deduction. The problem is in general undecidable, not even semi-decidable. Hence the focus is on finding models for restricted classes of clauses. Our restrictions consists of allowing only unary predicates, unary function symbols, one constant and only one variable.<sup>1</sup>

Such sets of clauses can be seen as skolemized versions of Ackermann formulas, for which satisfiability is known to be decidable [8; 11]. If a set of such clauses is satisfiable, then it has a Herbrand model. This model may not be finite, i.e. the predicates may be interpreted by infinite sets of the first order elements from Herbrand universe.

We search for the finite Herbrand models, i.e. the models in which each predicate is interpreted by a finite set of terms. In other words, we search for a special kind of ground atomic representation of a Herbrand model in the terminology of [5].<sup>2</sup> Hyperresolution-based methods have been proposed, cf. [4; 9], but its precise complexity is still unclear. We prove that the problem of deciding whether a finite Herbrand model of a given set of first-order clauses exists, is EXPTIME-complete, provided that the signature is restricted as mentioned above.

This restricted class of clauses came to our focus, while looking for methods

---

<sup>1</sup>One constant restriction is not essential.

<sup>2</sup>Ground atomic representations are not finite, because Herbrand universe is not finite. Our notion of finite Herbrand model is not equivalent to a ground atomic representation as such.

of solving certain language equations or unification for description logics [1]. We have found that these problems may be reduced to finding finite Herbrand models for a set of anti-Horn clauses restricted in this way, [2]. As a continuation of that research, we show a generalization of the methods used there to the set of *all* clauses under the same restrictions. It is interesting that this more general problem is in the same complexity class as in the case of anti-Horn clauses. In future one can attempt to further generalize this class by lifting some of the restrictions.

Our solution to the problem is given by an algorithm that is not based on resolution techniques nor is an enumeration procedure explicitly constructing a model in a bottom-up fashion. Instead it follows a top-down approach, starting from the largest terms that appear in the model. However, we also demonstrate how to use our approach to actually construct models, and discuss how to find all so-called *acyclic* Herbrand models of a given set of clauses.

Our methods may be relevant to problems like that of (i) *termination* in logic programming with function symbols [10] and (ii) *acyclicity* for existential rules [7]. While in (i) also Herbrand models are of chief interest, the setting (ii) usually considers arbitrary first-order structures, but the so-called *Skolem chase*, related to Herbrand models, is sometimes investigated in the context of finding approximating criteria for termination [13]. However, there are two main differences to our setting. First, termination requires that *all* (minimal) models must be finite, instead of just one (as in our case). Only in the case of Horn clauses, our algorithm is a decision procedure for termination, because in this case there is a unique minimal Herbrand model. Second, usually predicates and functions of arbitrary arity are allowed, whereas we restrict ourselves to the unary case.

The problem of finding *one* finite Herbrand model is related to finite model finding [12], and useful for explaining to a user why a consequence does not hold since such a finite Herbrand model is relatively easy to understand due to its tree-like structure. Our construction can also be used for finding one terminating computation of a disjunctive logic program, which cannot be guaranteed to terminate for all possible computation paths [6].

In the next section, we introduce propagation rules as a class of first-order clauses and discuss the problem of finding finite Herbrand models. In Section 3, we introduce a characterization of such models via so-called *shortcuts*, and use them in Section 4 to decide the existence of finite Herbrand models. Finally, in Section 5 we discuss how to actually construct such models. Detailed proofs can be found in the accompanying technical report [3].

## 2 Clauses and Herbrand models

Let  $\mathbf{P}$  be a set of unary predicate symbols,  $\mathbf{F}$  be a set of unary function symbols,  $a$  be a constant symbol, and  $x$  be a variable. We consider first-order clauses  $\mathfrak{c}$  of the form

$$P_1(t_1) \wedge \cdots \wedge P_k(t_k) \rightarrow Q_1(s_1) \vee \cdots \vee Q_\ell(s_\ell),$$

where  $P_1, \dots, P_k, Q_1, \dots, Q_\ell \in \mathbf{P}$  and the terms  $t_1, \dots, t_k, s_1, \dots, s_\ell$  are constructed over  $\mathbf{F}$ ,  $a$ , and  $x$ . The single variable  $x$  is considered to be universally quantified. Since all function symbols are unary, we usually denote such terms without parentheses, e.g. as  $fga$  for  $f, g \in \mathbf{F}$ . The *height*  $|t|$  of a term  $t$  is defined as the number of occurrences of function symbols it contains. The left-hand side of  $\mathfrak{c}$  is called its *body*, and the right-hand side is the *head*. The set of all predicates in the body of  $\mathfrak{c}$  is denoted by  $\mathbf{b}(\mathfrak{c})$ , and  $\mathbf{h}(\mathfrak{c})$  is the set of all predicates in the head of  $\mathfrak{c}$ . Note that the body may be empty, in which case we denote it by  $\top$ , and similarly write  $\perp$  for an empty head.

We call such clauses (*extended*) *propagation rules* [2]. In the following, we consider a set  $\mathcal{C}$  of such clauses, and are looking for a finite Herbrand model  $\mathcal{H}$  of  $\mathcal{C}$ , i.e. essentially a finite subset of the Herbrand universe over  $\mathbf{P}$ ,  $\mathbf{F}$  and  $a$ . By “finite” we mean that each predicate symbol is interpreted by a finite set of ground terms, i.e. objects of the Herbrand universe. We will use two equivalent representations for Herbrand interpretations  $\mathcal{H}$ , depending on which is more convenient for our arguments: either  $\mathcal{H}$  specifies sets of ground terms  $P^{\mathcal{H}}$  for each  $P \in \mathbf{P}$ , or alternatively it can be seen as a single set of ground atoms of the form  $P(t)$ , specifying exactly which atoms are true in  $\mathcal{H}$ .

### 2.1 Normalization

Before we check for the existence of a finite Herbrand model, we first normalize the set  $\mathcal{C}$ .

**Definition 1.** A set of propagation rules is in *normal form* iff it consists of only

- one *positive ground clause* of the form  $\top \rightarrow A(a)$  such that  $A$  does not occur in the head of any other clause,
- *flat* clauses of the form  $P_1(x) \wedge \cdots \wedge P_k(x) \rightarrow Q_1(x) \vee \cdots \vee Q_\ell(x)$  with  $k \geq 1$ , and
- *increasing* and *decreasing* clauses of the forms

$$Q^f(x) \rightarrow Q(fx) \quad \text{and} \quad Q(fx) \rightarrow Q^f(x),$$

respectively, where  $Q^f$  is a unique predicate, i.e. for every  $Q \in \mathbf{P}$  and  $f \in \mathbf{F}$ , there can be at most one  $Q^f \in \mathbf{P}$  for which the above two clauses exist (and then they must both exist).

Although technically  $\top \rightarrow A(a)$  can also be seen as a flat clause (as it involves only a single term of a fixed height, instead of multiple terms with different heights), we do not consider it part of the flat clauses, due to its special status as the only ground clause.

In order to transform  $\mathcal{C}$  into normal form, we first identify possibly existing pairs of increasing and decreasing clauses, then flatten the remaining clauses by introducing more increasing and decreasing clauses. Afterwards, we have to eliminate the clauses that are *mixed*, i.e. contain both  $a$  and  $x$ . We do this by guessing for each predicate  $P$ , whether  $P(a)$  will be true or not, and replacing  $P(a)$  in all clauses by  $\top$  or  $\perp$ , respectively. In this process, we also eliminate all ground clauses and replace them with the single ground clause  $\top \rightarrow A(a)$ .

In more detail, we do the following.

**N1** For any  $Q \in \mathbf{P}$  and  $f \in \mathbf{F}$ , if there is a unique  $P \in \mathbf{P}$  such that  $\mathcal{C}$  contains the clauses  $P(x) \rightarrow Q(fx)$  and  $Q(fx) \rightarrow P(x)$ , then rename  $P$  to  $Q^f$  (which we assume to be fresh). Initially, these are the only increasing and decreasing clauses.

**N2** Do the following until all clauses are either increasing, decreasing, or flat (but possibly mixed):

For any non-flat clause  $\mathfrak{c} \in \mathcal{C}$  that has not been identified as increasing or decreasing, choose an atom  $P(ft)$  in  $\mathfrak{c}$  and replace it by  $P^f(t)$ . Add the increasing and decreasing clauses  $P^f(x) \rightarrow P(fx)$  and  $P(fx) \rightarrow P^f(x)$  to  $\mathcal{C}$  if they do not already exist.

**N3** Introduce a fresh predicate  $A$  and the clause  $\top \rightarrow A(a)$ .

Guess a set  $\mathbf{P}_a \subseteq \mathbf{P} \setminus \{A\}$ . For every  $P \in \mathbf{P}$ , do the following:

- If  $P \in \mathbf{P}_a$ , then add the flat clause  $A(x) \rightarrow P(x)$  to  $\mathcal{C}$  and replace  $P(a)$  with  $\top$  in all clauses of  $\mathcal{C}$ .
- If  $P \notin \mathbf{P}_a$ , then add the flat clause  $A(x) \wedge P(x) \rightarrow \perp$  to  $\mathcal{C}$ , and replace  $P(a)$  with  $\perp$  in all clauses of  $\mathcal{C}$ .

**N4** Remove all clauses from  $\mathcal{C}$  that contain  $\perp$  in the body or  $\top$  in the head.

Remove  $\top$  from the body of all clauses, and  $\perp$  from the head of all clause.

**N5** If there remains any clause except  $\top \rightarrow A(a)$  that has an empty body ( $\top$ ), then the normalization fails.

**Example 2.** Consider the clause  $P_1(x) \wedge P_2(ga) \rightarrow P_2(x) \vee P_1(fa)$ .

- Step **N1** does not apply, because there are no clauses that we could identify as increasing or decreasing.

- In Step **N2**, we replace the atom  $P_2(ga)$  by an atom  $P_2^g(a)$  and the atom  $P_1(f(a))$  by  $P_1^f(a)$ , where  $P_2^g$  and  $P_1^f$  are new predicates. We add the corresponding increasing and decreasing clauses, and now the set of clauses has the following form:

$$\frac{\text{mixed clause: } P_1(x) \wedge P_2^g(a) \rightarrow P_2(x) \vee P_1^f(a),}{\text{increasing/decreasing } P_2^g(x) \rightarrow P_2(g(x)),}$$

$$\text{clauses: } P_2(g(x)) \rightarrow P_2^g(x),$$

$$P_1^f(x) \rightarrow P_1(f(x)),$$

$$P_1(f(x)) \rightarrow P_1^f(x).$$

- In Step **N3**, we add  $\top \rightarrow A(a)$ . Then we have to guess  $\mathbf{P}_a \subseteq \mathbf{P}$ . Let us guess that  $P_2^g \in \mathbf{P}_a$  and  $P_1, P_2, P_1^f \notin \mathbf{P}_a$ .  $P_2^g(a)$  will be replaced by  $\top$  and  $P_1^f(a)$  will be replaced by  $\perp$ . We add the clause  $A(x) \rightarrow P_2^g(x)$  and corresponding clauses for the other predicates. The set of clauses is now of the following form:

$$\frac{\text{start clause: } \top \rightarrow A(a),}{\text{flat clauses: } A(x) \rightarrow P_2^g(x),}$$

$$A(x) \wedge P_1(x) \rightarrow \perp,$$

$$A(x) \wedge P_2(x) \rightarrow \perp,$$

$$A(x) \wedge P_1^f(x) \rightarrow \perp,$$

$$P_1(x) \wedge \top \rightarrow P_2(x) \vee \perp,$$

$$\frac{\text{increasing/decreasing } P_2^g(x) \rightarrow P_2(g(x)),}{\text{clauses: } P_2(g(x)) \rightarrow P_2^g(x),}$$

$$P_1^f(x) \rightarrow P_1(f(x)),$$

$$P_1(f(x)) \rightarrow P_1^f(x).$$

- In Step **N4**, we remove  $\top$  and  $\perp$  from one flat clause and obtain the normalized set of clauses:

$$\frac{\text{start clause: } \top \rightarrow A(a),}{\text{flat clauses: } A(x) \rightarrow P_2^g(x),}$$

$$A(x) \wedge P_1(x) \rightarrow \perp,$$

$$A(x) \wedge P_2(x) \rightarrow \perp,$$

$$A(x) \wedge P_1^f(x) \rightarrow \perp,$$

$$P_1(x) \rightarrow P_2(x),$$

$$\frac{\text{increasing/decreasing } P_2^g(x) \leftrightarrow P_2(g(x)),}{\text{clauses: } P_1^f(x) \leftrightarrow P_1(f(x)).}$$

- Since there are no flat clauses with  $\top$  on the left, Step **N5** does not apply, hence the normalization succeeds.

Obviously, a successful normalization process will terminate with a set of propagation rules in normal form. It is correct in the following sense.

**Lemma 3.** *The set  $\mathcal{C}$  has a finite Herbrand model iff there is a set  $\mathbf{P}_a$  for which the normalization process produces a set  $\mathcal{C}'$  that has a finite Herbrand model.*

*Proof.* ( $\Rightarrow$ ): Let  $\mathcal{H}$  be a finite Herbrand model for  $\mathcal{C}$ . We show how to construct a set  $\mathbf{P}_a$  and a finite Herbrand model  $\mathcal{H}'$  for  $\mathcal{C}'$ , by following the normalization steps.

- Initially, we set  $\mathcal{H}' := \mathcal{H}$ , which is a model of the initial set of clauses.
- In Step **N1**, we simply replace some predicates by fresh predicates. If  $P$  is renamed to  $Q^f$ , we set  $(Q^f)^{\mathcal{H}'} := P^{\mathcal{H}'}$ , which preserves the property that  $\mathcal{H}'$  is a Herbrand model of the resulting set of clauses.
- Step **N2** consists of flattening the clauses and creating new predicates on the way. Each new predicate is of the form  $P^f$  for some already existing predicate  $P$  and a function symbol  $f$ . We define the interpretation of these new predicates recursively: if  $P^{\mathcal{H}'}$  is defined, we set  $(P^f)^{\mathcal{H}'} := \{t \mid ft \in P^{\mathcal{H}'}\}$ . Under this modification, it is clear that the atoms  $P(ft)$  and  $P^f(t)$  are equivalent in  $\mathcal{H}'$ , and hence  $\mathcal{H}'$  remains a model of the flattened clauses, and also satisfies the new increasing and decreasing clauses.
- In Step **N3** we define a special predicate  $A$  and we guess  $\mathbf{P}_a \subseteq \mathbf{P} \setminus \{A\}$ . We define  $A^{\mathcal{H}'} := \{a\}$ , which satisfies the new clause  $\top \rightarrow A(a)$ . We can further guess  $\mathbf{P}_a := \{P \in \mathbf{P} \setminus \{A\} \mid a \in P^{\mathcal{H}'}\}$ . Thus, for every  $P \in \mathbf{P}_a$ ,  $A(x) \rightarrow P(x)$  is satisfied by  $\mathcal{H}'$ , and for every  $P \notin \mathbf{P}_a$  except  $A$ ,  $A(x) \wedge P(x) \rightarrow \perp$  is satisfied.
- The transformations in Step **N4** clearly do not affect the satisfaction of the remaining clauses.
- The failure condition in Step **N5** cannot apply, because any remaining non-ground clause with  $\top$  on the left-hand side would mean that  $\mathcal{H}'$  is infinite, which implies by the above constructions that  $\mathcal{H}$  is infinite.

( $\Leftarrow$ ): Assume that the normalization process did not fail, so we have guessed a set  $\mathbf{P}_a$  to obtain  $\mathcal{C}'$ , and that  $\mathcal{H}$  is a finite Herbrand model of  $\mathcal{C}'$ . We show that  $\mathcal{H}$  is also a model of  $\mathcal{C}$  if we extend it with interpretations for the predicates that were renamed in the first step, i.e. if  $Q^f$  is the renamed version of  $P$  in  $\mathcal{C}'$ , then we extend  $\mathcal{H}$  by defining  $P^{\mathcal{H}} := (Q^f)^{\mathcal{H}}$ . To show this, we demonstrate that  $\mathcal{H}$  satisfies the set of clauses as it was before each normalization step, by going through the steps in reverse order.

- Step **N5** does not apply due to our assumption that the normalization did not fail.

- Regarding Step **N4**, the removed clauses are trivially satisfied by any Herbrand interpretation, and the addition of  $\top$  to the body or  $\perp$  to the head does not affect the satisfaction of a clause.
- For Step **N3**, observe that for any  $P \in \mathbf{P}_a$  we have  $a \in P^{\mathcal{H}}$  since  $\mathcal{H}$  satisfies  $\top \rightarrow A(a)$  and  $A(x) \rightarrow P(x)$ , i.e.  $P(a)$  is equivalent to  $\top$  in  $\mathcal{H}$ . Similarly, for  $P \notin \mathbf{P}_a$ , the clause  $A(x) \wedge P(x) \rightarrow \perp$  ensures that  $a \notin P^{\mathcal{H}}$ .
- In Step **N2** the new increasing and decreasing clauses ensure that the replaced atoms are equivalent to the original ones under  $\mathcal{H}$ .
- Finally, the original clauses before Step **N1** are satisfied by  $\mathcal{H}$  due to our initial extension of  $\mathcal{H}$  to the renamed predicates.  $\square$

In the following, we always assume  $\mathcal{C}$  to be normalized.

## 2.2 Finding Finite Herbrand Models: A Naive Algorithm

To guide our procedure for finding a finite Herbrand model, we first consider a naive model construction approach. We start with the small Herbrand interpretation  $\mathcal{H}_0 = \{A(a)\}$  that satisfies the only ground clause in  $\mathcal{C}$ , and nondeterministically add atoms that are required to satisfy the clauses in  $\mathcal{C}$ , in order to extend the interpretation to  $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \dots$ . If this process terminates, we have found a finite Herbrand model  $\mathcal{H}_m$ .

Formally, we follow the following steps to construct each  $\mathcal{H}_{i+1}$  from  $\mathcal{H}_i$ , as long as there are still unsatisfied clauses.

**Definition 4.** Let  $\mathcal{H}_i$  be a finite Herbrand interpretation.

- Choose an arbitrary clause  $\mathbf{c}$  from  $\mathcal{C}$  and an arbitrary ground term  $t$  to substitute for the variable  $x$ , such that  $\mathcal{H}_i \not\models \mathbf{c}[x \mapsto t]$ , i.e. the clause is not yet satisfied.

If  $\mathbf{c}$  is flat or increasing, we say that the term  $t$  is *active* in  $\mathcal{H}_i$  at the clause  $\mathbf{c}$ . If  $\mathbf{c}$  is a decreasing clause of the form  $Q(fx) \rightarrow Q^f(x)$ , then we say that  $ft$  is *active* in  $\mathcal{H}_i$  at  $\mathbf{c}$ .

The set of all pairs  $(t, \mathbf{c})$  such that  $t$  is active at  $\mathbf{c}$  in  $\mathcal{H}_i$  is denoted by  $\mathbf{act}(\mathcal{H}_i)$ .

- Choose an arbitrary disjunct  $Q(t')$  from the head of  $\mathbf{c}$  (where  $t'$  is either  $x$  or  $fx$  for some  $f \in \mathbf{F}$ ) and set  $\mathcal{H}_{i+1} := \mathcal{H}_i \cup \{Q(t')[x \mapsto t]\}$  to satisfy the clause for  $t$ .

We call this the *expansion rule*, and denote by  $\mathcal{H}_i \xrightarrow{\mathbf{c}, t, Q} \mathcal{H}_{i+1}$  the application of this rule for the clause  $\mathbf{c}$ , active term  $t$  and predicate  $Q$  (if it is applicable). The



*expansion step*  $\mathcal{H}_i \rightarrow \mathcal{H}_{i+1}$  denotes that the above relation holds for some  $\mathbf{c}$ ,  $t$  and  $Q$ . Given a Herbrand interpretation  $\mathcal{H}_0$ , an *expansion sequence* (starting in  $\mathcal{H}_0$ ) is a finite sequence  $\mathcal{H}_0 \rightarrow \mathcal{H}_1 \rightarrow \dots \rightarrow \mathcal{H}_m$ . The sequence is called *terminating* if  $\mathcal{H}_m$  is a model of  $\mathcal{C}$ , in which case the expansion rule is not applicable anymore. We say that  $\mathcal{C}$  *terminates* if it has a terminating expansion sequence starting in  $\mathcal{H}_0 = \{A(a)\}$ .

Termination is obviously equivalent to  $\mathcal{C}$  having a finite Herbrand model.

**Example 5.** Let our set of propagation rules  $\mathcal{C}$  be the following:

<b>start clause:</b>	$\mathbf{c}_0 = \top \rightarrow A(a)$
<b>flat clauses:</b>	$\mathbf{c}_1 = A(x) \rightarrow P_1(x)$
	$\mathbf{c}_2 = A(x) \rightarrow P_1^f(x)$
	$\mathbf{c}_3 = A(x) \rightarrow Q_2^f(x)$
	$\mathbf{c}_4 = A(x) \wedge Q_1(x) \rightarrow \perp$
	$\mathbf{c}_5 = A(x) \wedge Q_2(x) \rightarrow \perp$
	$\mathbf{c}_6 = P_1(x) \rightarrow Q_1(x) \vee Q_2^f(x)$
	$\mathbf{c}_7 = Q_1(x) \wedge Q_2(x) \rightarrow \perp$
<b>increasing/decreasing</b>	$\mathbf{c}_{8/9} = P_1^f(x) \leftrightarrow P_1(f(x))$
<b>clauses:</b>	$\mathbf{c}_{10/11} = Q_2^f(x) \leftrightarrow Q_2(f(x))$

The choice of  $P_a$  is obviously:  $\{P_1, P_1^f, Q_2^f\}$ .

We start with  $\mathcal{H}_0 = \{A(a)\}$  and construct an expansion sequence, by choosing  $\mathbf{c}_1$ ,  $a$  and  $P_1$ :  $\mathcal{H}_0 \xrightarrow{\mathbf{c}_1, a, P_1} \mathcal{H}_1$  where  $\mathcal{H}_1 = \{A(a), P_1(a)\}$ . Now we choose  $\mathbf{c}_2$  for expansion with the same term  $a$ :  $\mathcal{H}_1 \xrightarrow{\mathbf{c}_2, a, P_1^f} \mathcal{H}_2$ . Similarly, we do expansion with  $\mathbf{c}_3$ :  $\mathcal{H}_2 \xrightarrow{\mathbf{c}_3, a, Q_2^f} \mathcal{H}_3$ . At this moment clauses 4, 5 and 7 are satisfied. Since  $Q^f(a)$  is in  $\mathcal{H}_3$  clause 6 is also satisfied. This leave us with increasing clauses  $\mathbf{c}_8$  and  $\mathbf{c}_{10}$ .

$$\begin{aligned}
\mathcal{H}_0 &\xrightarrow{\mathbf{c}_1, a, P_1} \mathcal{H}_1 = \{A(a), P_1(a)\} \\
\mathcal{H}_1 &\xrightarrow{\mathbf{c}_2, a, P_1^f} \mathcal{H}_2 = \mathcal{H}_1 \cup \{P_1^f(a)\} \\
\mathcal{H}_2 &\xrightarrow{\mathbf{c}_3, a, Q_2^f} \mathcal{H}_3 = \mathcal{H}_2 \cup \{Q_2^f(a)\} \\
\mathcal{H}_3 &\xrightarrow{\mathbf{c}_8, a, P_1} \mathcal{H}_4 = \mathcal{H}_3 \cup \{P_1(fa)\} \\
\mathcal{H}_4 &\xrightarrow{\mathbf{c}_{10}, a, Q_2^f} \mathcal{H}_5 = \mathcal{H}_4 \cup \{Q_2^f(fa)\}
\end{aligned}$$

At this moment the  $\mathbf{c}_6$  is not satisfied, because  $P_1(fa)$  is in  $\mathcal{H}_5$ . We have a choice in expanding  $\mathcal{H}_5$  either adding  $Q_1(fa)$  or  $Q_2^f(fa)$  to the interpretation.

If we choose  $Q_1(fa)$  we will not get a terminating expansion sequence:

$$\mathcal{H}_5 \xrightarrow{\mathbf{c}_6, fa, Q_1} \mathcal{H}_6 = \mathcal{H}_5 \cup \{Q_1(fa)\}$$

At this moment  $\mathfrak{c}_7$  is not satisfied and we have no way to satisfy it. Hence since  $\mathcal{H}_6$  is not a model, the sequence does not terminate. Nevertheless another choice in expanding  $\mathcal{H}_5$  leads to termination.

$$\begin{aligned}
\mathcal{H}_0 &\xrightarrow{\mathfrak{c}_{1,a,P_1}} \mathcal{H}_1 = \{A(a), P_1(a)\} \\
\mathcal{H}_1 &\xrightarrow{\mathfrak{c}_{2,a,P_1^f}} \mathcal{H}_2 = \mathcal{H}_1 \cup \{P_1^f(a)\} \\
\mathcal{H}_2 &\xrightarrow{\mathfrak{c}_{3,a,Q_2^f}} \mathcal{H}_3 = \mathcal{H}_2 \cup \{Q_2^f(a)\} \\
\mathcal{H}_3 &\xrightarrow{\mathfrak{c}_{8,a,P_1}} \mathcal{H}_4 = \mathcal{H}_3 \cup \{P_1(fa)\} \\
\mathcal{H}_4 &\xrightarrow{\mathfrak{c}_{10,a,Q_2^f}} \mathcal{H}_5 = \mathcal{H}_4 \cup \{Q_2^f(fa)\} \\
\mathcal{H}_5 &\xrightarrow{\mathfrak{c}_{6,fa,Q_2^f}} \mathcal{H}_6 = \mathcal{H}_5 \cup \{Q_2^f(fa)\} \\
\mathcal{H}_6 &\xrightarrow{\mathfrak{c}_{10,fa,Q_2}} \mathcal{H}_6 = \mathcal{H}_5 \cup \{Q_2(ffa)\}
\end{aligned}$$

Here is another example.

**Example 6.** Let the set of predicates  $P_a = \{P^f, Q_2^f\}$ . Let our set of propagation rules  $\mathcal{C}$  be the following with some clauses added by the choice of  $P_a$  omitted.

<b>start clause:</b>	$\mathfrak{c}_0 = \top \rightarrow A(a)$
<b>flat clauses:</b>	$\mathfrak{c}_1 = A(x) \rightarrow P^f(x)$ $\mathfrak{c}_2 = A(x) \rightarrow Q_2^f(x)$ $\mathfrak{c}_3 = A(x) \wedge R^f(x) \rightarrow \perp$ $\mathfrak{c}_4 = P(x) \rightarrow Q_1(x) \vee Q_2(x)$ $\mathfrak{c}_5 = Q_1(x) \wedge Q_2(x) \rightarrow R(x)$
<b>increasing/decreasing</b>	$\mathfrak{c}_{6/7} = P^f(x) \leftrightarrow P(f(x))$
<b>clauses:</b>	$\mathfrak{c}_{8/9} = Q_2^f(x) \leftrightarrow Q_2(f(x))$ $\mathfrak{c}_{10/11} = R^f(x) \leftrightarrow R(f(x))$

We start with  $\mathcal{H}_0 = \{A(a)\}$  and construct an expansion sequence, by choosing  $\mathfrak{c}_1$ ,  $a$  and  $P^f$ :  $\mathcal{H}_0 \xrightarrow{\mathfrak{c}_{1,a,P^f}} \mathcal{H}_1$  where  $\mathcal{H}_1 = \{A(a), P^f(a)\}$ . Now we choose  $\mathfrak{c}_2$  for expansion with the same term  $a$ :  $\mathcal{H}_1 \xrightarrow{\mathfrak{c}_{2,a,Q_2^f}} \mathcal{H}_2$  and  $\mathcal{H}_2 = \{A(a), P^f(a), Q_2^f(a)\}$ . Now the increasing clauses are not satisfied, hence we have to create term  $f(a)$  in the interpretation of  $P$  and  $Q_2$ .

The expansion steps are in the following table.

$$\begin{aligned}
\mathcal{H}_0 &\xrightarrow{c_{1,a,P^f}} \mathcal{H}_1 = \{A(a), P^f(a)\} \\
\mathcal{H}_1 &\xrightarrow{c_{2,a,Q_2^f}} \mathcal{H}_2 = \mathcal{H}_1 \cup \{Q_2^f(a)\} \\
\mathcal{H}_2 &\xrightarrow{c_{6,a,P}} \mathcal{H}_3 = \mathcal{H}_2 \cup \{P(fa)\} \\
\mathcal{H}_3 &\xrightarrow{c_{4,fa,Q_1}} \mathcal{H}_4 = \mathcal{H}_3 \cup \{Q_1(fa)\} \\
\mathcal{H}_4 &\xrightarrow{c_{8,a,Q_2}} \mathcal{H}_5 = \mathcal{H}_4 \cup \{Q_2(fa)\} \\
\mathcal{H}_5 &\xrightarrow{c_{5,fa,R}} \mathcal{H}_6 = \mathcal{H}_5 \cup \{R(fa)\} \\
\mathcal{H}_6 &\xrightarrow{c_{11,fa,R^f}} \mathcal{H}_6 = \mathcal{H}_5 \cup \{R^f(a)\}
\end{aligned}$$

Now, because of the clause 3, which cannot be satisfied, we cannot terminate this sequence. Nevertheless there is a sequence that terminates. It differs from the above one by taking different order in satisfying the increasing and flat clauses.

$$\begin{aligned}
\mathcal{H}_0 &\xrightarrow{c_{1,a,P^f}} \mathcal{H}_1 = \{A(a), P^f(a)\} \\
\mathcal{H}_1 &\xrightarrow{c_{2,a,Q_2^f}} \mathcal{H}_2 = \mathcal{H}_1 \cup \{Q_2^f(a)\} \\
\mathcal{H}_2 &\xrightarrow{c_{6,a,P}} \mathcal{H}_3 = \mathcal{H}_2 \cup \{P(fa)\} \\
\mathcal{H}_3 &\xrightarrow{c_{8,a,Q_2}} \mathcal{H}_4 = \mathcal{H}_3 \cup \{Q_2(fa)\}
\end{aligned}$$

At this moment all clauses are satisfied. The sequence terminates.

There are two reasons for an expansion sequence starting in  $\mathcal{H}_0$  to be non-terminating:

- It may be impossible to avoid a contradiction, i.e. a clause with an active ground term and an empty head ( $\perp$ ).
- Any expansion sequence that avoids such contradictions is forced into a cycle of producing atoms with higher and higher ground terms.

The main idea of our decision procedure is based on an obvious observation that in order for a sequence to terminate, the atoms added in the expansion sequence must contain terms of a bounded height. Moreover we can *trace* expansion steps independently for each of these terms, because of the restrictions on the signature.

The simplest terminating sequence would be one which avoids applying the expansion rule to increasing clauses completely (if that is possible), i.e. that never involves terms other than  $a$ . Since there are only finitely many predicates, such a *flat* sequence must be terminating.

### 3 Taking Shortcuts

Let us consider a terminating expansion sequence  $\mathcal{H}_0 \rightarrow \mathcal{H}_1 \rightarrow \dots \rightarrow \mathcal{H}_m$ . We want to divide it into subsequences that can serve as building blocks for the whole sequence. Here is an informal description of how this is done.

As observed above, the terms that appear in the atoms are of bounded height. We first consider a term of maximal height, e.g.  $ft$ .<sup>3</sup> Hence it was *created* by an expansion rule applied to an increasing clause with a predicate  $P^f$ . Let  $\mathcal{X} = \{P \mid P^f(t) \in \mathcal{H}_i\}$  for some interpretation  $\mathcal{H}_i$  in the original sequence. Now we can reconstruct an expansion sequence starting with  $\mathcal{H}'_0 = \{P(ft) \mid P \in \mathcal{X}\}$  following the way the term  $ft$  was deactivated in the clauses  $\mathcal{C}$ , in the original expansion sequence. We say that the expansion sequence for  $ft$  obtained in this way has height 0, because  $ft$  is of maximal height and no bigger term is created.

Now, consider  $t$ , the subterm of  $ft$ . Let us assume that  $t = gt'$ . Such term had to be created by an increasing clause with a predicate  $Q^g$ . We repeat the process to extract an expansion sequence for the term  $gt'$  setting  $\mathcal{X}' = \{Q \mid Q^g(t') \in \mathcal{H}_j\}$  for some interpretation  $\mathcal{H}_j$  and starting with  $\mathcal{H}''_0 = \{Q(gt') \mid Q \in \mathcal{X}'\}$ . Now, the sequence has height 1, because for the bigger terms (of maximal height), we have already extracted expansion sequences of heights 0.

We can proceed in the same way for the term  $t'$ , extracting an expansion sequence of height 2, etc., until we come to the term  $a$ . For this term the original sequence is reconstructed by starting with  $\mathcal{H}_0 = \{A(a)\}$ , and following the way  $a$  is deactivated in the original sequence, while the bigger terms are taken care of by the already extracted subsequences.

In order to capture these ideas, we define the notion of a *shortcut* that represents an extracted expansion subsequence of an assumed terminating sequence.

**Definition 7.** Let  $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathbf{P}$ . The pair  $(\mathcal{X}, \mathcal{Y})$  is called a *shortcut (for  $\mathcal{C}$ )* if there is a ground term  $t$  and an expansion sequence  $\mathcal{H}_0 \rightarrow \dots \rightarrow \mathcal{H}_m$  that satisfies the following conditions.

- S1**  $\mathcal{H}_0 = \{P(t) \mid P \in \mathcal{X}\}$ , i.e.  $\mathcal{H}_0$  contains only the term  $t$  in the predicates from  $\mathcal{X}$ .
- S2** For all  $P(t') \in \mathcal{H}_m$ , we have  $t' = wt$  with  $w \in \mathbf{F}^*$ , i.e. every term in  $\mathcal{H}_m$  has suffix  $t$ .
- S3**  $\mathcal{Y} = \{P \mid P(t) \in \mathcal{H}_m\}$ , i.e.  $\mathcal{H}_m$  contains the term  $t$  in exactly the predicates from  $\mathcal{Y}$ .
- S4** For every  $(t', \mathbf{c}) \in \text{act}(\mathcal{H}_m)$ , we have  $t' = t$  and  $\mathbf{c}$  is decreasing.

<sup>3</sup>Here we assume that the maximal term is not  $a$ .

The *height* of this expansion sequence is  $\max\{|wt| - |t| \mid P(wt) \in \mathcal{H}_m\}$ . The *height* of the shortcut  $(\mathcal{X}, \mathcal{Y})$  is the smallest height of an expansion sequence satisfying **S1–S4** for  $(\mathcal{X}, \mathcal{Y})$  and some ground term  $t$ .

It is easy to check that, whenever there is a sequence satisfying **S1–S4**, then there is a similar sequence for *all* ground terms  $t$ . This is because, without applying the expansion rule to decreasing clauses with active term  $t$ , the term  $t$  cannot be decomposed, and all other applications of expansion rules can only extend  $t$  by more function symbols. For this reason, shortcuts  $(\mathcal{X}, \mathcal{Y})$  do not have to refer to a specific term, because the definition is satisfied either for all terms or for none.

There always exists the trivial shortcut  $(\emptyset, \emptyset)$ , which corresponds to a trivially empty expansion sequence. This shortcut does not have a well-defined height, so we just fix it to  $-1$  (which is impossible for any other shortcut).

Obviously,  $\mathcal{C}$  is terminating iff there is a shortcut of the form  $(\{A\}, \mathcal{Y})$ , because  $a$  can never be active at decreasing clauses. We can thus try to construct an expansion sequence as in **S1–S4** for  $(\{A\}, \mathcal{Y})$ , by composing it from smaller sequences. We could try to use a bottom-up approach: if we have to add the atom  $P(fa)$  due to an increasing clause in  $\mathcal{C}$ , we can try to deal with this atom by finding a shortcut of the form  $(\{P\}, \mathcal{Y}')$  and apply it to  $fa$ . Once we have found this shortcut, we can apply the expansion rule to any atoms  $Q(fa)$ ,  $Q \in \mathcal{Y}'$ , for which a matching decreasing clause  $Q(fx) \rightarrow Q^f(x)$  exists, thereby adding the smaller term  $a$  to the interpretation of  $Q^f$ , which then may lead to more atoms over  $a$ .

Any time we produce a new term  $t$  in this process, we look for a shortcut that we can use for  $t$  in a recursive manner. This bottom-up approach can easily lead to a large complexity algorithm.

Here we will instead compute shortcuts of increasing height in a top-down manner, using the following observations. If there is a shortcut  $(\{A\}, \mathcal{Y})$  of height 0, it is quite simple to find it, because it is not necessary to produce terms of height  $> 0$  using increasing clauses. Hence, we simply have to find a way to satisfy all flat clauses without the expansion rule becoming applicable for any increasing clauses.

Now consider a shortcut  $(\{A\}, \mathcal{Y})$  of height larger than 0, and let  $t$  be a term of maximal height in the corresponding expansion sequence for  $a$ . Then we can extract a sub-sequence for  $t$ , starting with the atoms  $P(t)$  that were initially produced by increasing clauses, applying the expansion rule only to flat clauses using  $t$  (since  $t$  is maximal), and ending with some atoms  $Q(t)$ , where the term  $t$  is active only at decreasing clauses. We can extract in this way a shortcut  $(\mathcal{X}, \mathcal{Y})$  of height 0.

Since the original shortcut was of positive height, we know that  $t = ft'$  for some  $f \in \mathbf{F}$ , and that  $t$  was produced by increasing clauses  $P^f(x) \rightarrow P(fx)$  with  $P \in \mathcal{X}$ . Hence, can similarly extract a sub-sequence for  $t'$ , by again starting with the atoms  $P(t')$  produced by increasing clauses, and ending with only atoms

$Q(t')$  active at decreasing clauses, such that whenever we produce a term of larger height, we can employ a shortcut of height 0 that we had extracted in the previous step. The shortcut corresponding to the resulting sub-sequence will be of height 1. In this way, we can eventually compose the expansion sequence for  $(\{A\}, \mathcal{Y})$  from expansion sequences of smaller height.

**Example 8.** Let the following be  $\mathcal{C}$ ,  $\mathbf{P}_a = \{P^f, Q_1, Q_2^f\}$ , with some of the clauses obtained from guessing  $\mathbf{P}_a$  are omitted.

<b>start clause:</b>	$\mathfrak{c}_0 = \top \rightarrow A(a)$
<b>flat clauses:</b>	$\mathfrak{c}_1 = A(x) \rightarrow P^f(x)$
	$\mathfrak{c}_2 = P(x) \rightarrow Q_1(x) \vee Q_2(x)$
	$\mathfrak{c}_3 = Q_1(x) \wedge Q_2(x) \rightarrow \perp$
	$\mathfrak{c}_4 = Q_2^f(x) \rightarrow Q_1(x)$
<b>increasing/decreasing</b>	$\mathfrak{c}_{5/6} = P^f(x) \leftrightarrow P(fx)$
<b>clauses:</b>	$\mathfrak{c}_{7/8} = Q_2^f(x) \leftrightarrow Q_2(fx)$

A terminating expansion sequence is the following (some of the steps are omitted, because of the absent clauses):

$$\begin{aligned}
\mathcal{H}_0 = \{A(a)\} &\xrightarrow{\mathfrak{c}_{1,a,P^f}} \mathcal{H}_2 = \{A(a), P^f(a)\} \xrightarrow{\mathfrak{c}_{5,a,P}} \mathcal{H}_3 = \{A(a), P^f(a), P(fa)\} \\
&\xrightarrow{\mathfrak{c}_{2,fa,Q_2}} \mathcal{H}_4 = \{A(a), P^f(a), P(fa), Q_2(fa)\} \\
&\xrightarrow{\mathfrak{c}_{8,fa,Q_2^f}} \mathcal{H}_5 = \{A(a), P^f(a), P(fa), Q_2(fa), Q_2^f(a)\} \\
&\xrightarrow{\mathfrak{c}_{4,a,Q_1}} \mathcal{H}_6 = \{A(a), P^f(a), P(fa), Q_2(fa), Q_2^f(a), Q_1(a)\}
\end{aligned}$$

The term of maximal height is  $fa$ . It was produced by  $\mathfrak{c}_5 = P^f(x) \rightarrow P(fx)$ . After  $fa$  has been created, it is active at  $\mathfrak{c}_2 = P(x) \rightarrow Q_1(x) \vee Q_2(x)$ . We see that it was deactivated in the expansion sequence by adding  $Q_2(fa)$  to the next interpretation. This has activated the decreasing clause,  $\mathfrak{c}_8$ .

In this way we can extract the shortcut  $(\{P\}, \{P, Q_2\})$  for height 0 from the terminating expansion sequence. This shortcut will correspond to the following expansion sequence:

$$\mathcal{H}'_0 = \{P(fa)\} \xrightarrow{\mathfrak{c}_{2,fa,Q_2}} \mathcal{H}'_1 = \{P(fa), Q_2(fa)\}$$

Of course, this shortcut and expansion sequence can be used with an arbitrary term,  $t$ .

$$\mathcal{H}'_0 = \{P(t)\} \xrightarrow{\mathfrak{c}_{2,t,Q_2}} \mathcal{H}'_1 = \{P(t), Q_2(t)\}$$

Having this shortcut, we can present the original expansion sequence in the following way:

---

**Algorithm 1** (Main algorithm)

---

**Input:** A set  $\mathcal{C}$  of propagation rules**Output:** true iff  $\mathcal{C}$  terminates

```
1:  $\mathfrak{S}_0 \leftarrow \{(\emptyset, \emptyset)\}$ 
2:  $n \leftarrow 0$ 
3: repeat
4:    $n \leftarrow n + 1$ 
5:    $\mathfrak{S}_n \leftarrow \text{nextShortcuts}(\mathcal{C}, \mathfrak{S}_{n-1})$ 
6:   if there exists  $(\{A\}, \mathcal{Y}) \in \mathfrak{S}_n$  then
7:     return true
8:   end if
9: until  $\mathfrak{S}_n = \mathfrak{S}_{n-1}$ 
10: return false
```

---

$$\begin{aligned} \{A(a)\} &\xrightarrow{c_{1,a,P^f}} \{A(a), P^f(a)\} \xrightarrow{\text{incr}} \boxed{\text{shortcut}} \xrightarrow{\text{decr}} \{A(a), P^f(a), Q_2^f(a)\} \\ &\xrightarrow{c_{4,a,Q_1}} \{A(a), P^f(a), Q_2^f(a), Q_1(a)\}. \end{aligned}$$

Notice that the atoms  $P(fa), Q_2(fa)$  are not visible in the final interpretation, because they are hidden in the shortcut. When we reconstruct a model, we have to put them back there. But in constructing a shortcut we are interested only in the atoms containing terms of small height. In this case, constants.

At this point our algorithm will check if there exists a shortcut of the form  $(\{P, Q_2\}, \mathcal{Y})$  and it will discover that all flat and increasing clauses are satisfied for  $\{P(fa), Q_2(fa)\}$  hence there is a shortcut  $(\{P, Q_2\}, \{P, Q_2\})$  as required.<sup>4</sup>

In this way we have obtained the shortcut  $(\{A\}, \{A, P^f, Q_2^f, Q_1\})$  of height 1 based on the shortcut of height 0.

## 4 Detecting the Existence of a Finite Herbrand Model

As explained above, in order for a finite Herbrand model of  $\mathcal{C}$  to exist, there must be a shortcut of the form  $(\{A\}, \mathcal{Y})$ , that corresponds to a terminating expansion sequence. We are not given such a sequence from which we can extract all or some shortcuts. Instead we compute all possible shortcuts for  $\mathcal{C}$ . If a shortcut  $(\{A\}, \mathcal{Y})$  for some  $\mathcal{Y} \subseteq \mathbf{P}$  is in this set of shortcuts, then we can be sure that a terminating expansion sequence exists.

This idea is formalized in Algorithm 1.

We describe next the procedure  $\text{nextShortcuts}(\mathcal{C}, \mathfrak{S}_{n-1})$ , which computes all

---

<sup>4</sup>This last step seems redundant in this example, but it is not in general.

possible shortcuts of height at most  $n$  based on all shortcuts of height at most  $n - 1$ .

## 4.1 Possibilities

The algorithm for computing shortcuts is using a procedure for the so called *possibilities*. Possibilities are just different ways of extending an expansion sequence with a term  $t$  with respect to *flat* clauses in  $\mathcal{C}$ , whereas the bigger terms required by the increasing clauses are dealt with using the already computed shortcuts.

**Definition 9.** The set  $\text{possibilities}(\mathbf{c})$  for a flat clause  $\mathbf{c}$  is  $\mathbf{h}(\mathbf{c})$ , i.e. the set of all predicates in the head of  $\mathbf{c}$ . For a finite set  $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_k\}$  of clauses, we define  $\text{possibilities}(\mathcal{C})$  as

$$\{\{P_1, \dots, P_k\} \mid \forall i \in \{1, \dots, k\}: P_i \in \text{possibilities}(\mathbf{c}_i)\}.$$

Finally, for  $\mathcal{X} \subseteq \mathbf{P}$ , we set  $\text{possibilities}(\mathcal{X}) := \text{possibilities}(\mathcal{C}|_{\mathcal{X}})$ , where  $\mathcal{C}|_{\mathcal{X}}$  is defined as the set of all flat clauses  $\mathbf{c}$  from  $\mathcal{C}$  with  $\mathbf{b}(\mathbf{c}) \subseteq \mathcal{X}$ .

This definition is basically the same as in [2]. Intuitively,  $\text{possibilities}(\dots)$  describes all possibilities of deactivating a given term at the involved clauses (either a single clause, a set of clauses, or the specific set of clauses activated by  $\mathcal{X}$ ). If there are no predicates on the right-hand side of  $\mathbf{c}$ , then  $\text{possibilities}(\mathbf{c}) = \emptyset$ , and if such a  $\mathbf{c}$  belongs to  $\mathcal{C}$ , then  $\text{possibilities}(\mathcal{C}) = \emptyset$ . In contrast, if  $\mathcal{C}|_{\mathcal{X}} = \emptyset$ , then  $\text{possibilities}(\mathcal{X}) = \text{possibilities}(\emptyset) = \{\emptyset\}$ , i.e. there is exactly one possibility, which is to do nothing, since no terms are active and thus no action is required.

**Example 10.** Let us look at some clauses in Example 8.

$$\begin{aligned} \text{possibilities}(\mathbf{c}_1) &= \{P^f\} \\ \text{possibilities}(\mathbf{c}_2) &= \{Q_1, Q_2\} \\ \text{possibilities}(\mathbf{c}_3) &= \emptyset \\ \text{possibilities}(\mathbf{c}_4) &= \{Q_1\} \end{aligned}$$

Now, let us compute possibilities for some sets of predicates.

- $\text{possibilities}(\{A\}) = \{\{P^f\}\}$ , because  $\mathbf{b}(\mathbf{c}_1) \in \{A\}$ .
- $\text{possibilities}(\{P\}) = \{\{Q_1\}, \{Q_2^f\}\}$ , which corresponds to a choice we have when deactivating a term at a clause  $\mathbf{c}_2$ .
- Notice that  $\text{possibilities}(\{Q_1, Q_2\}) = \emptyset$ , hence we cannot terminate a sequence if we reach these two predicates with the same term.
- $\text{possibilities}(\{Q_1\}) = \{\emptyset\}$ , because  $\mathcal{C}|_{\{Q_1\}} = \emptyset$ .



---

**Algorithm 2** possibilities( $\mathcal{C}, \mathfrak{S}, \mathcal{X}$ )

---

**Input:** A set  $\mathcal{C}$  of propagation rules, a set  $\mathfrak{S}$  of shortcuts, a set  $\mathcal{X}$  of predicates,

**Output:** The set  $\mathfrak{P}$  of possibilities for  $\mathcal{X}$  w.r.t.  $\mathcal{C}, \mathfrak{S}$

```
1:  $\mathfrak{P} \leftarrow \{\emptyset\}$ 
2: for all  $f \in \mathbf{F}$  do ▷ Increasing clauses
3:    $\mathfrak{P} \leftarrow \{\mathcal{Z} \cup \mathcal{W}^{+f} \mid \mathcal{Z} \in \mathfrak{P}, (\mathcal{X}^{-f}, \mathcal{W}) \in \mathfrak{S}\}$ 
4: end for
5: for all  $\mathfrak{c} = P_1(x) \wedge \dots \wedge P_k(x) \rightarrow Q_1(x) \vee \dots \vee Q_\ell(x) \in \mathcal{C}$  do ▷ Flat clauses
6:   if  $\{P_1, \dots, P_k\} \subseteq \mathcal{X}$  then
7:      $\mathfrak{P} \leftarrow \{\mathcal{Z} \cup \{Q_1\}, \dots, \mathcal{Z} \cup \{Q_\ell\} \mid \mathcal{Z} \in \mathfrak{P}\}$ 
8:   end if
9: end for
10: return  $\mathfrak{P}$ 
```

---

- possibilities( $\{Q_2\}$ ) =  $\{\emptyset\}$ , for the same reason as above.

In the presence of shortcuts, we can extend the notion of possibilities to the increasing clauses, the predicates labeled with a function symbol, and sets of such predicates that include them. Notice that if  $P^f \in \mathcal{X}$  and there are no shortcuts of the form  $(\{P\}, \mathcal{Y})$ , we don't know what to do with the bigger terms with the prefix  $f$  and then if we take increasing clauses into consideration, possibilities( $\mathcal{X}$ ) =  $\emptyset$ . If  $\mathfrak{S}$  is a set of shortcuts, the extended possibilities( $P^f$ ) =  $\{Q^f \in \mathbf{P} \mid (\{P\}, \mathcal{Y}) \in \mathfrak{S}, Q \in \mathcal{Y}\}$ . The extended notion of possibilities is defined in the Algorithm 2.

For convenience, given a set  $\mathcal{X} \subseteq \mathbf{P}$  and a function symbol  $f \in \mathbf{F}$ , we define the following abbreviations to translate back and forth between the predicates  $P$  and  $P^f$ :

$$\begin{aligned}\mathcal{X}^{+f} &:= \{P^f \in \mathbf{P} \mid P \in \mathcal{X}\}, \\ \mathcal{X}^{-f} &:= \{P \in \mathbf{P} \mid P^f \in \mathcal{X}\}.\end{aligned}$$

Notice that if a predicate is in  $\mathcal{X}^{+f}$ , it has the form  $P^f$ . Hence if  $P(ft)$  is in an interpretation, then  $P^f(t)$  must be there too. Hence the term  $ft$  is decomposed in  $P^f$ . Notice too that not all predicates in  $\mathcal{X}$  has the form  $P^f$  defined in the set  $\mathbf{P}$ . Similarly, if a predicate is in  $\mathcal{X}^{-f}$ , it has the form  $P$ , where  $P^f \in \mathcal{X}$ . We know that if  $P^f(t)$  is in an interpretation, then  $P(ft)$  must be there too.

Algorithm 2 computes all possibilities for a set of predicates  $\mathcal{X}$ , with respect to a set of shortcuts  $\mathfrak{S}$ , and a set of clauses  $\mathcal{C}$ . Hence it has two parts: computing possibilities for increasing clauses with the available shortcuts and computing possibilities for flat clauses.

Notice that since  $(\emptyset, \emptyset)$  is always a shortcut, if the set  $\mathcal{X}^{-f}$  is empty, then in Line 3 the set  $\mathfrak{P}$  is not changed. In contrast, if there is no shortcut of the form

$(\mathcal{X}^{-f}, \mathcal{Y}) \in \mathfrak{S}$ , then  $\mathfrak{P}$  will become empty. In Line 7, if  $\mathfrak{P}$  is already empty, then it will stay empty; if it is not, it can become empty in case  $\ell = 0$ .

Algorithm 2 can be optimized, because in the present formulation it produces many unuseful possibilities, i.e. it includes predicates even if a clause is already satisfied. Nevertheless, all *productive* possibilities are included in  $\mathfrak{P}$ .

**Example 11.** Consider Example 8 and assume that we have not yet computed any shortcuts. We consider the computation of  $\text{possibilities}(\mathcal{C}, \mathfrak{S}, \{A\})$  with  $\mathfrak{S} = \{(\emptyset, \emptyset)\}$ .

At the beginning the set of possibilities  $\mathfrak{P}$  is initialized to the set containing empty set, which means that no extension is required. Since  $\{A\}$  does not contain any predicates of the form  $P^f$ , for each  $f \in \mathbf{F}$ , hence for each  $f \in \mathbf{F}$ , the empty shortcut  $(\emptyset, \emptyset)$  is used and thus we add the empty set to the empty possibility in  $\mathfrak{P}$ , hence the set of possibilities remains unchanged (line 4).

In the for-loop for the flat clauses, the algorithm considers  $\mathbf{c}_1 = A(x) \rightarrow P^f(x)$ . Inside the body of this loop, the set of possibilities is redefined to  $\mathfrak{P} = \{\{A, P^f\}\}$ . Since no other flat clauses satisfy the conditions in the if-condition, the algorithm will return  $\mathfrak{P} = \{\{A, P^f\}\}$ , i.e. there is only one possibility for  $\{A\}$ , which is  $\{A, P^f\}$ .<sup>5</sup>

If we want to compute possibilities for  $\{A, P^f\}$ , we will get  $\emptyset$  as a result. This is because there is no shortcut of the form  $(\{P\}, \mathcal{W})$  in  $\mathfrak{S} = \{(\emptyset, \emptyset)\}$ . Since there is no  $\mathcal{W}^{+f}$  available,  $\mathfrak{P}$  becomes empty (Line 3). It will stay empty through the loop for flat clauses too (there is no  $\mathcal{Z}$  available in Line 7).

The following lemma shows that Algorithm 2 is sound.

**Lemma 12.** *Assume that for every  $(\mathcal{X}, \mathcal{W}) \in \mathfrak{S}$  there is a shortcut  $(\mathcal{X}, \mathcal{Y})$  for  $\mathcal{C}$  of height  $\leq n$  with  $\mathcal{Y} \subseteq \mathcal{W}$ , and let  $\mathcal{Z}$  be an element of  $\text{possibilities}(\mathcal{C}, \mathfrak{S}, \mathcal{X})$ . Then, for every ground term  $t$ , there is an expansion sequence  $\mathcal{H}_0 \rightarrow \dots \rightarrow \mathcal{H}_m$  of height  $\leq n + 1$  satisfying **S1**, **S2** (see Definition 7), and the following conditions.*

**S3'**  $\mathcal{X} \cup \mathcal{Z} \supseteq \{P \mid P(t) \in \mathcal{H}_m\}$ , i.e.  $\mathcal{H}_m$  contains the term  $t$  only in the predicates from  $\mathcal{X} \cup \mathcal{Z}$ .

**S4'** For every  $(t', \mathbf{c}) \in \text{act}(\mathcal{H}_m)$  we have  $t' = t$  and, if  $\mathbf{c}$  is not decreasing, then  $\mathbf{b}(\mathbf{c}) \not\subseteq \mathcal{X}$ .<sup>6</sup>

*Proof.* We construct the required expansion sequence, initializing  $\mathcal{H}_0$  to the set  $\{P(t) \mid P \in \mathcal{X}\}$ , as required for **S1**. We then have  $\text{act}(\mathcal{H}_0) = \{(t, \mathbf{c}) \mid \mathbf{b}(\mathbf{c}) \subseteq \mathcal{X}\}$ .

<sup>5</sup>For the sake of clarity this example (following Example 8) omits some clauses that should be in  $\mathcal{C}$  because of the choice of  $\mathbf{P}_a$ . The set of possibilities for  $\{A\}$  is exactly  $\mathbf{P}_a$ . In Example 8  $\mathbf{P}_a$  can be defined as  $\mathbf{P}_a = \{A, P^f, Q_2^f, Q_1\}$ .

<sup>6</sup>Computing possibility for  $\mathcal{X}$  corresponds to an expansion sequence, which allows for some clauses to be still activated, but not those that were activated in the predicates in  $\mathcal{X}$  alone.

We proceed to deactivate  $t$  at all such increasing and flat clauses; the decreasing clauses are allowed to have  $t$  active due to Condition **S4'**.

Consider first the increasing clauses with active term  $t$ . We follow the computation of Algorithm 2 that led to  $\mathcal{Z}$ . By Line 3, for each  $f \in \mathbf{F}$ , there must be a pair  $(\mathcal{X}^{-f}, \mathcal{W}) \in \mathfrak{S}$  with  $\mathcal{W}^{+f} \subseteq \mathcal{Z}$ . By assumption, there is a shortcut  $(\mathcal{X}^{-f}, \mathcal{Y})$  of height  $\leq n$  with  $\mathcal{Y} \subseteq \mathcal{W}$ . Such a shortcut induces an expansion sequence of height  $\leq n$  starting with the term  $ft$  in the predicates  $\mathcal{X}^{-f}$ . This sequence uses only terms with suffix  $ft$  and ends with  $ft$  active only at decreasing clauses  $P(fx) \rightarrow P^f(x)$  with  $P \in \mathcal{Y} \subseteq \mathcal{W}$ .

We hence start constructing our expansion sequence for  $t$  by expansion steps for all increasing clauses with active term  $t$ , followed by the sequences of height  $\leq n$  that exist due to the pairs  $(\mathcal{X}^{-f}, \mathcal{W}) \in \mathfrak{S}$  as described above. We then add expansion steps for the decreasing clauses with active term  $ft$ , which induces new atoms of the form  $P^f(t)$  with  $P^f \in \mathcal{W}^{+f} \subseteq \mathcal{Z}$ . After this,  $t$  is still the only active term, and it is not active anymore at any increasing clauses  $\mathbf{c}$  with  $\mathbf{b}(\mathbf{c}) \subseteq \mathcal{X}$ .

For  $\mathbf{c} = P_1(x) \wedge \dots \wedge P_k(x) \rightarrow Q_1(x) \vee \dots \vee Q_\ell(x) \in \mathcal{C}$  with  $\{P_1, \dots, P_k\} \subseteq \mathcal{X}$ , by Line 7 we must have  $Q_\ell \in \mathcal{Z}$  for some  $i \in \{1, \dots, m\}$ . Hence, we can add the appropriate expansion step to deactivate  $t$  at  $\mathbf{c}$ . The final Herbrand interpretation  $\mathcal{H}_m$  clearly satisfies **S2** and **S3'**. Moreover, the term  $t$  is only active at flat or increasing clauses whose body contains at least one predicate from  $\mathcal{Z} \setminus \mathcal{X}$ , as well as possibly some decreasing clauses. Hence, Condition **S4'** is also satisfied. Finally, the obtained expansion sequence for  $t$  is of height  $\leq n + 1$  since it is composed only of expansion steps for flat clauses with active term  $t$ , as well as expansion sequences of height  $\leq n$  for terms  $ft$  with  $f \in \mathbf{F}$  (and associated steps for increasing and decreasing clauses translating between  $t$  and  $ft$ )  $\square$

Next we explain in more detail, but still informally, how computing possibilities is used for obtaining a shortcut.

## 4.2 Shortcuts of Height 0

To construct a shortcut of height 0 starting from a set  $\mathcal{X}$  of predicates, we need to choose from  $\text{possibilities}(\mathcal{X})$  in such a way that no predicates of the form  $P^f$  are reached. One way to realize this is to assume for now that  $\text{possibilities}(\mathbf{c}) = \emptyset$  for all increasing clauses  $\mathbf{c}$ , and extend Definition 9 accordingly, obtaining an empty set of possibilities whenever an increasing clause is involved [2]. We will relax this assumption later, when we consider shortcuts of larger height (see Algorithm 2).

In order to compute all possible shortcuts of height 0 for  $\mathcal{X}$ , we must then consider

$$\begin{aligned}\mathfrak{P}_1 &:= \text{possibilities}(\mathcal{X}) \\ \mathfrak{P}_2 &:= \bigcup \{\text{possibilities}(\mathcal{X} \cup \mathcal{Z}) \mid \mathcal{Z} \in \mathfrak{P}_1\} \\ &\vdots \\ \mathfrak{P}_n &:= \bigcup \{\text{possibilities}(\mathcal{X} \cup \mathcal{Z}) \mid \mathcal{Z} \in \mathfrak{P}_{n-1}\}\end{aligned}$$

until there are no new possibilities, i.e.  $\mathfrak{P}_n = \mathfrak{P}_{n-1}$ .

Intuitively, we iteratively expand the set  $\mathcal{X}$  by new possibilities  $\mathcal{Z}$  (which are sets of predicates), until nothing more needs to be added.<sup>7</sup> Each of the final possibilities represents one way of using the expansion rule to deactivate a given term  $t$  at all flat clauses, assuming that we start with the atoms  $P(t)$  for all  $P \in \mathcal{X}$ . Hence, each  $\mathcal{Z} \in \mathfrak{P}_n$  yields a kind of *extended shortcut*  $(\mathcal{X}, \mathcal{Y})$  of height 0 with  $\mathcal{Y} \subseteq \mathcal{X} \cup \mathcal{Z}$ . Note that  $\mathcal{Z}$  may contain superfluous predicates that are not necessary to form a shortcut, because some clauses are considered multiple times during the computation, and a different possibility may be chosen each time.

### 4.3 Shortcuts of larger height

We proceed to compute shortcuts by induction on their height: assuming that we know all possible shortcuts of height (at most)  $n$ , we use them to construct new shortcuts of height  $n + 1$ . In an expansion sequence of height  $n + 1$ , we now allow a term  $t$  to become active at increasing clauses  $P^f(x) \rightarrow P(fx)$ , but only if we know a shortcut for  $\{P\}$ , or more generally for a set predicates  $\mathcal{X}$  that all contain the higher term  $ft$ . Assuming that  $(\mathcal{X}, \mathcal{Y})$  is such a shortcut of height  $\leq n$  (which we have already computed before), the corresponding expansion sequence may cause  $ft$  to become active at decreasing clauses  $Q(fx) \rightarrow Q^f(x)$  with  $Q \in \mathcal{Y}$ , which forces us to add  $t$  to each such  $Q^f$ . Essentially, we can now extend the definition of  $\text{possibilities}(\mathfrak{c})$  for increasing clauses  $\mathfrak{c}$  by considering shortcuts to deal with the propagation of the produced higher terms.

### 4.4 Computing shortcuts

The above ideas are formalized in Algorithm 3 which uses Algorithm 2 to compute all shortcuts of height  $\leq n + 1$  from the already known shortcuts of height  $\leq n$ . The algorithm works with a set  $\mathfrak{T}$  of triples of the form  $(\mathcal{X}, R_{\mathcal{X}}, V_{\mathcal{X}})$  with  $\mathcal{X} \subseteq R_{\mathcal{X}}$  and  $V_{\mathcal{X}} \subseteq R_{\mathcal{X}}$ , where  $\mathcal{X}$  represents the starting predicates,  $R_{\mathcal{X}}$  is the set of

---

<sup>7</sup>At each step we have to check possibilities for the augmented set of predicates, because more predicates reached means that more clauses may be activated. This is the difference with our previous work [2] concerning the anti-Horn clauses.

---

**Algorithm 3** ( $\text{nextShortcuts}(\mathcal{C}, \mathfrak{S})$ )

---

**Input:** A set  $\mathcal{C}$  of propagation rules and a set  $\mathfrak{S}$  of shortcuts

**Output:** A set of shortcuts for the next height

- 1:  $\mathfrak{T} \leftarrow \{(\{A\}, \{A\}, \emptyset)\} \cup \{(\mathcal{X}^{-f}, \mathcal{X}^{-f}, \emptyset) \mid \mathcal{X} \subseteq \mathbf{P}, f \in \mathcal{F}\}$
  - 2: **while** there is  $(\mathcal{X}, R_{\mathcal{X}}, V_{\mathcal{X}}) \in \mathfrak{T}$  with  $R_{\mathcal{X}} \setminus V_{\mathcal{X}} \neq \emptyset$  **do**
  - 3:      $\mathfrak{T} \leftarrow \mathfrak{T} \setminus \{(\mathcal{X}, R_{\mathcal{X}}, V_{\mathcal{X}})\}$
  - 4:     **for all**  $\mathcal{Z} \in \text{possibilities}(\mathcal{C}, \mathfrak{S}, R_{\mathcal{X}})$  **do**
  - 5:          $\mathfrak{T} \leftarrow \mathfrak{T} \cup \{(\mathcal{X}, R_{\mathcal{X}} \cup \mathcal{Z}, R_{\mathcal{X}})\}$
  - 6:     **end for**
  - 7: **end while**
  - 8: **return**  $\{(\mathcal{X}, R_{\mathcal{X}}) \mid (\mathcal{X}, R_{\mathcal{X}}, R_{\mathcal{X}}) \in \mathfrak{T}\}$
- 

predicates that have already been reached from  $\mathcal{X}$  by choosing some possibilities, and  $V_{\mathcal{X}}$  is a set of predicates for which we have already computed the possibilities. While  $R_{\mathcal{X}} \setminus V_{\mathcal{X}} \neq \emptyset$ , there are still some new predicates that need to be processed using Algorithm 2. The form of the initial triples in Line 1 reflects the kind of shortcuts that we need: first, a shortcut starting from  $\{A\}$  (to show termination of  $\mathcal{C}$ ), and, second, shortcuts for any combination of predicates  $\mathcal{X}$  which may contain a common term  $ft$  due to increasing clauses for  $\mathcal{X}^{+f}$  being applied to the shallower term  $t$ . The latter case also covers the trivial shortcut  $(\emptyset, \emptyset)$ , which is always included in the output of Algorithm 3.<sup>8</sup>

**Example 13.** Consider Example 8. If the algorithm `newShortcuts` is used with the empty set of shortcuts, i.e.  $\mathfrak{S}_0 = \{(\emptyset, \emptyset)\}$ , one of the triples initialized in Line 1 is  $(\{A\}, \{A\}, \emptyset)$ . This triple is an element of the set  $\mathfrak{T}$ .

It is removed from there in the while-loop starting at Line 2. Now, possibilities for  $\{A\}$  are computed, and as we have already seen in Example 11, the only possibility for  $\{A\}$  is  $\{A, P^f\}$ . Hence a new triple  $(\{A\}, \{A, P^f\}, \{A\})$  is added to  $\mathfrak{T}$ .<sup>9</sup>

In the next execution of the while-loop, this triple is again removed from  $\mathfrak{T}$ . The possibilities for  $\{A, P^f\}$  are computed. But as we have already seen in Example 11, there are no possibilities for this set. Hence nothing is added to  $\mathfrak{T}$  in Line 5. There are no shortcut of the form  $(\{A\}, \mathcal{Y})$  if  $\mathfrak{S}$  is empty.

In the same run of the algorithm `newShortcuts`, another triple  $(\{P\}, \{P\}, \emptyset)$  is considered. Inside the while-loop it is removed from  $\mathfrak{T}$ , and possibilities for  $\{P\}$  are being computed. The only clause relevant for  $P$  is  $\mathbf{c}_2 = P(x) \rightarrow Q_1(x) \vee Q_2(x)$ . Hence possibilities for  $P$  computed by Algorithm 2 is the set  $\{\{P, Q_1\}, \{P, Q_2\}\}$ .

---

<sup>8</sup>First the main algorithm initializes set of shortcuts  $\mathfrak{S}_0$  to contain the empty shortcut. Then for the next shortcuts, in line 3, if we take  $\mathcal{X} = \emptyset$ , then the triple  $(\emptyset, \emptyset, \emptyset)$  is created, and never modified. Thus, line 8 returns  $(\emptyset, \emptyset)$ .

<sup>9</sup>In fact the set of possibilities will be larger. For explanation look at the footnote in Example 11.

Hence two triples are added to  $\mathfrak{T}$ :  $(\{P\}, \{P, Q_1\}, \{P\})$  and  $(\{P\}, \{P, Q_2\}, \{P\})$ .

In the next execution of the while-loop, these triples will be replaced in  $\mathfrak{T}$  by  $(\{P\}, \{P, Q_1\}, \{P, Q_1\})$  and  $(\{P\}, \{P, Q_2\}, \{P, Q_2\})$ . Finally, from these two triples we obtain two shortcuts:  $(\{P\}, \{P, Q_1\})$  and  $(\{P\}, \{P, Q_2\})$ .

The soundness of the procedure is stated in the following lemma.

**Lemma 14.** *If for every  $(\mathcal{X}, \mathcal{W}) \in \mathfrak{S}_n$  there is a shortcut  $(\mathcal{X}, \mathcal{Y})$  for  $\mathcal{C}$  of height  $\leq n$  with  $\mathcal{Y} \subseteq \mathcal{W}$ , then for every  $(\mathcal{X}, R_{\mathcal{X}}) \in \text{nextShortcuts}(\mathcal{C}, \mathfrak{S})$ , there is a shortcut  $(\mathcal{X}, \mathcal{Y})$  for  $\mathcal{C}$  of height  $\leq n + 1$  with  $\mathcal{Y} \subseteq R_{\mathcal{X}}$ .*

*Proof.* We consider an arbitrary ground term  $t$  and a triple  $(\mathcal{X}, R_{\mathcal{X}}, V_{\mathcal{X}})$  as it appears in the final set  $\mathfrak{T}$  in Line 8 of Algorithm 3. We follow this triple through the whole computation, starting with an initial triple  $(\mathcal{X}, \mathcal{X}, \emptyset)$  in Line 1, and show that the following invariant remains satisfied: there exists an expansion sequence  $\mathcal{H}_0 \rightarrow \dots \rightarrow \mathcal{H}_m$  of height  $\leq n + 1$  that satisfies **S1**, **S2**, and the following conditions.

**S3''**  $R_{\mathcal{X}} \supseteq \{P \mid P(t) \in \mathcal{H}_m\}$ , i.e.  $\mathcal{H}_m$  contains the term  $t$  only in the predicates from  $R_{\mathcal{X}}$ .

**S4''** For every  $(t', \mathbf{c}) \in \text{act}(\mathcal{H}_m)$ , we have  $t' = t$  and, if  $\mathbf{c}$  is not decreasing, then  $\mathbf{b}(\mathbf{c}) \not\subseteq V_{\mathcal{X}}$ .

This condition is satisfied for the initial triple  $(\mathcal{X}, \mathcal{X}, \emptyset)$  since we can simply choose  $\mathcal{H}_0 = \mathcal{H}_m$  according to **S1**. Now assume that such a sequence  $\mathcal{H}_0 \rightarrow \dots \rightarrow \mathcal{H}_m$  exists for the triple  $(\mathcal{X}, R_{\mathcal{X}}, V_{\mathcal{X}})$  with  $R_{\mathcal{X}} \setminus V_{\mathcal{X}} \neq \emptyset$  in Line 2 and that the next triple  $(\mathcal{X}, R'_{\mathcal{X}}, V'_{\mathcal{X}}) = (\mathcal{X}, R_{\mathcal{X}} \cup \mathcal{Z}, R_{\mathcal{X}})$  is obtained via  $\mathcal{Z} \in \text{possibilities}(\mathcal{C}, \mathfrak{S}, R_{\mathcal{X}})$  in Line 5. By Lemma 12, there is an expansion sequence of height  $\leq n + 1$  that deactivates  $t$  at all clauses  $\mathbf{c}$  with  $\mathbf{b}(\mathbf{c}) \subseteq R_{\mathcal{X}}$ , thereby adding  $t$  to the interpretation of some predicates from  $\mathcal{Z}$ . From this expansion sequence, we can reuse all steps that deactivate terms at clauses that were not already deactivated in  $\mathcal{H}_m$ .

In the resulting Herbrand interpretation, we have the term  $t$  only in the predicates from  $R'_{\mathcal{X}} = R_{\mathcal{X}} \cup \mathcal{Z}$ , and, except for decreasing clauses,  $t$  can only be active at clauses  $\mathbf{c}$  with  $\mathbf{b}(\mathbf{c}) \not\subseteq V'_{\mathcal{X}} = R_{\mathcal{X}}$ . This shows that Conditions **S3''** and **S4''** remain satisfied.

For the final triple  $(\mathcal{X}, R_{\mathcal{X}}, V_{\mathcal{X}})$  with  $V_{\mathcal{X}} = R_{\mathcal{X}}$ , these conditions yield an expansion sequence satisfying **S1–S4** for the set  $\mathcal{Y} := \{P \mid P(t) \in \mathcal{H}_m\}$ , because  $(t, \mathbf{c}) \in \text{act}(\mathcal{H}_m)$  with  $\mathbf{b}(\mathbf{c}) \not\subseteq V_{\mathcal{X}} = R_{\mathcal{X}}$  is impossible when only the predicates from  $R_{\mathcal{X}}$  can contain  $t$  in  $\mathcal{H}_m$ . Since we have used Lemma 12 only w.r.t. the term  $t$ , the constructed expansion sequence is of height at most  $n + 1$ , which shows that  $(\mathcal{X}, \mathcal{Y})$  is a shortcut for  $\mathcal{C}$  of height  $\leq n + 1$ .  $\square$

We now prove correctness of Algorithm 1.

**Lemma 15.** (*Soundness*) *If Algorithm 1 returns true, then  $\mathcal{C}$  terminates.*

*Proof.* From Lemma 14, it follows that for every element  $(\mathcal{X}, \mathcal{W}) \in \mathfrak{S}_n$ , as computed in Line 5, there is a shortcut  $(\mathcal{X}, \mathcal{Y})$  of height  $\leq n$  with  $\mathcal{Y} \subseteq \mathcal{W}$ . Hence, if Algorithm 1 returns **true** in Line 7, this means that  $\mathcal{C}$  has a shortcut of the form  $(\{A\}, \mathcal{Y})$ . By Definition 7, there exists an expansion sequence starting in  $\mathcal{H}_0 = \{A(a)\}$  that is terminating since  $a$  cannot be active at decreasing clauses.  $\square$

**Lemma 16.** (*Completeness*) *If  $\mathcal{C}$  terminates, then Algorithm 1 returns true.*

*Proof.* If  $\mathcal{C}$  terminates, then by Definition 7 there must be a shortcut of the form  $(\{A\}, \mathcal{Y})$ . We show that all shortcuts that exist for  $\mathcal{C}$  must eventually be included in one of the sets  $\mathfrak{S}_n$  computed in Algorithm 1, and hence the algorithm must return **true**. More precisely, we show that, for all  $n \geq 0$ , all shortcuts  $(\mathcal{X}, \mathcal{Y})$  of height  $\leq n$  with either  $\mathcal{X} = \{A\}$  or  $\mathcal{X} = \mathcal{Z}^{-f}$  for some  $\mathcal{Z} \subseteq \mathbf{P}$  and  $f \in \mathbf{F}$  are contained in  $\mathfrak{S}_{n+1}$ . We show the claim by induction on  $n$ . The set  $\mathfrak{S}_0 = \{(\emptyset, \emptyset)\}$  contains all shortcuts of height  $\leq -1$ . Assume now that Algorithm 3 receives the set  $\mathfrak{S}_n$  as input, which contains all shortcuts of height at most  $n \geq -1$  that have one of the specified forms. We show that the algorithm returns all required shortcuts of height  $\leq n + 1$ . For this, consider any expansion sequence  $\mathcal{H}_0 \rightarrow \dots \rightarrow \mathcal{H}_m$  of height  $\leq n + 1$ , for some ground term  $t$  and sets  $\mathcal{X}, \mathcal{Y}$ , where  $\mathcal{X}$  is of the right form, satisfying conditions of Definition 7.

We describe how Algorithm 3 computes  $(\mathcal{X}, \mathcal{Y})$  based on the steps of the given expansion sequence. It computes the first possibility for predicates instantiated with  $t$  in flat or increasing clauses  $\mathbf{c}$  with  $\mathbf{b}(\mathbf{c}) \subseteq \mathcal{X}$  in Line 4.

Next it extends this possibility to obtain a fixed point. Each step in the extension sequence using a flat clause and the term  $t$  immediately gives rise to a single possibility as considered in Line 7 of Algorithm 2. For all involved increasing clauses with function symbol  $f$ , we additionally consider the sub-sequence of all expansion steps that are required by the atoms  $P(ft)$  with  $P \in X^{-f}$  and do not apply decreasing clauses to  $ft$ . This sub-sequence can be viewed as starting in  $\mathcal{H}'_0 = \{P(ft) \mid P \in \mathcal{X}^{-f}\}$  and clearly satisfies **S2–S4** for  $ft$  and some set  $\mathcal{W}$ . Hence,  $(\mathcal{X}^{-f}, \mathcal{W})$  is a shortcut of height  $\leq n$ , which is contained in  $\mathfrak{S}_n$  by the induction hypothesis and has the required form for Line 3 of Algorithm 2. Subsequently applying the required decreasing clauses corresponds to adding the set  $\mathcal{W}^{+f}$  to the possibility we are constructing. In case that  $\mathcal{X}^{-f} = \emptyset$ , the constructed sub-sequence is empty and corresponds to the empty shortcut  $(\emptyset, \emptyset)$ , which is always present, even in  $\mathfrak{S}_0$ .

In total, Algorithm 2 yields a possibility  $\mathcal{Z}$  for  $\mathcal{X}$  that is now added to  $R_{\mathcal{X}}$  in Line 5 of Algorithm 3 and corresponds to a subset of predicates that contain  $t$  in  $\mathcal{H}_m$ .

In subsequent steps, the set  $R_{\mathcal{X}}$  is further expanded by possibilities for  $R_{\mathcal{X}}$  until no new predicates are reached. The corresponding possibilities output by Algorithm 2 can be obtained as above, by considering the extended set  $R_{\mathcal{X}}$  as starting point for our sub-sequence. This may involve a lot of the expansion steps that were already considered for the previous possibilities, but is always correct in the sense that the computed possibility  $\mathcal{Z}$  represents a set of predicates that contains  $t$  in  $\mathcal{H}_m$ . In the end, the set  $\mathcal{Y} = \{P \mid P(t) \in \mathcal{H}_m\}$  can furthermore not contain any predicates that are not in  $R_{\mathcal{X}}$  since any such predicates must have been added by the expansion rule due to an active term in other predicates, and this rule can only be applied if it is really necessary to deactivate a term at a clause. This means that we obtain a triple  $(\mathcal{X}, R_{\mathcal{X}}, R_{\mathcal{X}})$  with  $R_{\mathcal{X}} = \mathcal{Y}$ , and thus Algorithm 3 returns the shortcut  $(\mathcal{X}, \mathcal{Y})$  when called with  $\mathfrak{S}_n$  as input, i.e. we have  $(\mathcal{X}, \mathcal{Y}) \in \mathfrak{S}_{n+1}$ .  $\square$

We can now examine the complexity of our algorithm.

**Lemma 17.** *Let  $n > 0$  be such that  $\mathfrak{S}_n$  was computed by Algorithm 1. Then  $\mathfrak{S}_{n-1} \subseteq \mathfrak{S}_n$ .*

*Proof.* For  $n = 1$ , it suffices to observe that Algorithm 3 (with the help of Algorithm 2) always produces the empty shortcut if the input set  $\mathfrak{S}$  contains it. Now assume  $\mathfrak{S}_{n-1} \subseteq \mathfrak{S}_n$ , and consider the computation steps in these two algorithms when given either  $\mathfrak{S}_{n-1}$  or  $\mathfrak{S}_n$ . Since all pairs from  $\mathfrak{S}_{n-1}$  are also included in  $\mathfrak{S}_n$ , all possibilities w.r.t.  $\mathfrak{S}_{n-1}$  are still possibilities w.r.t.  $\mathfrak{S}_n$ , i.e.  $\text{possibilities}(\mathcal{C}, \mathfrak{S}_{n-1}, \mathcal{X}) \subseteq \text{possibilities}(\mathcal{C}, \mathfrak{S}_n, \mathcal{X})$  holds for all  $\mathcal{X}$  (see Line 3 of Algorithm 2). Due to this, all triples computed by Algorithm 3 on input  $\mathfrak{S}_{n-1}$  are also computed when given  $\mathfrak{S}_n$  as input (see Line 4 of Algorithm 3). Hence,  $\mathfrak{S}_n = \text{nextShortcuts}(\mathcal{C}, \mathfrak{S}_{n-1}) \subseteq \text{nextShortcuts}(\mathcal{C}, \mathfrak{S}_n) = \mathfrak{S}_{n+1}$ .  $\square$

**Lemma 18.** *Algorithm 1 terminates after time at most exponential in the size of  $\mathcal{C}$ .*

*Proof.* Since the number of relevant predicates is linear, there are at most exponentially many possible shortcuts. Hence, the main loop of Algorithm 1 may be executed at most exponentially many times.

In each iteration, the execution of Algorithm 3 can be seen as a construction of a set of trees with the roots  $(\mathcal{X}, \mathcal{X}, \emptyset)$  and leaves of the form  $(\mathcal{X}, \mathcal{Y}, \mathcal{Y})$ . There are exponentially many such trees, because we create them for every set of the form  $\mathcal{X}^{-f}$ . The height of each tree is at most polynomial, because in each round of the loop the third component of the triple  $(\mathcal{X}, R_{\mathcal{X}}, V_{\mathcal{X}})$  grows. The branching factor is bounded exponentially, because we can add different possibilities  $\mathcal{Z}$  to the second component of the triple in each step. Hence, each tree contains at most exponentially many nodes, and thus each call of  $\text{nextShortcuts}(\mathcal{C}, \mathfrak{S}_n)$  can take only exponential time.



In conclusion, Algorithm 1 needs at most exponentially many iterations, each taking exponential time.  $\square$

**Theorem 19.** *Deciding whether a finite set of propagation rules has a finite Herbrand model is EXPTIME-complete.*

*Proof.* Algorithm 1 is a deterministic algorithm that decides termination in exponential time. The problem is EXPTIME-hard since this is the case already for anti-Horn propagation rules [2].  $\square$

## 5 Acyclic Herbrand Models

In the previous sections of this paper we were designing algorithms for detecting existence of a finite Herbrand model for a set of restricted clauses  $\mathcal{C}$ . In this section we study the problem of generating such models. The soundness lemma (Lemma 15) suggests the possibility of reconstructing a finite Herbrand model from a set of shortcuts.

The question then arises whether every finite Herbrand model can be reconstructed in this way. The answer is negative. We can reconstruct the models which are obtained by terminating expansion sequences. If a model does not have such a sequence, this can be caused by an atom added to the interpretation in an arbitrary way, when it is *not* required by an unsatisfied clause. Such an atom can contain a term which is not constructed by increasing/decreasing clauses, hence we cannot know what form it has. Thus we cannot reconstruct the model that contains such atoms.

One can see from the proof of completeness (Lemma 16) that given a finite Herbrand model, our algorithms compute a model that is a subset of this one. Hence, the next question is: can we reconstruct all minimal finite Herbrand models? Consider finite Herbrand models  $M_1, M_2$  of a set of clauses  $\mathcal{C}$ . We say that  $M_1$  is smaller than  $M_2$  iff  $M_1 \subseteq M_2$ , and it is minimal if there is no smaller model of  $\mathcal{C}$ . First let us notice that since a set of clauses  $\mathcal{C}$  may contain some non-Horn clauses, there may be multiple minimal finite Herbrand models. In fact there can be an infinite number of them.

**Example 20.** Let  $P(x) \rightarrow P^f(x) \vee Q^g(x)$  be the only flat clause in  $\mathcal{C}$ . Then there are infinitely many minimal Herbrand models of the form

$$\{P(a), P(fa), \dots, P(f^n a), Q(gf^n a)\},$$

where  $n > 0$  and atoms of  $P^f, Q^g$  are omitted.

Hence if we want to output all minimal finite Herbrand models, the process would not terminate in some cases. An obvious idea to reduce the number of computed

models would be to restrict ourselves to those models containing atoms with terms of minimal height. But this would leave out some models which can be of interest, as illustrated in the following example.

**Example 21.** Let the set  $\mathcal{C}$  contain the following flat clauses:

$$A(x) \rightarrow R^f(x) \vee Q^g(x), \quad R(x) \rightarrow S^h(x)$$

Then a finite Herbrand model of minimal height will contain  $\{A(a), Q(ga)\}$ . But another one, containing  $\{A(a), R(fa), S(hfa)\}$  is not of minimal height, although it represents an alternative way of satisfying the clauses.

Example 20 shows that an infinite number of finite models is generated because of some cycle in the process of constructing terms in atoms. We would like to avoid computing such models, since they just repeat constructing new terms following the same pattern. For that we define *acyclic* Herbrand models.

**Definition 22.** Let  $(\mathcal{X}_1, \mathcal{Y}_1)$  be a shortcut. Hence for each  $f \in \mathbf{F}$ , such that  $\mathcal{Y}_1^{-f}$  is not empty, there is a shortcut  $(\mathcal{X}_2, \mathcal{Y}_2)$  such that  $\mathcal{X}_2 = \mathcal{Y}_1^{-f}$  and  $\mathcal{Y}_2^{+f} \subseteq \mathcal{Y}_1$ . In this case, we say that  $(\mathcal{X}_1, \mathcal{Y}_1)$  *requires (depends on)*  $(\mathcal{X}_2, \mathcal{Y}_2)$ , denoted by  $(\mathcal{X}_1, \mathcal{Y}_1) \xrightarrow{f} (\mathcal{X}_2, \mathcal{Y}_2)$ . We call this relation the *dependency relation* between shortcuts.

In Section 3 we describe how to extract shortcuts of height  $n$  from a terminating expansion sequence. For each such sequence  $\mathcal{H}_0 \rightarrow \dots \rightarrow \mathcal{H}_m$ , we now define a *dependency tree* that is composed of dependency relations between shortcuts. There must be an initial shortcut of the form  $(\{A\}, \mathcal{Y})$  that corresponds to the whole sequence and will be the root of this tree. Then, for every  $f \in \mathbf{F}$  for which  $\mathcal{Y}^{-f}$  is not empty, there is a unique shortcut  $(\mathcal{X}_f, \mathcal{Y}_f)$  that  $(\{A\}, \mathcal{Y})$  depends on and corresponds to a subsequence of  $\mathcal{H}_0 \rightarrow \dots \rightarrow \mathcal{H}_m$ . We thus add the edges  $(\{A\}, \mathcal{Y}) \xrightarrow{f} (\mathcal{X}_f, \mathcal{Y}_f)$  to the tree. Continuing this process, we obtain a finite dependency tree that describes the dependency relations between the shortcuts used in the sequence.

Notice that for a given shortcut there may be multiple expansion sequences, and hence there can be different dependency trees with the same root. Given a terminating expansion sequence, we say that this sequence is *acyclic* if the dependency tree of shortcuts extracted from this sequence is *acyclic*, i.e. there is no repetition of shortcuts on any branch of the tree. A finite Herbrand model is called *acyclic* if there is an acyclic terminating expansion sequence for this model.

**Example 23.** In Example 8, we show how to extract a shortcut  $(\{P\}, \{P, Q_2\})$  of height 0 from an expansion sequence. This shortcut is required by the shortcut

$(\{A\}, \{A.P^f, Q_2^f, Q_1\})$  of height 1. To be more precise,  $(\{A\}, \{A.P^f, Q_2^f, Q_1\})$  requires the shortcut  $(\{P, Q_2\}, \{P, Q_2\})$ .<sup>10</sup>

Hence, from that example, we can extract a small dependency tree:

$$(\{A\}, \{A.P^f, Q_2^f, Q_1\}) \xrightarrow{f} (\{P, Q_2\}, \{P, Q_2\})$$

From any acyclic dependency tree rooted in a shortcut  $(\{A\}, \mathcal{Y})$ , we can easily define an acyclic Herbrand model  $\mathcal{H}$  in the following way. For each shortcut  $(\mathcal{X}, \mathcal{Y})$  in the tree, the path from the root uniquely defines a term  $t$ , and we add all atoms  $P(t)$  to  $\mathcal{H}$  where  $P \in \mathcal{Y}$ . For a given set of propagation rules  $\mathcal{C}$ , we can thus compute all possible acyclic Herbrand models, by first computing all shortcuts using Algorithm 1, and from these all acyclic dependency trees.

The above constructions justify the following theorem.

**Theorem 24.** *If a set of propagation rules  $\mathcal{C}$  has a finite Herbrand model, then it has an acyclic Herbrand model, and there are at most triple exponentially many acyclic Herbrand models of  $\mathcal{C}$ .*

*Proof.* Since there are at most exponentially many shortcuts, any path in the acyclic dependency tree is at most exponentially long. Hence all terms in an acyclic Herbrand model are at most exponentially long, too. There are at most double exponentially many such terms. An interpretation of a predicate is a subset of these terms. There are at most triple exponentially many possible interpretations of a predicate, and there are at most polynomially many predicates. Hence there are at most triple exponentially many acyclic Herbrand models.  $\square$

**Example 25.** Let us consider Example 20 again. We have three acyclic dependency trees for this  $\mathcal{C}$ :

1.  $(\{A\}, \{A, P, Q^g\}) \xrightarrow{g} (\{Q\}, \{Q\})$   
corresponding to the acyclic Herbrand model  $\{A(a), P(a), Q^g(a), Q(ga)\}$ .
2.  $(\{A\}, \{A, P, P^f\}) \xrightarrow{f} (\{P\}, \{P, Q^g\}) \xrightarrow{g} (\{Q\}, \{Q\})$   
describing the model  $\{A(a), P(a), P^f(a), P(fa), Q^g(fa), Q(gfa)\}$ .
3.  $(\{A\}, \{A, P, P^f\}) \xrightarrow{f} (\{P\}, \{P, P^f\}) \xrightarrow{f} (\{P\}, \{P, Q^g\}) \xrightarrow{g} (\{Q\}, \{Q\})$   
with the model  
 $\{A(a), P(a), P^f(a), P(fa), P^f(fa), P(ffa), Q^g(ffa), Q(gffa)\}$ .

<sup>10</sup>In computing the set of possibilities, the shortcut  $(\{P\}, \{P, Q_2\})$  is used first, and then the shortcut  $(\{P, Q_2\}, \{P, Q_2\})$ .

## 6 Conclusions

We have developed a method to decide the existence of a finite Herbrand model of a set of propagation rules in exponential time. Moreover, we have demonstrated that finite acyclic Herbrand models may be of double exponential size if one actually wants to construct them. In future work, it would be interesting to extend these results to other classes of formulas, e.g. using multiple constants and variables. We also want to study the applications of our algorithm for the analysis of termination in logic programming and acyclicity for existential rules.

## References

- [1] Franz Baader and Paliath Narendran. Unification of concept terms in description logics. *Journal of Symbolic Computation*, 31(3):277–305, 2001. doi:10.1006/jscs.2000.0426.
- [2] Stefan Borgwardt and Barbara Morawska. Finding finite Herbrand models. In Nikolaj Bjørner and Andrei Voronkov, editors, *Proc. of the 18th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-18)*, volume 7180 of *Lecture Notes in Computer Science*, pages 138–152. Springer, 2012. URL [https://doi.org/10.1007/978-3-642-28717-6\\_13](https://doi.org/10.1007/978-3-642-28717-6_13).
- [3] Stefan Borgwardt and Barbara Morawska. Finite Herbrand models for restricted first-order clauses. LTCS-Report 19-07, Chair for Automata Theory, Institute for Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, 2019. URL <https://tu-dresden.de/inf/lat/reports/>.
- [4] François Bry and Adnan Yahya. Positive unit hyperresolution tableaux and their application to minimal model generation. *Journal of Automated Reasoning*, 25(1):35–82, 2000. doi:10.1023/A:1006291616338.
- [5] R. Caferra, A. Leitsch, and N. Peltier. *Automated Model Building*, volume 31 of *Applied Logic Series*. Springer, Dordrecht, 2004. doi:<https://doi.org/10.1007/978-1-4020-2653-9>.
- [6] Marco Calautti, Sergio Greco, Francesca Spezzano, and Irina Trubitsyna. Checking termination of bottom-up evaluation of logic programs with function symbols. *Theory and Practice of Logic Programming*, 15(6):854–889, 2015. doi:10.1017/S1471068414000623.
- [7] David Carral, Irina Dragoste, and Markus Krötzsch. Restricted chase (non)termination for existential rules with disjunctions. In *Proc. of the 26th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 922–928, 2017. doi:10.24963/ijcai.2017/128.
- [8] Burton Dreben and Warren D Goldfarb. *The decision problem : solvable classes of quantificational formulas*. Reading, Mass : Addison-Wesley, Advanced Book Program, 1979. ISBN 978-0-201-02540-8. Includes indexes.
- [9] Lilia Georgieva, Ullrich Hustadt, and Renate A. Schmidt. Hyperresolution for guarded formulae. *Journal of Symbolic Computation*, 36(1-2):163–192, 2003. doi:10.1016/S0747-7171(03)00034-8.
- [10] Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. Logic programming with function symbols: Checking termination of bottom-up evaluation through program adornments. *Theory and Practice of Logic Programming*, 13(4-5):737–752, 2013. doi:10.1017/S147106841300046X.

- [11] Alexander Leitsch. *The Resolution Calculus*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1997. ISBN 978-3-540-61882-9. doi:10.1007/978-3-642-60605-2.
- [12] Andrew Reynolds, Cesare Tinelli, and Clark Barrett. Constraint solving for finite model finding in SMT solvers. *Theory and Practice of Logic Programming*, 17(4):516–558, 2017. doi:10.1017/S1471068417000175.
- [13] Heng Zhang, Yan Zhang, and Jia-Huai You. Existential rule languages with finite chase: Complexity and expressiveness. In Blai Bonet and Sven Koenig, editors, *Proc. of the 29th AAAI Conf. on Artificial Intelligence (AAAI)*, pages 1678–1685. AAAI Press, 2015. URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9579>.