Decision Procedures and Model Building in Equational Clause Logic

CHRISTIAN G. FERMÜLLER, *Technische Universität Wien, Institut für Computersprachen 185.2, Resselgasse 3/1, A-1040 Vienna, Austria. E-mail: chrisf@logic.at*

ALEXANDER LEITSCH, *Technische Universität Wien, Institut für Computersprachen 185.2, Resselgasse 3/1, A-1040 Vienna, Austria. E-mail: leitsch@logic.at*

Abstract

It is shown that a combination of semantic resolution and ordered paramodulation provides a decision procedure for a large class $\mathrm{PVD}_g^=$ of clause sets with equality. It is also demonstrated how the inference system can be transformed into an algorithm that extracts finite descriptions of Herbrand models from sets of clauses. This algorithm always terminates on clause sets in $\mathrm{PVD}_g^=$ and yields an appropriate model representation. Moreover, an algorithm for evaluating arbitrary clauses over the represented models is defined. Finally, it is proved that $\mathrm{PVD}_g^=$ enjoys the finite model property and shown how finite models can be algorithmically extracted from model representations of this type.[1]

Keywords: model building, resolution, paramodulation, decision procedures

1 Introduction

Traditionally Automated Deduction focuses on algorithms for the detection of unsatisfiability (or, equivalently, validity) of sets of clauses (first-order formulas). However, it has been demonstrated that resolution methods can also profitably be employed as decision procedures for a wide range of classes of clause sets; see [10] for a monograph on this approach to the decision problem. Building on results about hyperresolution as decision procedure we have shown in [8] and [9] that one can effectively extract descriptions of models from satisfiable clause sets that are finitely saturated with respect to a certain hyperresolution operator. The rather young field of "Automated Model Building" promises a high potential in applications and has already been successfully explored in various different forms by R. Caferra and his co-researchers in Grenoble (e.g., [5], [4]), J. Slaney in Canberra (e.g., [18, 19]), T. Tammet (e.g., [16, 15]) in Göteborg, and our group in Vienna. (For an earlier approach to the use of models in Automated Deduction see [20].)

All of the deduction based approaches to Automated Model Building mentioned above deal with classical first-order clause logic without equality. Proper and efficient treatment of equality literals is a well known challenge in Automated Deduction in general. Here we want to extend the decidability results and model building proce-

[1] Full version of a contributed paper presented at the *3rd Workshop on Logic, Language, Information and Computation* (*WoLLIC'96*, http://www.di.ufpe.br/simposios/wollic/wollic96.html), May 8–10, Salvador (Bahia), Brazil, organised by UFPE and UFBA, and sponsored by IGPL, FoLLI, and ASL.

dures of [9] to clause logic with equality. We show (in Section 3) that the class PVD_+, which has been demonstrated to be decidable via hyperresolution in [9], becomes undecidable if the syntax is enriched by the equality predicate. However, in Section 4, we prove that semantic resolution combined with ordered paramodulation provides a decision procedure for the closely related class $\mathrm{PVD}_g^=$ of clause sets. In Section 5 we describe how (generally infinite) term models of such clause sets can be represented by a finite number of (equational and non-equational) atoms. In Section 6 we transform the inference mechanism into a model building procedure. This procedure is correct on all sets of clauses that are deductively closed under positive resolution and ordered paramodulation (with respect to any complete simplification ordering) and where all positive clauses are disconnected (i.e., different literals in a positive clause don't share variables). The algorithm is proved to terminate on all inputs from the class $\mathrm{PVD}_g^=$. In Section 7 we show how to evaluate arbitrary clauses effectively over models whose representations can be constructed as described in Section 6. Finally we prove in Section 8 that class $\mathrm{PVD}_g^=$ also enjoys the finite model property and, more importantly, allows for the efficient construction of finite models from atomic representations of term models.

2 Basic notions

We assume the reader to be familiar with clause logic and the concept of resolution. In particular, *terms*, *atoms*, and *literals* are defined as usual. By an *expression* we mean a term, atom or literal. *Clauses* are finite sets of literals. A *clause set* is a finite set of clauses. An expression or clause is called *ground* if no variables occur in it. The set of all positive, i.e. unnegated, literals of a clause C is denoted by C_+. Similarly, C_- denotes the set of negative literals in C. A clause C with $C = C_+$ is called *positive*. For sets of clauses $\mathcal{S}$, the set of positive clauses in $\mathcal{S}$ is denoted by $\mathcal{S}_+$. The *empty clause* (representing contradiction) is denoted by $\Box$. We deal with equational clause logic; thus the syntax includes a special binary predicate symbol "$\doteq$" for which we employ infix notation as usual. Atoms based on this predicate are called *equalities*. We treat equalities modulo commutativity; i.e., $s \doteq t$ is not distinguished from $t \doteq s$. The negation of $s \doteq t$ is denoted by $s \not\doteq t$.

Given a set $\mathcal{S}$ of clauses or expressions we call the triple $\langle PS(\mathcal{S}), FS(\mathcal{S}), CS(\mathcal{S})\rangle$ the *signature* $\Sigma_{\mathcal{S}}$ of $\mathcal{S}$, where $PS(\mathcal{S})$, $FS(\mathcal{S})$, and $CS(\mathcal{S})$ are the sets of predicate, function and constant symbols, respectively, that occur in $\mathcal{S}$.

The *Herbrand universe* $\mathbf{HU}(\Sigma)$ of a signature $\Sigma = \langle PS, FS, CS\rangle$ is the set of all ground terms built up from $FS \cup CS$, augmented by a special constant symbol if CS is empty. We write $\mathbf{HU}(\mathcal{S})$ for $\mathbf{HU}(\Sigma_{\mathcal{S}})$. Gatoms_Σ is the set of ground atoms over a signature $\Sigma = \langle PS, FS, CS\rangle$, i.e. the set of all atoms $P(t_1, \ldots, t_n)$ where $P \in PS$ and $t_i \in \mathbf{HU}(\Sigma)$, for $1 \le i \le n$.

By $\mathrm{var}(E)$ we denote the set of variables occurring in an expression or set of expressions E. A *substitution* σ is a mapping from the set of variable symbols into the set of terms s.t. $\sigma(x) = x$ almost everywhere. The set $dom(\sigma) = \{x \mid \sigma(x) \neq x\}$ is called the *domain* of σ; $rg(\sigma) = \{\sigma(x) \mid x \in dom(\sigma)\}$ is the *range* of σ. The result of applying a substitution σ to an expression E is denoted by $E\sigma$. It is defined by $a\sigma = a$ for $a \in CS$; $x\sigma = \sigma(x)$ for variables x; and $F(t_1, \ldots, t_n) = F(t_1\sigma, \ldots, t_n\sigma)$ for $F \in FS$ or $F \in PS$, possibly preceded by a negation sign. For sets of expressions C

we define $C\sigma = \{E\sigma \mid E \in C\}$. A *most general unifier (mgu)* of expressions is defined as usual.

A clause C *subsumes* a clause D – we write: $C \leq_{sub} D$ – if $C\theta \subseteq D$ for some substitution θ. This relation is extended to sets of clauses by defining $\mathcal{S} \leq_{sub} \mathcal{S}'$ if for all $D \in \mathcal{S}'$ there exists a $C \in \mathcal{S}$ s.t. $C \leq_{sub} D$. We write $C <_{sub} D$ if $C \leq_{sub} D$ but $D \not\leq_{sub} C$. A minimal subset C' of C that is subsumed by C is called *condensation* $\mathrm{cond}(C)$ of C. Like mgus, condensations are unique up to renaming of variables. (We refer to [7] for details on condensation.)

Terms can be represented as trees; a *position* then represents a node in the tree. $s[n \leftarrow t]$ denotes the expression that arises by replacing the subterm of s at position n by the term t.

The *depth* $\tau(t)$ of a term t is defined as $\tau(t) = 0$ if t is a constant or variable, and $\tau(f(t_1, \ldots, t_n)) = 1 + \max\{\tau(t_i) \mid 1{\leq}i{\leq}n\}$ for a functional term. For an atom or literal we define $\tau(^{(\neg)}P(t_1, \ldots, t_n)) = \max\{\tau(t_i) \mid 1{\leq}i{\leq}n\}$. If C is a set of expressions then $\tau(C)$ is an abbreviation for $\max\{\tau(t) \mid t \in C\}$. The *maximal depth of occurrence* $\tau_{\max}(x, t)$ of a variable x in a term t is defined by $\tau_{\max}(x, x) = 0$ and $\tau_{\max}(x, f(t_1, \ldots, t_n)) = 1 + \max\{\tau_{\max}(x, t_i) \mid x \in \mathrm{var}(t_i), 1{\leq}i{\leq}n\}$. If x does not occur in t then $\tau_{\max}(x, t) = -1$. By $\tau_v(t)$ we denote $\max\{\tau_{\max}(x, t) \mid x \in \mathrm{var}(t)\}$. These definitions are extended to expressions and sets of expressions in the obvious way.

The definitions so far concern only the *syntax* of clause logic. We also have to fix some *semantic* notions:

An *interpretation* for a signature $\Sigma = \langle PS, FS, CS\rangle$ is a pair $\langle D, \varphi\rangle$ where

(1) D, the domain, is a non-empty set,

(2) φ is a signature interpretation; i.e. a function that assigns an element of D to each symbol in CS, a function of type $D^n \rightarrow D$ to each n-ary function symbol in FS, and a function of type $D^n \rightarrow \{\mathbf{true}, \mathbf{false}\}$ to each n-ary predicate symbol in PS.

Each interpretation $\mathcal{I}$ together with a variable assignment d induces an evaluation function $v_{\mathcal{I},d}$ in the usual way s.t. $v_{\mathcal{I},d}(t) \in D$ for terms t and $v_{\mathcal{I},d}(L) \in \{\mathbf{true}, \mathbf{false}\}$ for literals L. Concerning the evaluation of clauses, remember that clauses represent universally closed disjunctions of literals. Let $v_{\mathcal{I},d}$ be extended to ordinary first-order formulas, as usual. We define $v_{\mathcal{I}}(C) = v_{\mathcal{I},d}(\hat{C})$ for $\hat{C} = \forall x_1 \ldots \forall x_k (L_1 \vee \ldots \vee L_n)$, where C is the clause $\{L_1, \ldots, L_n\}$ and $\{x_1, \ldots, x_k\} = \mathrm{var}(C)$. Observe that the variable assignment is irrelevant for the evaluation of clauses and ground expressions. Consequently, we drop the reference to d from the evaluation function in this case.

$\mathcal{I}$ is an *equality interpretation* if, for all ground terms s, t we have: $v_{\mathcal{I}}(s \doteq t) = \mathbf{true}$ iff $v_{\mathcal{I}}(t) = v_{\mathcal{I}}(s)$. However, we are primarily concerned with the construction of descriptions of *equality Herbrand interpretations (EH-interpretations)*, where the domain is a Herbrand universe and "$\doteq$" is interpreted as equivalence relation over ground terms that satisfies the usual substitutional laws.[2]

A clause set $\mathcal{S}$ is *E-satisfiable* if there exists an equality interpretation $\mathcal{I}$ (or, equivalently, an EH-interpretation) such that $v_{\mathcal{I}}(C) = \mathbf{true}$ for all $C \in \mathcal{S}$; $\mathcal{I}$ is called an *E-model* of $\mathcal{S}$. $\mathcal{S}$ is *E-unsatisfiable* if it has no E-model. A corresponding consequence

[2] Of course, the difference is only marginal since every EH-interpretation can be filtrated to an equivalent equality interpretation; and every equality interpretation induces a corresponding EH-interpretation.

relation is defined by: $\mathcal{S} \models \mathcal{S}'$ iff every E-model of $\mathcal{S}$ is also an E-model of the clause set $\mathcal{S}'$. We drop the set parenthesis if $\mathcal{S}'$ is a singleton set. We call a class $\mathcal{C}$ of clause sets *decidable*, if there is an effective procedure, terminating on all $\mathcal{S}$ in $\mathcal{C}$, that tests whether $\mathcal{S}$ is E-satisfiable.

3 Incorporating equality

In [9] we have described how to use hyperresolution to decide and construct descriptions of Herbrand models for all clause sets contained in the following class *without* the equality predicate:

DEFINITION 3.1
PVD_+ is the set of all clause sets $\mathcal{S}$ over signatures without the equality symbol s.t. for all $C \in \mathcal{S}$:

(D) for all $x \in \mathrm{var}(C_+)$: $\tau_{\max}(x, C_+) \leq \tau_{\max}(x, C_-)$.

Observe, that (D) implies

(V) $\mathrm{var}(C_+) \subseteq \mathrm{var}(C_-)$.

This in turn implies that all positive clauses in a clause sets in PVD_+ must be ground[3]

PVD_+ is a subclass of PVD, a class that has been demonstrated to be decidable in [13] and [10]. But there is no loss of generality in investigating PVD_+ instead of PVD, since any decision procedure for PVD can be easily reduced to one for PVD_+ by mapping certain literals into their duals.

It is natural to ask whether the mentioned results of [9] can be extended to clause sets that also contain equality literals.

DEFINITION 3.2
The class defined exactly as PVD_+, but based on a signature that includes the equality symbol, is called $\mathrm{PVD}_+^=$.

PROPOSITION 3.3
Class $\mathrm{PVD}_+^=$ is undecidable.

PROOF. It is well known that word problems for finitely generated algebras, as well as halting problems for Turing machines and other universal models of computation are reducible to the problem of deciding whether a finite set $\mathcal{E}$ of equalities (considered as singleton clauses) augmented by a single negative ground clause is E-satisfiable. In general, $\mathcal{E} \notin \mathrm{PVD}_+^=$. But one obtains a clause set $\mathcal{S}_\mathcal{E} \in \mathrm{PVD}_+^=$, that is logically equivalent to $\mathcal{E}$ by replacing each $\{s \doteq t\} \in \mathcal{E}$ by the clause $\{s \doteq t, t \not\doteq t, s \not\doteq s\}$. Therefore E-satisfiability for $\mathrm{PVD}_+^=$ is undecidable. ∎

Given this undecidability result one can appreciate the following.

DEFINITION 3.4
The class defined exactly as $\mathrm{PVD}_+^=$, except for requiring all equality literals to be ground, is called $\mathrm{PVD}_g^=$.

[3] In [9] we have explicitly included condition (V) in the the definition of PVD_+.

THEOREM 3.5
Class $\mathrm{PVD}_g^=$ is decidable.

In Section 4 we prove this theorem by showing that a refutationally complete proof search procedure always terminates on inputs from $\mathrm{PVD}_g^=$. Obviously, resolution has to be augmented by a derivation rule for the proper treatment of equalities to maintain completeness w.r.t. E-satisfiability. The best investigated rule for this purpose is *paramodulation*. However, a moment of reflection reveals that we have to employ refinements of paramodulation that allow to direct the equalities w.r.t. suitable term orderings if we want to achieve termination of the proof search procedure; even the clause set consisting of the single ground unit clause $\{f(a) \doteq a\}$ allows to derive infinitely many different paramodulants if no ordering restriction is imposed. Such completeness preserving refinements are described, e.g., in [12], where it is also shown that we can combine ordering restrictions (on the application of paramodulation and resolution) with semantic restrictions. The latter are needed here to simulate effects of hyperresolution that are essential in obtaining the results of [9].

More formally we define a set of inference rules that depend on some *complete simplification ordering* (CSO) $\prec$. A partial ordering $\prec$ on the set of expressions over some signature is called a CSO if it satisfies the following conditions:

(O1) $\prec$ is well-founded.

(O2) $\prec$ is total on ground expressions.

(O3) For all expressions v, w and substitutions θ, $v \prec w$ implies $v\theta \prec w\theta$.

(O4) For all terms s, t and expressions w, $s \prec t$ implies $w[s] \prec w[t]$.

(O5) For all terms s, t, a, b with $t \preceq s$, all expressions u and all atoms w,
1. if s is a proper subterm of u, then $s \prec u$,
2. if s is a subterm of w and w is not an equality atom, then $(s \doteq t) \prec w$,
3. if s is a proper subterm of a or b, then $(s \doteq t) \prec (a \doteq b)$,

We do not have to use special properties of CSOs here, but rely on the completeness of the CSO-based inference system of [12]. It turns out that for the purposes of model building (see Section 6), the original derivation rules of [12] are too restrictive. In particular we have to remove the ordering restrictions from resolution and factorization steps. Moreover, we internalize factorization to allow for a proper treatment of the subsumption rule.

DEFINITION 3.6 (factoring)
Let $C = \{L_1, \ldots, L_k\} \cup C'$ ($k \geq 1$) be a clause s.t. $L_1, \ldots, L_k$ are unifiable with mgu σ. Then $C\sigma$ is a *factor* of C. If $k = 1$ then $C\sigma = C$, where σ is the empty substitution. Thus every C is a (trivial) factor of itself.

DEFINITION 3.7 (P-resolution)
Let $C_1' = \{\neg A_1\} \cup D_1$ be a factor of an arbitrary clause C_1 and $C_2' = \{A_2\} \cup D_2$ be a factor of a positive clause C_2 s.t. A_1 and A_2 are unifiable with mgu σ and $\mathrm{var}(C_1) \cap \mathrm{var}(C_2) = \emptyset$. Then $D_1\sigma \cup D_2\sigma$ is a *P-resolvent* of C_1 and C_2. It is called *binary P-resolvent* if the factors C_1' and C_2' are trivial.

The set of all P-resolvents of (pairwise variable disjoint copies of) clauses in a set of clauses $\mathcal{S}$ is denoted by $\mathrm{PR}(\mathcal{S})$.

Definition 3.8 (PO-paramodulation)
Let $C_1' = \{s \dot{=} t\} \cup D_1$ be a factor of a positive clause C_1 and $C_2' = \{L[n \leftarrow r]\} \cup D_2$ be a factor of an arbitrary clause C_2, where r is a subterm of L at position n, but not a variable, and $\mathrm{var}(C_1) \cap \mathrm{var}(C_2) = \emptyset$. Moreover, let σ be an mgu of s and r. If $s\sigma \not\preceq t\sigma$ then $D_1\sigma \cup \{L[n \leftarrow t]\sigma\} \cup D_2\sigma$ is a *PO-paramodulant* of C_1 into C_2.

The set of all PO-paramodulants of (pairwise variable disjoint copies of) clauses in a set of clauses $\mathcal{S}$ is denoted by $\mathrm{OP}_{\prec}(\mathcal{S})$.

The resulting inference system is called *positive resolution with ordered paramodulation* (PROP). Since we identify clauses that are equal up to renaming of variables, $\mathrm{PR}(\mathcal{S})$ and $\mathrm{OP}_{\prec}(\mathcal{S})$ are finite whenever $\mathcal{S}$ is finite.

Definition 3.9
We denote $\mathrm{PR}(\mathcal{S}) \cup \mathrm{OP}_{\prec}(\mathcal{S})$ by $\Pi_{\prec}(\mathcal{S})$. Moreover, we define

$$\Pi^0_{\prec}(\mathcal{S}) = \mathcal{S} \cup \{\{x \dot{=} x\}\}, \text{ and}$$

$$\Pi^i_{\prec}(\mathcal{S}) = \Pi_{\prec}(\Pi^{i-1}_{\prec}(\mathcal{S})) \cup \Pi^{i-1}_{\prec}(\mathcal{S}),$$

for $i \geq 1$. $\Pi^*_{\prec}(\mathcal{S})$ denotes $\bigcup_{i\geq 0} \Pi^i_{\prec}(\mathcal{S})$.

The following completeness result for PROP is a simple corollary to theorem 12 of [12].

Theorem 3.10 (Hsiang and Rusinowitch)
For every CSO $\prec$ and every set of clauses $\mathcal{S}$: $\Box \in \Pi^*_{\prec}(\mathcal{S})$ iff $\mathcal{S}$ is E-unsatisfiable.

The inference system PROP is not yet sufficiently restrictive for the purposes of model building (as will be seen in Section 6). We also have to make use of subsumption.

Definition 3.11
For a clause set $\mathcal{S}$, we denote by $\mathrm{subs}(\mathcal{S})$ a reduction of $\mathcal{S}$ w.r.t. subsumption; i.e. any subset $\mathcal{S}'$ of $\mathcal{S}$ s.t. each $C \in \mathcal{S}$ is subsumed by some $D \in \mathcal{S}'$, but no clause in $\mathcal{S}'$ is subsumed by another clause in $\mathcal{S}'$.

$$\Pi^i_{s,\prec}(\mathcal{S}) = \mathrm{subs}(\Pi^i_{\prec}(\mathcal{S})), \text{ and}$$

$$\Pi^*_{s,\prec}(\mathcal{S}) = \Pi^k_{s,\prec}(\mathcal{S}),$$

where k is the least number s.t. $\Pi^k_{s,\prec}(\mathcal{S}) = \Pi^{k+1}_{s,\prec}(\mathcal{S})$ if such a k exists (otherwise $\Pi^*_{s,\prec}(\mathcal{S})$ remains undefined).

A clause set is called $\Pi_{s,\prec}$*-saturated* if $\mathcal{S} \cup \{\{x \dot{=} x\}\} = \Pi_{s,\prec}(\mathcal{S})$.

Remark 3.12
(1) Reducing a clause set w.r.t. to subsumption usually does not result in a uniquely determined set. subs may be any functional set operator that reduces the set w.r.t. subsumption. I.e., we do not care which particular strategy for application of the deletion rule is applied in implementations of it. This allows us to speak of "the" set $\Pi^*_{s,\prec}(\mathcal{S})$.

(2) Hsiang and Rusinowitch [12] describe a proof system without subsumption. As already observed in [11], the fact that the use of subsumption does not spoil the

completeness of PROP can be easily seen by inspecting the completeness proof in [12].

(3) Observe that $\Pi^*_{s,\prec}(\mathcal{S}) = \{\Box\}$ for any unsatisfiable clause set $\mathcal{S}$, since $\Box$ is derivable and subsumes all clauses.

(4) The decidability theorem 4.2, below, remains valid (and even leads to a more efficient algorithm) if we employ the additional ordering restrictions on resolvents and paramodulants of [12]. However, the central fact for model building as expressed in proposition 6.2 only holds if all P-resolvents are considered.

(5) A final remark on the choice of the underlying inference calculus seems to be appropriate. Other calculi for equational theorem proving, that are even more powerful that of [12], have been introduced in recent years. In particular, [2], [3], and [14] examplarily formulate and investigate a number of important refinements of ordered paramodulation and resolution. Still, we prefer to use the calculus of Hsiang and Rusinowitch in the context of automated model building. The main reason is that in the more modern formalizations mentioned above, equality and non-equality literals are not distinguished explicitly. It is certainly an important discovery to recognize resolution, paramodulation and Knuth-Bendix superposition as different instances of one and the same inference principle. However, if one wants to extend resolution based procedures for the construction of (equality-free) Herbrand models to cases that include (ground) equality literals it seems more appropriate to stick to a "weaker" calculus that presents the rules for the handling of equalities as augmentations to an ordinary resolution calculus. Moreover, we find it convenient to work with the definition of clauses as sets instead of multisets.[4]

4 Decidability of $\mathrm{PVD}_g^=$

It follows from theorem 3.10 that a class $\mathcal{C}$ of clause sets is decidable if there is a simplification ordering $\prec$ s.t. for all $\mathcal{S} \in \mathcal{C}$, $\Pi^*_\prec(\mathcal{S})$ is finite. To prove that $\mathrm{PVD}_g^=$ is decidable in this way, we first show that $\mathrm{PVD}_g^=$ is closed under $\Pi_{s,\prec}$.

LEMMA 4.1
Let $\mathcal{S} \in \mathrm{PVD}_g^=$. Then, for any CSO $\prec$, $\Pi_\prec(\mathcal{S} \cup \{\{x\dot{=}x\}\}) - \{\{x\dot{=}x\}\}$ is in $\mathrm{PVD}_g^=$, too.

PROOF. Observe that the defining condition of $\mathrm{PVD}_g^=$

(D) for all $x \in \mathrm{var}(C_+)$: $\tau_{\max}(x, C_+) \leq \tau_{\max}(x, C_-)$

is stable under the application of substitutions. In particular, any factor $C\theta$ of C still fulfills (D). Since every positive clause in some $\mathcal{S} \in \mathrm{PVD}_g^=$ is ground, the mgu that is used to define a P-resolvent or a PO-paramodulant is a ground substitution. From this it follows that condition (D) cannot be violated by resolution or paramodulation if both parent clauses are in $\mathcal{S}$.

We also have to allow for resolvents using $\{x\dot{=}x\}$ as a parent clause. But resolving a clause C, in which all equality literals are ground, with $\{x\dot{=}x\}$ amounts to removing a ground literal of the form $t\dot{\neq}t$ from C. Thus, also in this case, the defining condition for class $\mathrm{PVD}_g^=$ cannot be violated by the resolvent. ∎

[4] Clauses have to be defined as multisets if one wants to extend CSOs to clauses, as needed for the usual proofs of completeness of superposition calculi. Here we only need CSOs on *expressions*, i.e. terms, atoms and literals.

By definition, $\mathrm{PVD}_g^=$ is closed w.r.t. subsets. Thus $\Pi_{\prec}(\mathcal{S}) \in \mathrm{PVD}_g^=$ implies $\Pi_{s,\prec}(\mathcal{S}) \in \mathrm{PVD}_g^=$.

THEOREM 4.2
For any $\mathcal{S} \in \mathrm{PVD}_g^=$, $\Pi^*_{s,\prec}(\mathcal{S})$ is finite.

Observe that theorem 4.2 together with the completeness of $\Pi_{s,\prec}$ implies theorem 3.5. To decide whether some $\mathcal{S} \in \mathrm{PVD}_g^=$ is E-satisfiable, we only have to compute $\Pi^*_{s,\prec}(\mathcal{S})$ and see whether it equals $\{\Box\}$.

PROOF. (OF THEOREM 4.2). We first show that neither the maximal depth of occurrences of variables nor the number of different variables in a clause can be increased by binary P-resolution or paramodulation. Let E be a binary P-resolvent of a clause C and a positive ground clause D. Since D is ground also the relevant mgu is ground and thus we have $|\operatorname{var}(E)| \leq |\operatorname{var}(C)|$. If the $\mathrm{PVD}_g^=$-condition, $\tau_{\max}(x, C_+) \leq \tau_{\max}(x, C_-)$ for all $x \in \operatorname{var}(C_+)$, holds for C this also implies $\tau_v(E) \leq \tau_v(C)$. The only other possibility for a resolution step to occur is to have $\{x \doteq x\}$ as a parent clause of the resolvent E. As already mentioned in the proof of lemma 4.1, resolving a clause C, in which all negative equality literals are ground with $\{x \doteq x\}$ amounts to removing a ground literal $t \not\doteq t$ from C. Thus $\operatorname{var}(E) = \operatorname{var}(C)$ and $\tau_v(E) \leq \tau_v(C)$. Moreover, we have $\tau(E) \leq \max\{\tau(C), \tau(D)\}$.

On the other hand, if E is a PO-paramodulant of a positive ground clause C into a clause D then, since the used equality atom is ground, we clearly obtain $\operatorname{var}(E)(= \emptyset) \subseteq \operatorname{var}(C)$ and thus $\tau_v(E) \leq \tau_v(C)$, too.

The depth of occurrence of variables may increase through factorization. But, obviously, this can only happen if some variable of the parent clause is replaced by a term not containing this variable. Therefore for any such factor C' of C we have $|\operatorname{var}(C')| < |\operatorname{var}(C)|$. In other words, an increase of τ_v can only occur a finite number of times in any sequence of inference steps. By the above observations on resolvents and paramodulants and the fact that class $\mathrm{PVD}_g^=$ is closed under our inference system (lemma 4.1), it thus follows that there exist some k and ℓ (depending only on $\mathcal{S}$) s.t. $\tau_v(C) \leq k$ and $|\operatorname{var}(C)| \leq \ell$ for all $C \in \Pi^*_{\prec}(\mathcal{S})$.

Finally, observe that there is also a bound on the depth of any term occurring in an equality literal. Since all equalities are ground and the ordering $\prec$ is well founded and total on ground terms, only a finite number of different (ground) equalities can occur in $\Pi^*_{\prec}(\mathcal{S} \cup \{x \doteq x\})$. Therefore a paramodulation step can replace a variable only by one of finitely many different ground terms. Given the bound on τ_v we thus also get a bound on τ for all derivable clauses. (Remember that binary resolution cannot increase the term depth at all, and factorization only finitely often.) Together with the bound on the number of variables this implies that there are only finitely different clauses in $\Pi^*_{\prec}(\mathcal{S})$ for $\mathcal{S} \in \mathrm{PVD}_g^=$.

Thus there exists an $i \in \mathbb{N}$ with $\Pi^{i+1}_{\prec}(\mathcal{S}) = \Pi^i_{\prec}(\mathcal{S})$ and $\Pi^{i+1}_{s,\prec}(\mathcal{S}) = \operatorname{subs}(\Pi^{i+1}_{\prec}(\mathcal{S})) = \operatorname{subs}(\Pi^i_{\prec}(\mathcal{S})) = \Pi^i_{s,\prec}(\mathcal{S})$. By definition 3.11 the set $\Pi^*_{s,\prec}(\mathcal{S})$ is defined and finite. ∎

5 Representations of equational term models

We aim at the automatized construction of computationally simple descriptions of term models. For this purpose we have introduced in [9] the notion of an *atomic* *representation* of an Herbrand model. We now extend this concept to signatures containing the equality symbol.

Definition 5.1
Let $\mathcal{A}$ be a finite set of atoms (possibly including equalities) and H be a Herbrand universe containing $\mathbf{HU}(\mathcal{A})$.

We define the set $\mathrm{TRUE}_H(\mathcal{A}) = \{B \mid B \in \mathrm{Gatoms}_H, \mathcal{A}' \models B\}$ and its complement $\mathrm{FALSE}_H(\mathcal{A}) = \{B \mid B \in \mathrm{Gatoms}_H, \mathcal{A}' \not\models B\}$, where $\mathcal{A}' = \{\{A\} \mid A \in \mathcal{A}\}$.

As is well known, every Herbrand interpretation is uniquely determined by the set of ground atoms that are true in it. Therefore we may define the EH-interpretation $INT_H(\mathcal{A})$ over H by specifying that $v_{INT_H(\mathcal{A})}(B) = \mathbf{true}$ iff $B \in \mathrm{TRUE}_H(\mathcal{A})$. We call $\mathcal{A}$ *EA-representation* – *E*quality *A*tomic representation – of the EH-interpretation $INT_H(\mathcal{A})$.

Differently from the non-equational case it does not suffice to check whether an atom B is an instance of some element of $\mathcal{A}$ in order to test whether B is true in the corresponding model $INT_H(\mathcal{A})$. In general, it is undecidable whether $B \in \mathrm{TRUE}_H(\mathcal{A})$ if $\mathcal{A}$ contains arbitrary equalities. However, we focus on atomic representations where all equalities are ground.

Definition 5.2
A finite set of ground atoms $\mathcal{A}$ (possibly including ground equalities) is called *GEA-representation* – *G*round *E*quality *A*tomic representation) – (w.r.t. a Herbrand universe H containing $\mathbf{HU}(\mathcal{A})$) of the EH-interpretation $INT_H(\mathcal{A})$.

Since a finite set of ground equations can always be extended to a canonical (i.e. terminating and confluent) term rewrite system there is a straightforward method to evaluate ground atoms over a model specified by a GEA-representation. More exactly, let $\mathcal{A} = AT(\mathcal{A}) \cup EQ(\mathcal{A})$ be a GEA-representation w.r.t. some Herbrand universe H, where $AT(\mathcal{A})$ is the set of non-equality atoms and $EQ(\mathcal{A})$ the set of equalities in $\mathcal{A}$. By orienting the members of $EQ(\mathcal{A})$ in any way that respects some fixed well-ordering of H, we obtain a canonical term rewrite system $\mathcal{R}$. Let us write $\nu_{\mathcal{R}}(E)$ for the normal form of a ground atom or ground term E under $\mathcal{R}$. This is extended to sets of atoms $\mathcal{A}$ by defining $\nu_{\mathcal{R}}(\mathcal{A}) = \{\nu_{\mathcal{R}}(A) \mid A \in \mathcal{A}\}$. The evaluation of ground atoms over $INT_H(\mathcal{A})$ is straightforward by the following facts. For any ground non-equality atom A we have

$$A \in \mathrm{TRUE}_H(\mathcal{A}) \ \text{ iff } \ \nu_{\mathcal{R}}(A) \in \nu_{\mathcal{R}}(AT(\mathcal{A})).$$

For ground equalities we have

$$s \dot{=} t \in \mathrm{TRUE}_H(\mathcal{A}) \ \text{ iff } \ \nu_{\mathcal{R}}(s) = \nu_{\mathcal{R}}(t)$$

(see Section 7).

Observe that the presence of equalities allows to represent Herbrand models that cannot be represented by finite sets of arbitrary non-equality atoms, even if we disregard the interpretation of the equality predicate (i.e., if we use equalities only in the representation mechanism, but describe models over signatures without the equality predicate).

Example 5.3
Let Σ be a signature consisting of a monadic predicate symbol P, a monadic function symbol f and a constant a. The Herbrand model $\mathcal{M}$ defined by stipulating that

$$\{P(f^{2n}(a)) \mid n \geq 0\}$$

is the set of ground atoms that are true in $\mathcal{M}$ obviously cannot be characterized as the set of all ground instances of any finite set of atoms. But the GEA-representation

$$\mathcal{A} = \{P(a),\ a \doteq f(f(a))\}$$

clearly represents $\mathcal{M}$ in the sense of Definition 5.2.

On the other hand, GEA-representations are not expressive enough to represent all models that can be described as instances of non-ground atoms. E.g., the set of all ground instances of the atom $Q(x, x)$ over some infinite Herbrand universe cannot be characterized by a GEA-representation.

6 Constructing EA-representations

We outline a method of automated model building based on iterative deductive closure under an operator $\Pi_{s,\prec}$ (for an arbitrary CSO $\prec$) and on clause reduction. For the operator $\Pi_{\prec}$ defined in section 4 the method always terminates on $\mathrm{PVD}_g^=$ and, in case of satisfiability, constructs a GEA-interpretation of a Herbrand model.

Notation 6.1
(G)EA-representations of models are sets of atoms, but the output of the model building algorithm described below is a set of positive unit clauses, i.e. set of singleton sets of atoms. To avoid cumbersome additional notations we shall also consider such sets as (G)EA-representations by identifying the singleton sets with their elements whenever the context is unambiguous.

The following proposition shows that if all positive clauses of some $\Pi_{s,\prec}$-saturated set $\mathcal{S}$ are unit, then they form an EA-representation of a model of $\mathcal{S}$.

Proposition 6.2
Let $\mathcal{S}$ be a finite set of non-positive clauses and let $\mathcal{A}$ be a finite set of positive unit clauses (i.e. equational and non-equational atoms) s.t. $\mathcal{S} \cup \mathcal{A}$ is satisfiable and $\Pi_{s,\prec}$-saturated for a CSO $\prec$. Then $\mathcal{A}$ is an EA-representation of a model of $\mathcal{S} \cup \mathcal{A}$ (over the Herbrand universe $\mathbf{HU}(\mathcal{S} \cup \mathcal{A})$).

Proof. We proceed indirectly. Assume that $INT_H(\mathcal{A})$ falsifies $\mathcal{S} \cup \mathcal{A}$ (where $H = \mathbf{HU}(\mathcal{S} \cup \mathcal{A})$). Then the set $\mathcal{S} \cup \mathcal{A} \cup \{\{\neg B\} \mid B \in \mathrm{FALSE}_H(\mathcal{A})\}$ is E-unsatisfiable. By the compactness theorem there exists a finite subset $\mathcal{F}$ of $\{\{\neg B\} \mid B \in \mathrm{FALSE}_H(\mathcal{A})\}$ s.t. $\mathcal{S} \cup \mathcal{A} \cup \mathcal{F}$ is E-unsatisfiable. By the completeness of $\Pi_{s,\prec}$ we get $\Box \in \Pi^*_{s,\prec}(\mathcal{S} \cup \mathcal{A} \cup \mathcal{F})$. As $\mathcal{S} \cup \mathcal{A}$ is satisfiable and $\Box \notin \mathcal{F}$ there exists some smallest $i > 0$ s.t.

$$\Box \in \Pi^i_{s,\prec}(\mathcal{S} \cup \mathcal{A} \cup \mathcal{F}) - \Pi^{i-1}_{s,\prec}(\mathcal{S} \cup \mathcal{A} \cup \mathcal{F}).$$

By assumption $\mathcal{S} \cup \mathcal{A}$ is $\Pi_{s,\prec}$-saturated and satisfiable, i.e. $\Pi^*_{s,\prec}(\mathcal{S} \cup \mathcal{A}) = \mathcal{S} \cup \mathcal{A} \cup \{x \doteq x\}$. By definition of $\Pi_{\prec}$ no inferences between clauses of $\mathcal{F}$ and $\mathcal{S}$ are possible. All additional clauses in $\Pi^{i-1}_{s,\prec}(\mathcal{S} \cup \mathcal{A} \cup \mathcal{F})$ must be paramodulants from $\mathcal{A} \cup \mathcal{F}$. In particular, there exists a finite set of negative unit clauses $\mathcal{F}'$ (not necessarily ground) s.t. $\mathcal{F}' \subseteq \mathrm{OP}^*_{\prec}(\mathcal{A} \cup \mathcal{F})$ and

$$\Pi^{i-1}_{s,\prec}(\mathcal{S} \cup \mathcal{A} \cup \mathcal{F}) = \mathcal{S} \cup \mathcal{A} \cup \mathcal{F}'.$$

By $\Box \in \Pi^{i}_{s,\prec}(\mathcal{S} \cup \mathcal{A} \cup \mathcal{F})$ there exists an $\{A\} \in \mathcal{A}$ and an $\{\neg B\} \in \mathcal{F}'$ s.t. $\{A\}$ and $\{\neg B\}$ resolve to $\Box$. In other words, there must be a ground substitution γ s.t. $A\gamma = B\gamma$. Observe that $\{\neg B\}$ cannot be resolvable with $\{x \doteq x\}$; since then B were of the form $t \doteq t$. But such equalities cannot be in $\mathrm{FALSE}_H(\mathcal{A})$.

By $\{\neg B\} \in \mathrm{OP}^*_{\prec}(\mathcal{A} \cup \mathcal{F})$ and the correctness of paramodulation we have $\mathcal{A} \cup \mathcal{F} \models \neg B$ and consequently also $\mathcal{A} \cup \mathcal{F} \models \neg B\gamma$. But $\mathcal{F}$ is defined s.t. $\mathcal{A} \cup \mathcal{F}$ is E-satisfiable and thus $\mathcal{A} \cup \mathcal{F} \not\models \{B\gamma\}$. As a consequence we get $\mathcal{A} \not\models B\gamma$ and thus $B\gamma \in \mathrm{FALSE}_H(\mathcal{A})$. But $\{A\} \in \mathcal{A}$ which implies $A\gamma \in \mathrm{TRUE}_H(\mathcal{A})$. On the other hand, $A\gamma = B\gamma$ and therefore $B\gamma \in \mathrm{TRUE}_H(\mathcal{A})$. By this contradiction we conclude that $INT_H(A)$ is a model of $\mathcal{S} \cup \mathcal{A}$. ∎

Remark 6.3

A closer look on the proof of proposition 6.2 reveals that the ordering $\prec$ does not play a role in the arguments. Indeed the proposition holds also for ordinary (i.e. unordered) paramodulation.

If $\mathcal{S}$ is an E-satisfiable Horn set in $\mathrm{PVD}^{=}_{g}$ then $\Pi^*_{s,\prec}(\mathcal{S})$ is finite, saturated and Horn. Therefore $\Pi^*_{s,\prec}(\mathcal{S}) = \mathcal{S}' \cup \mathcal{A} \cup \{x \doteq x\}$ for some finite set of non-positive clauses $\mathcal{S}'$ and a finite set of unit clauses $\mathcal{A}$. By proposition 6.2 $\mathcal{A}$ is an EA-representation of a model of $\Pi^*_{s,\prec}(\mathcal{S})$. Clearly $\mathcal{A}$ also represents an equational model of $\mathcal{S}$ itself.

To summarize, proposition 6.2 shows us that we directly get a model out of a finite set $\Pi_{s,\prec}(\mathcal{S})$ if all positive clauses of this set are unit. If $\Pi^*_{s,\prec}(\mathcal{S})$ also contains positive non-unit clauses we have to reduce $\Pi^*_{s,\prec}(\mathcal{S})$ to "simpler" sets of clauses fulfilling this property. For this purpose we apply a backtracking free splitting method, similar to the one developed in [9]. The method can be applied to any set of clauses which is "positively disconnected".

Definition 6.4

A clause C is called *disconnected* if for all $L, L' \in C$, where $L \neq L'$, $\mathrm{var}(L) \cap \mathrm{var}(L') = \emptyset$.

According to Definition 6.4 all unit clauses and all ground clauses are disconnected. In particular all positive clauses in $\mathrm{PVD}^{=}_{g}$ are ground and thus are disconnected.

Definition 6.5

Let $\prec$ be a complete simplification ordering. A (not necessarily finite) set of clauses $\mathcal{S}$ is in $\mathrm{PDC}_{\prec}$ – *P*ositively *D*isconnected *C*lass – if all positive clauses in $\Pi^*_{\prec}(S)$ are disconnected.

Clearly $\mathrm{PVD}^{=}_{g} \subseteq \mathrm{PDC}_{\prec}$: for all $S \in \mathrm{PVD}^{=}_{g}$ we have $\Pi^*_{\prec}(S) \in \mathrm{PVD}^{=}_{g}$ and all positive clauses are disconnected. Moreover every set of (equational) Horn clauses is positively disconnected w.r.t. any CSO $\prec$; as Horn logic is undecidable we cannot guarantee the finiteness of $\Pi^*_{\prec}(S)$ (which is not required in the definition anyway).

Definition 6.6

Let C be a positive non-unit clause in $\mathcal{S}$ and $A \in C$. Then $(\mathcal{S} - \{C\}) \cup \{\{A\}\}$ is called a *unit reduct* of $\mathcal{S}$ w.r.t. (C, A).

Clearly unit reduction of clause sets $\mathcal{S}$ is incorrect in general (and even for $\mathcal{S} \in \mathrm{PDC}_{\prec}$). I.e., the unit reduct of a satisfiable set may be unsatisfiable. However we will show that unit reduction is sound on $\Pi_{s,\prec}$-saturated sets in $\mathrm{PDC}_{\prec}$ (for any CSO $\prec$). Note that unit reduction is a kind of *one-sided* splitting. For $\mathcal{S} \in \mathrm{PDC}_{\prec}$, $\mathcal{S}$ is satisfiable iff either $(\mathcal{S} - \{C\}) \cup \{\{A\}\}$ or $(\mathcal{S} - \{C\}) \cup \{C - \{A\}\}$ is satisfiable. Instead of considering both alternatives we may always decide for, say, the first one. (This is what we mean by "backtracking-free".) To prove this fact we need a technical notion $C \leq_D E$ expressing that the clause E differs from C "at most by adding D".

DEFINITION 6.7
$C \leq_D E$ if either $\mathrm{cond}(C) = \mathrm{cond}(E)$ or there exists a renaming substitution η s.t. $\mathrm{cond}(C \cup D\eta) = \mathrm{cond}(E)$ and $\mathrm{var}(C) \cap \mathrm{var}(D\eta) = \emptyset$. (For the definition of the condensation operator cond see Section 2.) The relation $\leq_D$ is extended to sets of clauses in the following way: $\mathcal{S}_1 \leq_D \mathcal{S}_2$ if for all $C \in \mathcal{S}_1$ there exists an $E \in \mathcal{S}_2$ s.t. $C \leq_D E$.

EXAMPLE 6.8
Let $C = \{P(x), P(f(y))\}$, $E = \{P(f(y)), Q(z, z)\}$ and $D = \{Q(x, x)\}$. Then $C \leq_D E$: note that $\mathrm{cond}(C) = \{P(f(y))\}$ and choose $\eta = \{x \leftarrow z\}$. For $F = \{P(f(y)), Q(y, y)\}$ we get $C \not\leq_D F$. (There is no renaming η with $\mathrm{var}(C) \cap \mathrm{var}(D\eta) = \emptyset$ and $\mathrm{cond}(C \cup D\eta) = \mathrm{cond}(F)$).

If C is an arbitrary disconnected clause and $A \in C$ then clearly $\{A\} \leq_D C$ for $D = C - \{A\}$.

LEMMA 6.9
Let $\mathcal{S} \in \mathrm{PDC}_{\prec}$ (for an arbitrary CSO $\prec$) and let $\mathcal{S}'$ be a unit reduct of $\mathcal{S}$ w.r.t. (C, A). Then $\Pi^*_{\prec}(\mathcal{S}') \leq_D \Pi^*_{\prec}(\mathcal{S})$ for $D = C - \{A\}$.

PROOF. We show by induction on i that $\Pi^i_{\prec}(\mathcal{S}') \leq_D \Pi^*_{\prec}(\mathcal{S})$.

(IB) $i = 0$:
By definition of a unit reduct $\mathcal{S}' = (\mathcal{S} - \{C\}) \cup \{\{A\}\}$, C must be positive. Since $C \in \mathcal{S} \in \mathrm{PDC}_{\prec}$, C is also disconnected. Clearly $\{A\} \leq_D C$ for $D = C - \{A\}$. $\mathcal{S}' \leq_D \mathcal{S}$ then directly follows from Definition 6.7. Hence also $\mathcal{S}' \leq_D \Pi^*_{\prec}(\mathcal{S})$ by $\mathcal{S} \subseteq \Pi^*_{\prec}(\mathcal{S})$.

(IH) Assume that $\Pi^i_{\prec}(\mathcal{S}') \leq_D \Pi^*_{\prec}(\mathcal{S})$.
It suffices to show that for any $R \in \Pi^{i+1}_{\prec}(\mathcal{S}') - \Pi^i_{\prec}(\mathcal{S}')$: $\{R\} \leq_D \Pi^*_{\prec}(\mathcal{S})$.

Case a: $R \in \mathrm{PR}(\Pi^i_{\prec}(\mathcal{S}'))$.

Then R is resolvent of a positive clause E and some other clause F in $\Pi^i_{\prec}(\mathcal{S}')$. By (IH) there are clauses $E', F' \in \Pi^*_{\prec}(\mathcal{S})$ s.t. $E \leq_D E'$ and $F \leq_D F'$. W.l.o.g., suppose that
$\mathrm{cond}(E') = \mathrm{cond}(E \cup D\eta)$ and $\mathrm{cond}(F') = \mathrm{cond}(F \cup D\theta)$
for appropriate renaming substitutions η and θ with

$$\mathrm{var}(E') \cap \mathrm{var}(F') = \emptyset \quad \text{and} \quad \mathrm{var}(E) \cap \mathrm{var}(D\eta) = \mathrm{var}(F) \cap \mathrm{var}(D\theta) = \emptyset.$$

(The cases were $\mathrm{cond}(E) = \mathrm{cond}(E')$ or $\mathrm{cond}(F) = \mathrm{cond}(F')$ directly follow from this one.)

As $D\eta$ is positive, E' is positive, too. Thus E' and F' are admitted as parent clauses for P-resolution. As $\mathrm{cond}(E) \subseteq \mathrm{cond}(E')$ and $\mathrm{cond}(F) \subseteq \mathrm{cond}(F')$ we may select the same literals for resolution as in E and F (for obtaining R) to obtain a resolvent R'. Since the corresponding mgu does not affect variables occurring in $D\eta \cup D\theta$ we have:

$$\mathrm{cond}(R') = \mathrm{cond}(R \cup (D\eta \cup D\theta)).$$

This implies $R \leq_D R'$ and hence also $\{R\} \leq_D \Pi^*_{\prec}(\mathcal{S})$.

Case b: $R \in \mathrm{OP}_{\prec}(\Pi^i_{\prec}(\mathcal{S}'))$: analogous to case a.

Putting cases a and b together we obtain $\Pi^{i+1}_{\prec}(\mathcal{S}') \leq_D \Pi^*_{\prec}(\mathcal{S})$. ∎

Like in the proof of proposition 6.2 the ordering does not play a role in the proof of the $\leq_D$-property above. Indeed, we see that the same result holds for positive resolution augmented by (unrestricted) paramodulation. We will see later that the ordering only plays a role in showing termination of model building procedures, while their correctness is order-independent.

The following lemma shows the soundness of unit reduction on $\Pi_{s,\prec}$-saturated sets in $\mathrm{PDC}_{\prec}$.

Lemma 6.10
For any CSO $\prec$ let $\mathcal{S}$ be a $\Pi_{s,\prec}$-saturated set in $\mathrm{PDC}_{\prec}$ and $\mathcal{S}'$ be a unit reduct of $\mathcal{S}$ w.r.t. (C, A). Then $\mathcal{S}$ are $\mathcal{S}'$ satisfiability equivalent.

Proof. If $\mathcal{S}$ is unsatisfiable then $\mathcal{S}'$ is unsatisfiable by $\mathcal{S}' \leq_{sub} \mathcal{S}$.

For the other direction, assume that $\mathcal{S}'$ is unsatisfiable. By lemma 6.9 we have $\Pi^*_{\prec}(\mathcal{S}') \leq_D \Pi^*_{\prec}(\mathcal{S})$ for $D = C - \{A\}$. By the completeness of $\Pi_{\prec}$ we get $\Box \in \Pi^*_{\prec}(\mathcal{S}')$ and so $\{\Box\} \leq_D \Pi^*_{\prec}(\mathcal{S})$.

By definition of $\leq_D$ either $\Box \in \Pi^*_{\prec}(\mathcal{S})$ or $D' \in \Pi^*_{\prec}(\mathcal{S})$, s.t. $\mathrm{cond}(D) = \mathrm{cond}(D')$. $\Box \in \Pi^*_{\prec}(\mathcal{S})$ implies the unsatisfiability of $\mathcal{S}$ by the correctness of $\Pi_{\prec}$. Assume that $D' \in \Pi^*_{\prec}(\mathcal{S})$, s.t. $\mathrm{cond}(D) = \mathrm{cond}(D')$. Observe that every clause subsumes its condensation and vice versa. Therefore $D' \leq_{sub} C$. On the other hand, if $C \leq_{sub} D'$ then $\mathcal{S} \leq_{sub} \mathcal{S}'$ and therefore $\mathcal{S}$ is unsatisfiable. Thus the only remaining case is that $D' <_{sub} C$ (i.e. D' *properly* subsumes C).

$\mathcal{S} = \Pi_{s,\prec}(\mathcal{S}) = \Pi^*_{s,\prec}(\mathcal{S}) \leq_{sub} \Pi^*_{\prec}(\mathcal{S})$. Thus there must be a clause $E \in \mathcal{S}$ with

$$E \leq_{sub} D' <_{sub} C.$$

Therefore, $E <_{sub} C$ for $E, C \in \mathcal{S}$. But this contradicts the assumption that $\mathcal{S}'$ is $\Pi_{s,\prec}$-saturated and thus also reduced w.r.t. subsumption. ∎

Example 6.11
$\mathcal{S} = \{\{P(f(a)), Q(a)\},\ \{\neg P(a)\},\ \{f(a) \doteq a\},\ \{P(a), Q(a)\},\ \{Q(a)\}\}$.

$\mathcal{S}$ is $\Pi_{\prec}$-saturated but not $\Pi_{s,\prec}$-saturated. Clearly $\mathcal{S}$ is also satisfiable. Let $\mathcal{S}' = (\mathcal{S} - \{P(f(a)), Q(a)\}) \cup \{P(f(a))\}$. Then $\mathcal{S}'$ is unsatisfiable. On the other hand,

$$\Pi^*_{s,\prec}(\mathcal{S}) = \{\{\neg P(a)\},\ \{f(a) \doteq a\},\ \{Q(a)\}\}.$$

From the last set of clauses we directly read off the EA-representation

$$\mathcal{A} = \{f(a) \doteq a,\ Q(a)\}$$

of a model of $\mathcal{S}$.

Thus we see that subsumption plays a central role in the correctness of unit reduction. Pure (i.e. monotone) deduction operators do not suffice to obtain model representations.

Let $\prec$ be an arbitrary CSO and α be a non-deterministic operator that sends any set of clauses $\mathcal{S}$ into a unit reduct $\alpha(\mathcal{S})$ of $\mathcal{S}$. (We set $\alpha(\mathcal{S}) = \mathcal{S}$ if there is no unit reduct of $\mathcal{S}$). If $\mathcal{S} \in \mathrm{PDC}_\prec$ is satisfiable and $\Pi_{s,\prec}$-saturated then $\alpha(\mathcal{S})$ is satisfiable too, by lemma 6.10. However, $\alpha(\mathcal{S})$ is not necessarily $\Pi_{s,\prec}$-saturated. Thus if we like to iterate unit reduction we have to saturate first. This leads to the following definition:

Definition 6.12
Let α be an operator computing unit reducts and $\prec$ be a CSO. Then the operator T defined by

$$T(\mathcal{S}) = \Pi^*_{s,\prec}(\alpha(\mathcal{S}))$$

is called *unit closure operator* (w.r.t. $(\prec, \alpha)$).

Lemma 6.13
Let $\mathcal{S}$ be a satisfiable set in $\mathrm{PDC}_\prec$ and let T be a unit closure operator w.r.t. $(\prec, \alpha)$. If $\Pi^*_{s,\prec}(\mathcal{S})$ is finite then, for $\mathcal{S}' = T(\Pi^*_{s,\prec}(\mathcal{S}))$, $\mathcal{S}'$ is satisfiable, $\mathcal{S}' \in \mathrm{PDC}_\prec$ and $\mathcal{S}' \models \mathcal{S}$.

Proof. By the correctness of $\Pi_{s,\prec}$ the set $\Pi^*_{s,\prec}(\mathcal{S})$ is satisfiable. By lemma 6.10 $\alpha(\Pi^*_{s,\prec}(\mathcal{S}))$ is satisfiable too and so is its $\Pi_{s,\prec}$-closure $\mathcal{S}'$.

By definition of $\mathrm{PDC}_\prec$, $\Pi^*_\prec(\mathcal{S}) \in \mathrm{PDC}_\prec$ and thus by $\Pi^*_{s,\prec}(\mathcal{S}) \subseteq \Pi^*_\prec(\mathcal{S})$ also $\Pi^*_{s,\prec}(\mathcal{S}) \in \mathrm{PDC}_\prec$. We show now that $\alpha(\Pi^*_{s,\prec}(\mathcal{S})) \in \mathrm{PDC}_\prec$. By lemma 6.9

$$\Pi^*_\prec(\alpha(\Pi^*_{s,\prec}(\mathcal{S}))) \leq_D \Pi^*_\prec(\Pi^*_{s,\prec}(\mathcal{S})) \subseteq \Pi^*_\prec(\mathcal{S}).$$

But D is a positive disconnected clause and, by definition of $\leq_D$,

$$\Pi^*_\prec(\alpha(\Pi^*_{s,\prec}(\mathcal{S}))) \in \mathrm{PDC}_\prec.$$

Clearly $\Pi^*_\prec(T(\Pi^*_{s,\prec}(\mathcal{S}))) \subseteq \Pi^*_\prec(\alpha(\Pi^*_{s,\prec}(\mathcal{S})))$ and so

$$\mathcal{S}' = T(\Pi^*_{s,\prec}(\mathcal{S})) \in \mathrm{PDC}_\prec.$$

Moreover, $\alpha(\Pi^*_{s,\prec}(\mathcal{S}))$ implies $\mathcal{S}$ and therefore $T(\Pi^*_{s,\prec}(\mathcal{S}))$ implies $\mathcal{S}$. ∎

Lemma 6.13 shows that starting with a set $\mathcal{S} \in \mathrm{PDC}_\prec$, computing the $\Pi_{s,\prec}$-closure and then (iteratively) applying the operator T constitutes a correct model building procedure. Of course, the procedure only terminates if the computed saturation $\mathcal{S}'$ is finite and $\alpha(\mathcal{S}') = \mathcal{S}'$.

Definition 6.14
Let $\mathcal{S}$ be a finite saturated set of clauses in $\mathrm{PDC}_\prec$ and T be a unit closure operator defined by $(\prec, \alpha)$. We say that $\langle T^i(\mathcal{S})\rangle_{i\in\mathbb{N}}$ *finitely converges* to a set $\mathcal{S}_*$ if $\mathcal{S}_*$ is finite and there exists an $i \in \mathbb{N}$ s.t. $T^{i+1}(\mathcal{S}) = T^i(\mathcal{S}) = \mathcal{S}_*$.

The above lemmas suggest the following procedure **MBEQ** for constructing a model of a set of clauses in $\mathrm{PDC}_\prec$:

program MBEQ
{**Input:** a finite clause set $\mathcal{S} \in \mathrm{PDC}_{\prec}$ }
{**Output:** EA-representation of a model of $\mathcal{S}$
or **failure** if $\mathcal{S}$ is unsatisfiable}
begin
$\mathcal{S} := \Pi^*_{s,\prec}(\mathcal{S})$;
if $\Box \in \mathcal{S}$ **then return failure** {no model exists};
while $\mathcal{S} \neq T(\mathcal{S})$ **do** $\mathcal{S} := T(\mathcal{S})$;
return $\mathcal{S}_+$
end.

If $T^i(\Pi^*_{s,\prec}(\mathcal{S}))$ is a finite fixed point under T then $T^i(\Pi^*_{s,\prec}(\mathcal{S}))_+$ is an EA-representation of a model of $\mathcal{S}$. On the other hand, already the saturation procedure (computing $\Pi^*_{s,\prec}(\mathcal{S})$) may be non-terminating (if $\mathcal{S}$ is satisfiable and $\Pi^*_{s,\prec}(\mathcal{S})$ is infinite). However, whenever **MBEQ** terminates then the result is either **failure** (i.e. the information that $\mathcal{S}$ is unsatisfiable) or an EA-representation of a model of $\mathcal{S}$.

THEOREM 6.15
MBEQ is correct. More exactly, if **MBEQ** terminates on $\mathcal{S}$ with **failure** then $\mathcal{S}$ is unsatisfiable; if **MBEQ** terminates on $\mathcal{S}$ with some set $\mathcal{D}_+$ then $\mathcal{D}_+$ is an EA-representation of an Herbrand model of $\mathcal{S}$.

PROOF.

Case a: MBEQ terminates on $\mathcal{S}$ with **failure**.
Then $\Box \in \Pi^*_{s,\prec}(\mathcal{S})$ and therefore, by the correctness of $\Pi_{s,\prec}$, $\mathcal{S}$ is unsatisfiable.

Case b: MBEQ terminates and yields a set $\mathcal{D}_+$.
In this case there exists an i s.t. $T^i(\mathcal{S}') = T^{i+1}(\mathcal{S}')$ for $\mathcal{S}' = \Pi^*_{s,\prec}(\mathcal{S})$ and $\mathcal{S}'$ is finite. It follows from lemma 6.13 (by induction on i) that $T^i(\mathcal{S}')$ is in $\mathrm{PDC}_{\prec}$, is satisfiable and $T^i(\mathcal{S}') \models \mathcal{S}$. Note that if **MBEQ** does not terminate with failure then $\Box \notin \Pi^*_{s,\prec}(\mathcal{S})$ and, by the completeness of $\Pi_{s,\prec}$, $\mathcal{S}$ is satisfiable.

If $T^{i+1}(\mathcal{S}') = T^i(\mathcal{S}')$ then $\alpha(T^i(\mathcal{S}')) = T^i(\mathcal{S}')$ and there are no positive clauses in $T^i(\mathcal{S}')$ which are non-unit. For assume that $\alpha(T^i(\mathcal{S}')) \neq T^i(\mathcal{S}')$. Then there exists a non-unit clause $C \in T^i(\mathcal{S}')$ and an atom $A \in C$ s.t. $\alpha(T^i(\mathcal{S}'))$ is a unit reduct of $T^i(\mathcal{S}')$ w.r.t. (C, A) and $\{A\} <_{sub} C$. By definition of $\Pi_{s,\prec}$ there exists a unit clause $B \in T^{i+1}(\mathcal{S}')$ with $\{B\} \leq_{sub} \{A\} <_{sub} C$, what contradicts the saturatedness of $T^{i+1}(\mathcal{S}')$. Therefore we get $\alpha(T^i(\mathcal{S}')) = T^i(\mathcal{S}')$. So $T^i(\mathcal{S}')$ is $\Pi_{s,\prec}$-saturated and all positive clauses in $T^i(\mathcal{S}')$ are unit. By proposition 6.2 $T^i(\mathcal{S}')_+$ is an EA-representation of a Herbrand model of $T^i(\mathcal{S}')$. But $T^i(\mathcal{S}')_+$ implies $\mathcal{S}$ and is just the set $\mathcal{D}_+$ returned by **MBEQ**. ∎

Finally we will show that **MBEQ** always terminates on $\mathrm{PVD}^=_g$. To this aim we have to prove the following lemma.

LEMMA 6.16
Let $\mathcal{S}$ be a satisfiable, $\Pi_{s,\prec}$-saturated set of clauses in $\mathrm{PVD}^=_g$. Then $\langle T^i(\mathcal{S})_{i\in\mathbb{N}}\rangle$ finitely converges to a set $\mathcal{S}_*$.

PROOF. We have to show the existence of an i with $T^{i+1}(\mathcal{S}) = T^i(\mathcal{S})$; then $T^i(\mathcal{S})$ is the desired set $\mathcal{S}_*$.

Like in [9] we first show the existence of a Noetherian order $<$ for sets of clauses s.t. for $\alpha(\mathcal{S}) \neq \mathcal{S}$ we obtain $\Pi^*_\prec(T(\mathcal{S})) < \Pi^*_\prec(\mathcal{S})$ for the monotone operator $\Pi_\prec$. As for saturated $\mathcal{S} \in \mathrm{PVD}^=_g$ $T(\mathcal{S})$ is saturated and in $\mathrm{PVD}^=_g$ too, we may iterate the argument above and obtain a sequence $(\Pi^*_\prec(T^i(\mathcal{S})))_{i\in\mathbb{N}}$ s.t. for all i with $\alpha(T^i(\mathcal{S})) \neq T^i(\mathcal{S})$: $\Pi^*_\prec(T^{i+1}(\mathcal{S})) < \Pi^*_\prec(T^i(\mathcal{S}))$. As $<$ is Noetherian there must be a k s.t. $\alpha(T^k(\mathcal{S})) = T^k(\mathcal{S})$; but that means that all positive clauses in $T^k(\mathcal{S})$ are unit and that $T^k(\mathcal{S})$ is the desired set $\mathcal{S}_*$.

It remains to define the ordering $<$ and to prove $\Pi^*_\prec(T(\mathcal{S})) < \Pi^*_\prec(\mathcal{S})$ for $\alpha(\mathcal{S}) \neq \mathcal{S}$. We define $\mathcal{S} < \mathcal{D}$ iff

(1) $\mathcal{S} \leq_{sub} \mathcal{D}$

(2) For all $E \in \mathcal{S}$ there exists an $F \in \mathcal{D}$ s.t. $E \leq_{sub} F$ and $|E| \leq |F|$.

(3) $\mathcal{D} \not\leq_{sub} \mathcal{S}$.

In [9] it is shown that $<$ is irreflexive, transitive and Noetherian. This property directly carries over to sets of E-clauses, the equality predicate playing no special role in the relation $<$.

Now let $\alpha(\mathcal{S}) = (\mathcal{S} - \{C\}) \cup \{A\}$ for a non-unit positive clause C in $\mathcal{S}$ and an $A \in \mathcal{S}$. We define $D = C - A$ and first prove

$$\Pi^*_\prec(\alpha(\mathcal{S})) < \Pi^*_\prec(\mathcal{S}).$$

Property **(1)** of $<$ is easy to show: $\alpha(\mathcal{S}) \leq_{sub} \mathcal{S}$ by definition of α. Moreover the relation $\leq_{sub}$ is preserved under positive resolution and ordered paramodulation. For paramodulation on $\mathrm{PVD}^=_g$ this property is particularly trivial as all equations are ground.

Property **(3)** $\Pi^*_\prec(\mathcal{S}) \not\leq_{sub} \Pi^*_\prec(\alpha(\mathcal{S}))$:
Let us assume, on the contrary, that $\Pi^*_\prec(\mathcal{S}) \leq_{sub} \Pi^*_\prec(\alpha(\mathcal{S}))$. $\mathcal{S}$ is $\Pi_{s,\prec}$-saturated and so, by $\Pi^*_{s,\prec}(\mathcal{S}) \leq_{sub} \Pi^*_\prec(\mathcal{S})$, $\mathcal{S} \leq_{sub} \Pi^*_\prec(\mathcal{S})$. This, in turn, gives $\mathcal{S} \leq_{sub} \Pi^*_\prec(\alpha(\mathcal{S}))$. As $\{A\} \in \Pi^*_\prec(\alpha(\mathcal{S}))$ we obtain $\mathcal{S} \leq_{sub} \{\{A\}\}$. As $\{A\}$ is a ground unit clause and $\Box \notin \mathcal{S}$ we must have $\{A\} \in \mathcal{S}$. But $A <_{sub} C$ and both $\{A\}$ and C are in $\mathcal{S}$, contradicting the $\Pi_{s,\prec}$-stability of $\mathcal{S}$.

Property **(2)** For all $E \in \Pi^*_\prec(\alpha(\mathcal{S}))$ there exists an $F \in \Pi^*_\prec(\mathcal{S})$ s.t. $E \leq_{sub} F$ and $|E| \leq |F|$:
Lemma 6.9 gives $\Pi^*_\prec(\alpha(\mathcal{S})) \leq_D \Pi^*_\prec(\mathcal{S})$. But $E \leq_D F$ implies $E \leq_{sub} F$ and $|E| \leq |F|$.

This concludes the proof of $\Pi^*_\prec(\alpha(\mathcal{S})) < \Pi^*_\prec(\mathcal{S})$ and it remains to show $\Pi^*_\prec(T(\mathcal{S})) < \Pi^*_\prec(\mathcal{S})$.

By definition of T we have $T(\mathcal{S}) = \Pi^*_{s,\prec}(\alpha(\mathcal{S}))$ and so

(I) $\Pi^*_\prec(T(\mathcal{S})) = \Pi^*_\prec(\Pi^*_{s,\prec}(\alpha(\mathcal{S}))) \subseteq \Pi^*_\prec(\Pi^*_\prec(\alpha(\mathcal{S}))) = \Pi^*_\prec(\alpha(\mathcal{S}))$.

By definition of $\Pi_\prec$ and $\Pi_{s,\prec}$ we also have $\Pi^*_{s,\prec}(\alpha(\mathcal{S})) \leq_{sub} \Pi^*_\prec(\alpha(\mathcal{S}))$. The preservation of $\leq_{sub}$ under $\Pi_\prec$ and the idempotency of $\Pi^*_\prec$ then gives

(II) $\Pi^*_\prec(\Pi^*_{s,\prec}(\alpha(\mathcal{S}))) \leq_{sub} \Pi^*_\prec(\alpha(\mathcal{S}))$.

(I) and **(II)** together yield $\Pi^*_\prec(T(\mathcal{S})) =_{sub} \Pi^*_\prec(\alpha(\mathcal{S}))$ and $\Pi^*_\prec(T(\mathcal{S})) \subseteq \Pi^*_\prec(\alpha(\mathcal{S}))$. Combined with $\Pi^*_\prec(\alpha(\mathcal{S})) < \Pi^*_\prec(\mathcal{S})$ this eventually gives $\Pi^*_\prec(T(\mathcal{S})) < \Pi^*_\prec(\mathcal{S})$. ∎

Combining theorem 6.15 and lemma 6.16 we obtain:

THEOREM 6.17
Let $\mathcal{S}$ be a set of clauses in $\mathrm{PVD}_g^=$. If $\mathcal{S}$ is unsatisfiable then **MBEQ** terminates with **failure**; otherwise it returns an GEA-representation of a model of $\mathcal{S}$.

Note that **MBEQ** does not have to backtrack. The T-procedure is *"don't-care"*-indeterministic. Since the sets $T^k(\mathcal{S})$ are saturated, lemma 6.10 implies that the method works for every selection of non-unit clauses C by α and every selection of an atom A from C.

7 Evaluation of clauses

Truth evaluation of clauses over finite models plays a major role in semantic resolution and model-checking; generally it gives a powerful mean to reduce proof search. Evaluation algorithms over finite models are easy to design, but in case of infinite models the matter becomes substantially more complicated. In [9] we defined an algorithm to evaluate clauses over arbitrary atomic representations (without equality). In this section we develop a method to evaluate clauses without equality over GEA-representations of Herbrand models; as the method uses filtration techniques (making different ground terms equal) it cannot be easily generalized to clauses containing equality. The method is based on normalization in canonical rewrite systems.

Let $\mathcal{A} = \{A_1, \ldots, A_n\} \cup \{s_1 \doteq t_1, \ldots, s_m \doteq t_m\}$ $(= AT(\mathcal{A}) \cup EQ(\mathcal{A}))$ be a GEA-representation and $\mathcal{R}(\mathcal{A})$ be the canonical term rewrite system defined by $EQ(\mathcal{A})$. We write $\nu(A)$ for the normal form of a ground atom A under $\mathcal{R}(\mathcal{A})$. We define $\hat{A} = \{\nu(A_1), \ldots, \nu(A_n)\}$ and $\hat{T}(\mathcal{A}) = \nu(T(\mathcal{A}))$, where $T(\mathcal{A})$ is the set of all subterms occurring in $\mathcal{A}$ and $\nu(..)$ is the obvious extension of ν to sets of terms. An evaluation algorithm for ground atoms over a GEA-representation can easily be obtained via term rewriting.

LEMMA 7.1
Let $\mathcal{A}$ be a GEA-representation of an EH-interpretation $\mathcal{M}$ and B be a ground atom s.t. $H(\mathcal{A} \cup \{B\}) \subseteq H(\mathcal{M})$. Then B is true in $\mathcal{M}$ iff $\nu(B) \in \hat{A}$.

PROOF. B is true in $\mathcal{M}$ iff $\mathcal{A} \cup \{\neg B\}$ is E-unsatisfiable. By the completeness of narrowing (cf. [17]) $\mathcal{A} \cup \{\neg B\}$ is E-unsatisfiable iff $\mathcal{D}:\ \hat{A} \cup \{\neg\nu(B)\}$ is unsatisfiable. Note that here the fully narrowed set coincides with the $\mathcal{R}(\mathcal{A})$-normalized one as all expressions are ground and thus unification becomes matching. But $\mathcal{D}$ is a set of (non-equational) ground atoms and thus $\mathcal{D}$ is unsatisfiable iff $\nu(B) \in \hat{A}$. ∎

An evaluation method for ground clauses can directly be derived from lemma 7.1:

LEMMA 7.2
Let $\mathcal{A}$ be a GEA-interpretation of an EH-interpretation $\mathcal{M}$ and

$$C = \{A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m\}$$

be a ground clause (where the A_i, B_j are atoms) s.t. $H(\mathcal{A} \cup \{C\}) \subseteq H(\mathcal{M})$. Then C is false in $\mathcal{M}$ iff

1. For all $i = 1, \ldots, n : \nu(A_i) \notin \hat{A}$ and
2. For all $j = 1, \ldots, m : \nu(B_j) \in \hat{A}$.

Proof. C is false in $\mathcal{M}$ iff $v_{\mathcal{A}}(A_i) = \mathbf{false}$ and $v_{\mathcal{A}}(B_j) = \mathbf{true}$ for $i = 1, \ldots n$ and $j = 1, \ldots m$. Thus the result directly follows from lemma 7.1. ∎

To compute the truth value of a ground clause over a GEA-representation $\mathcal{A}$ we just construct $\hat{A}$ and test whether the normalized atoms (or their duals) are in $\hat{A}$. The truth evaluation of arbitrary clauses is more difficult and requires some preparatory steps. The key technique consists in reducing the evaluation of a clause C to the evaluation of a finite set of ground instances $\mathcal{S}'$ obtained effectively from C.

Definition 7.3
Let $\mathcal{A}$ be a GEA-representation and H be a Herbrand universe with $H(\mathcal{A}) \subseteq H$. Let

$$\vartheta : \ \{x_1 \leftarrow s_1, \ldots, x_m \leftarrow s_m\} \cup \{y_1 \leftarrow t_1, \ldots, y_n \leftarrow t_n\}$$

be a ground substitution s.t. $\nu(s_i) \in \hat{T}(\mathcal{A})$ and $\nu(t_j) \notin \hat{T}(\mathcal{A})$ for $i = 1, \ldots, m$ and $j = 1, \ldots n$. Then

$$\bar{\vartheta} = \{x_1 \leftarrow \nu(s_1), \ldots, x_m \leftarrow \nu(s_m)\} \cup \{y_1 \leftarrow d, \ldots, y_n \leftarrow d\}$$

where d is a constant symbol not occurring in H.

If the set of domain variables in the substitutions ϑ is kept fixed then the set of all $\bar{\vartheta}$ is always finite (even if the Herbrand universe is infinite). The central property in the design of the evaluation method is that the $\bar{\vartheta}$'s behave like the ϑ's w.r.t. truth evaluations.

Lemma 7.4
Let $\mathcal{A}$ be a GEA-interpretation and H be a Herbrand universe with $H(\mathcal{A}) \subseteq H$. Then for all $\vartheta \in GS(H)$, where $\mathrm{var}(A) \subseteq dom(\vartheta)$: $v_{\mathcal{A}}(A\vartheta) = v_{\mathcal{A}}(A\bar{\vartheta})$.

Proof.

Case a: $v_{\mathcal{A}}(A\vartheta) = \mathbf{true}$.
By lemma 7.1 we get $\nu(A\vartheta) \in \hat{A}$. By definition of $T(\mathcal{A})$, the set of all subterms in $A\vartheta$ is contained in $T(\mathcal{A})$ and so $rg(\vartheta) \subseteq T(\mathcal{A})$; this in turn implies $rg(\bar{\vartheta}) \subseteq \hat{T}(\mathcal{A})$. Particularly ϑ must be of the form $\{x_1 \leftarrow s_1, \ldots, x_m \leftarrow s_m\}$ for $\nu(s_i) \in \hat{T}(\mathcal{A})$ and

$$\bar{\vartheta} = \{x_1 \leftarrow \nu(s_1), \ldots, x_m \leftarrow \nu(s_m)\}.$$

By the confluence of $\mathcal{R}(\mathcal{A})$ we get $\nu(A\vartheta) = \nu(A\bar{\vartheta})$ and so $\nu(A\bar{\vartheta}) \in \hat{A}$. Applying lemma 7.1 once more we obtain $v_{\mathcal{A}}(A\bar{\vartheta}) = \mathbf{true}$.

Case b: $v_{\mathcal{A}}(A\vartheta) = \mathbf{false}$.
Lemma 7.1 gives $\nu(A\vartheta) \notin \hat{A}$.

Case b1: $rg(\vartheta) \not\subseteq T(\mathcal{A})$.
Then we have $\nu(rg(\vartheta)) \not\subseteq \hat{T}(\mathcal{A})$ and

$$\vartheta = \{x_1 \leftarrow s_1, \ldots, x_m \leftarrow s_m\} \cup \{y_1 \leftarrow t_1, \ldots, y_n \leftarrow t_n\}$$

for terms s_i, t_j with $\nu(s_i) \in \hat{T}(\mathcal{A})$ and $\nu(t_j) \notin \hat{T}(\mathcal{A})$ where $n \geq 1$.
By definition of $\bar{\vartheta}$ we obtain $d \in rg(\bar{\vartheta})$ and $d \notin H$. Therefore $A\bar{\vartheta}$ contains d and so $\nu(A\bar{\vartheta}) \notin \hat{A}$. Note that $\mathcal{R}(\mathcal{A})$ does not contain a rule for rewriting terms containing d and thus d does not disappear by normalization. (All terms are ground!) We infer $v_{\mathcal{A}}(A\bar{\vartheta}) = \mathbf{false}$.

Case b2: $rg(\vartheta) \subseteq \hat{T}(\mathcal{A})$.

Then ϑ is of the form $\{x_1 \leftarrow s_1, \ldots, x_m \leftarrow s_m\}$ for $\nu(s_i) \in \hat{T}(\mathcal{A})$. By definition 7.3 $\bar{\vartheta} = \{x_1 \leftarrow \nu(s_1), \ldots x_m \leftarrow \nu(s_m)\}$ and $\nu(A\vartheta) = \nu(A\bar{\vartheta})$ as in case a. This finally yields **false** $= v_{\mathcal{A}}(A\vartheta) = v_{\mathcal{A}}(\nu(A\vartheta)) = v_{\mathcal{A}}(A\bar{\vartheta})$. ∎

The next lemma shows that the additional constant d is "never needed" – provided $\hat{T}(\mathcal{A})$ represents all equivalence classes of the Herbrand universe under the normalization ν. As the Herbrand universe is infinite the corresponding test is not trivial and requires specific techniques. We will see below that a fixed point test w.r.t. the ν-normalized Herbrand term generator G_H^ν does the job.

Definition 7.5

Let H be a Herbrand universe and ν be a normalization operator based on a canonical term rewrite system $\mathcal{R}$. Then

$$G_H^\nu(T) = \nu(T \cup \{f(t_1, \ldots, t_k) | t_1, \ldots t_k \in T, f \in FS_k(H), k \in \mathbb{N}\})$$

for a set of ground terms T (constant symbols are considered as 0-place function symbols).

Lemma 7.6

Let $\mathcal{A}$ be a GEA-representation and H be a Herbrand universe with $H(\mathcal{A}) \subseteq H$. Moreover let us assume that $G_H^\nu(\hat{T}(\mathcal{A})) = \hat{T}(\mathcal{A})$. Then

$$\{\bar{\vartheta} | \vartheta \in GS(H)\} = \{\lambda | \, rg(\lambda) \subseteq \hat{T}(\mathcal{A})\}.$$

Proof. The inclusion $\{\lambda | \, rg(\lambda) \subseteq \hat{T}(\mathcal{A})\} \subseteq \{\bar{\vartheta} | \vartheta \in GS(H)\}$ is trivial as $\hat{T}(\mathcal{A}) \subseteq H$ and $\bar{\vartheta} = \vartheta$ if $rg(\vartheta) \subseteq \hat{T}(\mathcal{A})$.

We prove the other direction by induction on the depth τ of ϑ, i.e. we show for all n:

$$X(n): \quad \text{for all } \vartheta \in GS(H) \text{ and } \tau(\vartheta) \leq n : rg(\bar{\vartheta}) \subseteq \hat{T}(\mathcal{A}).$$

(IB) $n = 0$:

If $\tau(\vartheta) = 0$ then $\vartheta = \{x_1 \leftarrow c_1, \ldots, x_m \leftarrow c_m\}$ for constant symbols $c_1, \ldots, c_m \in H_0$. By $G_H^\nu(\hat{T}(\mathcal{A})) = \hat{T}(\mathcal{A})$ we have $\nu(H_0) \subseteq \hat{T}(\mathcal{A})$. So we obtain $\bar{\vartheta} = \{x_1 \leftarrow \nu(c_1), \ldots, x_m \leftarrow \nu(c_m)\}$ and $\bar{\vartheta} \in \{\lambda | \, rg(\lambda) \subseteq \hat{T}(\mathcal{A})\}$.

(IH) Suppose that X(n) holds.

Let ϑ be a ground substitution with $\tau(\vartheta) = n + 1$.

Then $\vartheta = \{x_1 \leftarrow t_1, \ldots, x_m \leftarrow t_m\} \cup \{z_1 \leftarrow s_1, \ldots, z_p \leftarrow s_p\}$

with $\tau(t_i) \leq n$ and $\tau(s_j) = n + 1$.

Let $s = g(r_1, \ldots, r_k)$ be one of the terms s_j. Then the substitution $\vartheta = \{x_1 \leftarrow r_1, \ldots, x_k \leftarrow r_k\}$ fulfills $\tau(\vartheta) \leq n$ and thus by (IH) $\nu(r_1), \ldots, \nu(r_k) \in \hat{T}(\mathcal{A})$.

From $G_H^\nu(\hat{T}(\mathcal{A})) = \hat{T}(\mathcal{A})$ we infer $\nu(g(\nu(r_1), \ldots, \nu(r_k))) \in \hat{T}(\mathcal{A})$. By the confluence of $\mathcal{R}(\mathcal{A})$ we get

$$\nu(s) = \nu(g(r_1, \ldots, r_k)) = \nu(g(\nu(r_1), \ldots, \nu(r_k)))$$

and so $\nu(s) \in \hat{T}(\mathcal{A})$.

We may apply this argumentation to all terms of depth $n + 1$ contained in ϑ and obtain $rg(\bar{\vartheta}) \subseteq \hat{T}(\mathcal{A})$. This concludes the induction step. ∎

If $\hat{T}(\mathcal{A})$ is not saturated under G^{ν}_{H} then, in general, there are infinitely many equivalence classes of H under ν. But even in this case a similar result holds, where the range is extended by the new constant d.

Lemma 7.7
Let $\mathcal{A}$ be a GEA-representation and H be a Herbrand universe with $H(\mathcal{A}) \subseteq H$. Moreover let us assume that $G^{\nu}_{H}(\hat{T}(\mathcal{A})) \neq \hat{T}(\mathcal{A})$. Then

$$\{\bar{\vartheta} | \vartheta \in GS(H)\} = \{\lambda | \, rg(\lambda) \subseteq \hat{T}(\mathcal{A}) \cup \{d\}\}.$$

Proof. The inclusion $\{\bar{\vartheta} | \vartheta \in GS(H)\} \subseteq \{\lambda | \, rg(\lambda) \subseteq \hat{T}(\mathcal{A}) \cup \{d\}\}$ is trivial by the definition of $\bar{\vartheta}$ from ϑ.

It remains to show the other direction.
Let $\lambda = \{x_1 \leftarrow t_1, \ldots, x_m \leftarrow t_m\} \cup \{y_1 \leftarrow d, \ldots, y_n \leftarrow d\}$ s.t. $t_i \in \hat{T}(\mathcal{A})$.

By $G^{\nu}_{H}(\hat{T}(\mathcal{A})) \neq \hat{T}(\mathcal{A})$ there exist terms $s_1, \ldots, s_k \in \hat{T}(\mathcal{A})$ and a function symbol $g \in FS(H)$ s.t. $\nu(g(s_1, \ldots, s_k)) \notin \hat{T}(\mathcal{A})$. Note that $g(s_1, \ldots, s_k)$ may also denote a constant symbol not contained in $\hat{T}(\mathcal{A})$. Here we define

$$\vartheta = \{x_1 \leftarrow t_1, \ldots x_m \leftarrow t_m\} \cup \{y_1 \leftarrow g(s_1, \ldots, s_k), \ldots y_n \leftarrow g(s_1, \ldots s_k)\}.$$

Clearly $\bar{\vartheta} = \lambda$. ∎

We are in the position now to define a finite set of ground substitutions characterizing the truth value of a clause in a GEA-represented model.

Definition 7.8
Let C be a clause, $\mathcal{A}$ be a GEA-representation and H be a Herbrand universe with $H(\mathcal{A} \cup \{C\}) \subseteq H$.

1. If $G^{\nu}_{H}(\hat{T}(\mathcal{A})) = \hat{T}(\mathcal{A})$ we set
$\Theta(C, \mathcal{A}, H) = \{\lambda | \, dom(\lambda) \subseteq V(C), rg(\lambda) \subseteq \hat{T}(\mathcal{A})\}$.
2. If $G^{\nu}_{H}(\hat{T}(\mathcal{A})) \neq \hat{T}(\mathcal{A})$ then we define
$\Theta(C, \mathcal{A}, H) = \{\lambda | \, dom(\lambda) \subseteq V(C), rg(\lambda) \subseteq \hat{T}(\mathcal{A}) \cup \{d\}\}$.
for a constant symbol d not occurring in H.

Remark 7.9
The set $\Theta(C, \mathcal{A}, H)$ is always finite.

Theorem 7.10
Let C be a clause, $\mathcal{A}$ be a GEA-representation and H be a Herbrand universe with $H(\mathcal{A} \cup \{C\}) \subseteq H$. Then C is true in $INT_H(\mathcal{A})$ iff $v_{\mathcal{A}}(\{C\vartheta | \vartheta \in \Theta(C, \mathcal{A}, H)\}) = \mathbf{true}$.

Proof. **1.** C is true in $INT_H(\mathcal{A})$.

If $G^{\nu}_{H}(\hat{T}(\mathcal{A})) = \hat{T}(\mathcal{A})$ then $\Theta(C, \mathcal{A}, H) \subseteq GS(C, H)$. By definition of the clause semantics $v_{\mathcal{A}}(C\vartheta) = \mathbf{true}$ for all $\vartheta \in GS(C, H)$ and therefore $v_{\mathcal{A}}(\{C\vartheta | \vartheta \in \Theta(C, \mathcal{A}, H)\}) = \mathbf{true}$.
If $G^{\nu}_{H}(\hat{T}(\mathcal{A}) \neq \hat{T}(\mathcal{A})$ then by the lemmas 7.2 and 7.4

$$v_{\mathcal{A}}(C\vartheta) = v_{\mathcal{A}}(C\bar{\vartheta}) \text{ for all } \vartheta \in GS(H).$$

Therefore $v_{\mathcal{A}}(C\bar{\vartheta}) = \mathbf{true}$ for all $\vartheta \in GS(H)$. By lemma 7.7 $\{\bar{\vartheta} | \vartheta \in GS(H)\} = \{\lambda | \, rg(\lambda) \subseteq \hat{T}(\mathcal{A}) \cup \{d\}\}$.

So let $\lambda \in \Theta(C, \mathcal{A}, H)$; then there exists a substituition $\bar{\vartheta}$ s.t. $\bar{\vartheta} = \lambda$. This gives $v_{\mathcal{A}}(C\lambda) = v_{\mathcal{A}}(C\bar{\vartheta}) = v_{\mathcal{A}}(C\vartheta) = \mathbf{true}$ and thus $v_{\mathcal{A}}(\{C\vartheta | \vartheta \in \Theta(C, \mathcal{A}, H)\}) = \mathbf{true}$.

2. $v_{\mathcal{A}}(\{C\vartheta | \vartheta \in \Theta(C, \mathcal{A}, H)\}) = \mathbf{true}$.

By the lemmas 7.6 and 7.7 we obtain

$$\{C\bar{\vartheta} | \, dom(\vartheta) \subseteq V(C), \vartheta \in GS(H)\} = \{C\lambda | \lambda \in \Theta(C, \mathcal{A}, H)\}.$$

By the lemmas 7.1, 7.2 and 7.4 we get $v_{\mathcal{A}}(C\vartheta) = v_{\mathcal{A}}(C\bar{\vartheta})$ for all ground substitutions ϑ. Therefore

$$v_{\mathcal{A}}(\{C\bar{\vartheta} | \vartheta \in GS(C, H)\} = v_{\mathcal{A}}(\{C\lambda | \lambda \in \Theta(C, \mathcal{A}, H)\} = \mathbf{true}.$$

As a consequence $v_{\mathcal{A}}(C\bar{\vartheta}) = \mathbf{true}$ for all $\vartheta \in GS(C, H)$ and so $v_{\mathcal{A}}(C\vartheta) = \mathbf{true}$ for all $\vartheta \in GS(C, H)$. But this implies that C is true in INT_H.

∎

From theorem 7.10 we easily derive the following evaluation algorithm:

{Input is a clause C and a GEA $\mathcal{A}$}

1. Compute $\hat{A}$ and $\hat{T}(\mathcal{A})$;

2. Test the property $G^{\nu}_{H}(\hat{T}(A)) = \hat{T}(A)$;

3. Compute $\Theta(C, \mathcal{A}, H)$ and $\mathcal{D} : \{C\lambda | \lambda \in \Theta(C, \mathcal{A}, H)\}$;

4. Evaluate $\mathcal{D}$ via the method in lemma 7.2.

Example 7.11

Let $\mathcal{A} = \{P(f(f(a))), f(f(a)) \doteq a\}$ and $\mathcal{S} = \{C_1, C_2\}$ for the clauses

$$C_1 = \{P(x), P(f(x))\} \text{ and } C_2 = \{\neg P(y), \neg P(f(y))\}.$$

Furthermore let $H = H(\mathcal{A}) = \{f^n(a) | n \geq 0\}$. We apply our evaluation algorithm to show that $\mathcal{S}$ is true in INT_H:

$T(\mathcal{A}) = \{a, f(a), f(f(a))\}$, $\mathcal{R} = \{f(f(a)) \rightarrow a\}$ and $\hat{T}(\mathcal{A}) = \{a, f(a)\}$, $\hat{A} = \{P(a)\}$. The sets of substitutions for C_1 and C_2 are

$\Theta(C_1, \mathcal{A}, H) = \{\{x \leftarrow a\}, \{x \leftarrow f(a)\}\}$,
$\Theta(C_2, \mathcal{A}, H) = \{\{y \leftarrow a\}, \{y \leftarrow f(a)\}\}$.

Thus for C_1 we obtain the set of ground instances $\mathcal{S}'_1 : \{C'_1, C'_2\}$ for

$\{C'_1 = \{P(a), P(f(a)\}, \; C'_2 = \{P(f(a)), P(f(f(a)))\}$.

As $\nu(P(a)) = \nu(P(f(f(a))) = P(a)$ and $P(a) \in \hat{A}$ we get $v_{\mathcal{A}}(C'_1) = v_{\mathcal{A}}(C'_2) = \mathbf{true}$ and so $\mathcal{S}'_1$ is true in $INT_H(\mathcal{A})$. By theorem 7.10 C_1 is true in $INT_H(\mathcal{A})$.

Similarly it is shown (via $P(f(a)) \notin \hat{A}$) that also C_2 is true in $INT_H(\mathcal{A})$; therefore $INT_H(\mathcal{A})$ is a model of $\mathcal{S}$. Note that there exists no atomic (non-equational) representation $\mathcal{B}$ s.t. $INT_H(\mathcal{B})$ is a model of $\mathcal{S}$.

Now let $H' = \{f^n(t) | t \in \{a, b\}, n \geq 0\}$.
We show that $\mathcal{S}$ is false in $INT_{H'}(\mathcal{A})$:

$\Theta(C_1, \mathcal{A}, H') = \{\{x \leftarrow a\}, \{x \leftarrow f(a)\}, \{x \leftarrow d\}\}$.

But $v_{\mathcal{A}}(\{P(d), P(f(d))\}) = \mathbf{false}$ by $P(d), P(f(d)) \notin \hat{A}$ and so $v_{\mathcal{A}}(C_1) = \mathbf{false}$ and $v_{\mathcal{A}}(\mathcal{S}) = \mathbf{false}$.

8 Finite models

Traditionally (see, e.g., [1]), decidability proofs often proceed by showing that every satisfiable formula of the class in question has a finite model. Since descriptions of all finite interpretations can be enumerated and formulas can be evaluated effectively over finite interpretations, this implies the decidability of the class. However, our proof of the decidability of class $\mathrm{PVD}_g^=$ does not reveal whether $\mathrm{PVD}_g^=$ also has the finite model property. Below we show that this is indeed the case.

We have seen that suitable descriptions of (certain types of) infinite models allow us to evaluate atoms and clauses effectively. The existence of finite models provides an alternative, straightforward approach to clause evaluation and deciding equivalence of model representations. However, this method will only be useful in practice if the models are reasonable small and, most importantly, we do not have to search the space of all (syntactically adequate) finite structures to detect a model. We rather aim at a procedure that allows to extract a description of a finite model of a clause set $\mathcal{S} \in \mathrm{PVD}_g^=$ directly from any GEA-representation of a (generally infinite) EH-model of $\mathcal{S}$ (as delivered by the model building algorithm described in Section 6).

Again, we use $\nu_{\mathcal{R}}(A)$ to denote the normal form of an expression A w.r.t. a canonical term rewrite system $\mathcal{R}$ that corresponds to the set of equalities in a GEA-representation $\mathcal{A}$. Remember that, for a set of expressions $\mathcal{E}$, we write $\nu_{\mathcal{R}}(\mathcal{E})$ for $\{\nu_{\mathcal{R}}(e) \mid e \in \mathcal{E}\}$ and $T(\mathcal{E})$ for the set of all subterms occurring in a member of $\mathcal{E}$. Moreover, let $T^{\neq}(\mathcal{S})$ denote the set of all terms that occur as subterm of an argument of an inequality in some clause in the set of clauses $\mathcal{S}$. We claim that for any finite GEA-representation $\mathcal{A}$ of a model of $\mathcal{S} \in \mathrm{PVD}_g^=$ we can construct a model with domain

$$D(\mathcal{A}, \mathcal{S}) = \nu_{\mathcal{R}}(T(\mathcal{A})) \cup \nu_{\mathcal{R}}(T^{\neq}(\mathcal{S})) \cup \{\xi\},$$

where ξ is a constant not occurring in $\mathcal{S}$ or $\mathcal{A}$. Observe that $T(\mathcal{A})$ and $T^{\neq}(\mathcal{S})$ are finite sets of ground atoms that are closed w.r.t. the subterm relation.

Definition 8.1

Given a GEA-representation $\mathcal{A}$ of a model of $\mathcal{S} \in \mathrm{PVD}_g^=$ let $\mathcal{F}(\mathcal{A}, \mathcal{S}) = \langle D(\mathcal{A}, \mathcal{S}), \phi, d\rangle$ be the equality interpretation where $D(\mathcal{A}, \mathcal{S})$ is as above and the signature interpretation ϕ is defined as follows.

For $c \in CS(\mathcal{S})$

$$\phi(c) = \begin{cases} c & \text{if } c \in D(\mathcal{A}, \mathcal{S}) \\ \xi & \text{otherwise.} \end{cases}$$

For an n-ary $f \in FS(\mathcal{S})$

$$\phi(f)(t_1, \ldots, t_n) = \begin{cases} \nu_{\mathcal{R}}(f(t_1, \ldots, t_n)) & \text{if } \nu_{\mathcal{R}}(f(t_1, \ldots, t_n)) \in D(\mathcal{A}, \mathcal{S}) \\ \xi & \text{otherwise.} \end{cases}$$

For an n-ary $P \in PS(\mathcal{S}) - \{\doteq\}$

$$\phi(P)(t_1, \ldots, t_n) = \begin{cases} \mathbf{true} & \text{if } P(t_1, \ldots, t_n) \in \nu_{\mathcal{R}}(\mathcal{A}) \\ \mathbf{false} & \text{otherwise.} \end{cases}$$

The equality predicate $\doteq$ is interpreted as syntactical identity, i.e. $\phi(t \doteq s) = \mathbf{true}$ iff $t = s$.

Since we are only interested in the evaluation of ground atoms and clauses (corresponding to closed formulas) the variable assignment d is irrelevant.

The following proposition is easily checked by induction on the term depth of t.

PROPOSITION 8.2
For any term $t \in \mathbf{HU}(\mathcal{S})$: $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(t) = t$ if $t \in D(\mathcal{A},\mathcal{S})$ and $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(t) = \xi$ if $\nu_{\mathcal{R}}(t) \notin D(\mathcal{A},\mathcal{S})$.

The interpretation $\mathcal{F}(\mathcal{A},\mathcal{S})$ conincides with the interpretation $\mathcal{I}(\mathcal{A})$ represented by $\mathcal{A}$ on all non-equality atoms. The only difference between the two interpretations is that $\mathcal{F}(\mathcal{A},\mathcal{S})$ projects all ground terms that do not occur in $\mathcal{A}$ or in an inequality of $\mathcal{S}$ into the special element ξ, whereas $\mathcal{I}(\mathcal{A})$ keeps all terms distinct that are not forced to be equal by the equalities in $\mathcal{A}$.

THEOREM 8.3
For all $\mathcal{S} \in \mathrm{PVD}_g^=$, if $\mathcal{A}$ is an GEA-representation of a model $\mathcal{I}(\mathcal{A})$ of $\mathcal{S}$, then $\mathcal{F}(\mathcal{A},\mathcal{S})$ is a model of $\mathcal{S}$, too.

PROOF. Let $C \in \mathcal{S}$. By assumption $v_{\mathcal{I}(\mathcal{A})}(C) = \mathbf{true}$. To check that also $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(C) = \mathbf{true}$ we distinguish three cases.

Case a: $v_{\mathcal{I}(\mathcal{A})}(t\doteq s) = \mathbf{true}$ for some $t\doteq s \in C$. $v_{\mathcal{I}(\mathcal{A})}(t\doteq s) = \mathbf{true}$ implies that $\nu_{\mathcal{R}}(t) = \nu_{\mathcal{R}}(s)$. This means that $\mathcal{F}(\mathcal{A},\mathcal{S})$ is defined such that $\mathcal{I}(\mathcal{A}) \models t\doteq s$ implies $\mathcal{F}(\mathcal{A},\mathcal{S}) \models t\doteq s$. Therefore also $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(C) = \mathbf{true}$.

Case b: $v_{\mathcal{I}(\mathcal{A})}(t\not\doteq s) = \mathbf{true}$ for some $t\not\doteq s \in C$. Since the ground terms s and t occur in inequalities of C, their normal forms $\nu_{\mathcal{R}}(t)$ and $\nu_{\mathcal{R}}(s)$, respectively, are in $D(\mathcal{A},\mathcal{S})$. By proposition 8.2, $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(t) = \nu_{\mathcal{R}}(t)$ and $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(s) = \nu_{\mathcal{R}}(s)$. The underlying term rewrite system $\mathcal{R}$ is based on equalities of $\mathcal{A}$. Therefore $v_{\mathcal{I}(\mathcal{A})}(t\not\doteq s) = \mathbf{true}$ implies $\nu_{\mathcal{R}}(t) \neq \nu_{\mathcal{R}}(s)$. Since $\doteq$ is interpreted as syntactical identity in $\mathcal{F}(\mathcal{A},\mathcal{S})$, we obtain $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(t\not\doteq s) = v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(C) = \mathbf{true}$.

Case c: The only remaining case is that for all ground instances γ, $v_{\mathcal{I}(\mathcal{A})}(L\gamma) = \mathbf{true}$ for some non-equality literal $L\gamma \in C\gamma$. But, by definition of the signature interpretation for non-equality predicates in $\mathcal{F}(\mathcal{A},\mathcal{S})$, $V_{\mathcal{I}(\mathcal{A})}$ and $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}$ agree on all non-equality atoms. (Exactly those ground atoms are true that occur in $\mathcal{A}$.) Therefore also $v_{\mathcal{F}(\mathcal{A},\mathcal{S})}(C) = \mathbf{true}$. ∎

9 Conclusion

We have demonstrated that an inference system combining positive resolution and ordered paramodulation with subsumption not only provides a decision procedure for class $\mathrm{PVD}_g^=$, but also forms the basis for a model building algorithm for this class. We like to think of class $\mathrm{PVD}_g^=$ as a paradigmatic example of the more general topic of inference based model construction. Other fragments of clause logic, which probably require different inference stystems as decision procedures and new mechanisms for representing models, should be studied in this context.

As a final remark, let us point out that a much more refined set of rules could have been used to demonstrate the termination of the proof search procedure on class $\mathrm{PVD}_g^=$. In particular, additional order restrictions on the resolved atom and on the equality used for paramodulation lead to more efficient decision procedures. However, such refinements block the use of the inference system for model building. For the latter purpose it is essential that only semantic, but no order restrictions are imposed on the resolvents. We expect this careful balancing of "restrictiveness" and "productiveness" of the underlying inference system to be an important topic in future investigations of deduction based Automated Model Building.

Acknowledgements

We like to thank the referees for their careful reading of preliminary versions of the paper and very helpful comments.
This work has been partly supported by the Austrian Science Foundation (FWF) grant P11624-MAT.

References

[1] W. Ackermann, *Solvable Cases of the Decision Problem.* Studies in Logic and the Foundation of Mathematics. North-Holland, Amsterdam, 1954.

[2] L. Bachmair and H. Ganzinger, Rewrite-based Equational Theorem Proving with Selection and Simplification. *J. of Logic and Computation 4:217–247, 1994.*

[3] L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder*, Basic Paramodulation.* Information and Computation *121:172–192, 1995.*

[4] R. Caferra and N. Peltier. *Extending semantic resolution via automated model building: applications.* In: *Proceeding of IJCAI'95*, Morgan Kaufman, 1995.

[5] R. Caferra and N. Zabel, Extending Resolution for Model Construction. In: *Logics in AI* (*JELIA '90*). Springer Verlag, LNCS 478 (1991), pp. 153–169.

[6] R. Caferra and N. Zabel, A Method for Simultaneous Search for Refutations and Models by Equational Constraint Solving. *J. Symbolic Computation* 13:613–641, 1992.

[7] G. Gottlob and C.G. Fermüller, Removing redundancy from a clause. *Artificial Intelligence* 61:263–289, 1993, Elsevier Science Publishers.

[8] C.G. Fermüller and A. Leitsch, Model Building by Resolution. In: *Computer Science Logic*, 6^{th} Workshop, CSL'92. San Miniato, Italy, September/October 1992. Springer Verlag, LNCS 702 (1993), pp. 134–148.

[9] C.G. Fermüller and A. Leitsch, Model Building by Hyperresolution. *J. of Logic and Computation* 6:173–203, 1996.

[10] C.G. Fermüller, A. Leitsch, T. Tammet, and N. Zamov, Resolution Methods for the Decision Problem. Springer Verlag, LNAI 679 (1993).

[11] C.G. Fermüller and G. Salzer, Ordered Paramodulation and Resolution as Decision Procedure. In: *Logic Programming and Automated Reasoning*, Springer Verlag, LNCS 698 (1993), pp. 122–133.

[12] J. Hsiang and M. Rusinowitch, Proving Refutational Completeness of Theorem-Proving Strategies: The Transfinite Semantic Tree Method. *J. of the ACM* 38(3):559–58, July 1991.

[13] A. Leitsch, Deciding Clause Classes by Semantic Clash Resolution. *Fundamenta Informaticae* 18:163–182, 1993.

[14] R. Nieuwenhuis and A. Rubio, Theorem Proving with Ordering and Equality Constrained Clauses. *J. Symbolic Computation* 19:321–351, 1995.

[15] T. Tammet, Using Resolution for Deciding Solvable Classes and Building Finite Models. In: *Baltic Computer Science*. Springer Verlag, LNCS 502 (1991), pp. 33–64.

[16] T. Tammet, *Resolution Methods for Decision Problems and Finite Model Building*. Dissertation, Department of Computer Science, Chalmers University of Technology. Chalmers/Göteborg, 1992.

[17] J.R. Slagle, Automated Theorem Proving for Theories with Simplifiers, Commutativity, and Associativity. *J. of the ACM* 21(4):622–642, October 1974.

[18] J. Slaney, *FINDER (Finite Domain Enumerator): Notes and Guide*. Technical report TR-ARP-1/92, Australian National University Automated Reasoning Project, Canberra, 1992.

[19] J. Slaney, SCOTT: A Model-Guided Theorem Prover. In: *Proceedings of the IJCAI 93* (*13th International Joint Conference on Artificial Intelligence*). Ed. Ruzena Bajcsy, Morgan Kaufmann Publishers, Vol. 1, pp. 109–114.

[20] S. Winker, Generation and Verification of Finite Models and Counterexamples Using an Automated Theorem Prover Answering Two Open Questions. *J. of the ACM* 29(2):273–284, April 1982.

Received 26 July 1996. Revised 22 October 1997