3. Jahrestagung der GI-Fachgruppe

Logik in der Informatik

Karlsruhe

31. Mai – 2. Juni 1995

In diesem Internen Bericht der Fakultät für Informatik der Universität Karlsruhe sind die Zusammenfassungen der Vorträge enthalten, die auf der dritten Arbeitstagung der GI-Fachgruppe 0.1.6 *Logik in der Informatik* gehalten wurden. Nach dem konstituierenden Treffen der Fachgruppe 1993 in Leipzig, dem Treffen 1994 in Paderborn fand 1995 die Arbeitstagung vom 31.5. bis 2.6. in Karlsruhe statt.

Was verbirgt sich hinter dem Sammeltitel *Logik in der Informatik*? Kann man anhand der Vorträge in diesem Bericht Schwerpunkte oder Trends feststellen? Diese Frage ist schnell beantwortet: das einzige Schwerpunktthema sind Dynamische Algebren (evolving algebras). Was nicht sehr verwunderlich ist, da zu diesem Komplex eine Sondersektion eingerichtet worden war. Ansonsten ist die thematische Breite des Vortragsprogramms beeindruckend. Natürlich dürfen die klassischen Themen aus der mathematischen Logik, wie Rekursionstheorie und Entscheidbarkeitsfragen aber auch Induktive Logik, nicht fehlen. Daneben ist das Automatische Beweisen vertreten, das einem wohl als erstes in den Sinn kommt, wenn man nach einem Beispiel für Logik in der Informatik gefragt wird. Der Vortrag zu diesem Thema leitet dieses Mal auch gleichzeitig über zu einem weiteren zentralen Beitrag der Logik für die Informatik, nämlich dem der Modellbildung und formalen Beschreibung. Daß Logik in der Informatik nicht bei theoretischen Betrachtungen stehen bleibt wird durch die Arbeiten zur operativen Prozeßführung und zur Plangenerierung belegt. Mit dem letzten Einzelthema, das ich hervorheben möchte, sind wir dann doch bei den Trends angekommen: bei dem Trend, die Bedeutung der visuellen Darstellung in der Vermittlung von Information und Wissen stärker zu betonen — sogar bis hin zur Ausrufung einer neuen Wissenschaft, der Visualistik. Der hier enthaltene Beitrag zu diesem Thema wirkt da doch wesentlich realistischer um nicht zu sagen kühl.

Ich möchte mich an dieser Stelle bedanken bei allen, die zum Gelingen der Arbeitstagung beigetragen haben. Bei der Fakultät für Informatik der Universität Karlsruhe und der Karlsruher Hochschulvereinung e. V. für die finanzielle Unterstützung, bei meinen Mitarbeitern Bernhard Beckert, Reiner Hähnle und Joachim Posegga für die Arbeit während der Vorbereitung und der Durchführung der Tagung, bei Herrn Alexander Leitsch für seine Bereitschaft den eingeladenen Vortrag zu halten.

Peter H. Schmitt, Karlsruhe, April 1995

Inhalt

Hyperresolution and Automated Model Building

A. Leitsch, TU Vienna
(joint work with C. Fermüller)

1 Introduction

Finding and investigating models of abstract structures is at the very heart of mathematical activity. Consequently the value of using models in Automated Deduction is widely admitted. But although mathematical logic has uncovered an impressive body of knowledge about models of first order formulas, very little is known about algorithmical methods for model building. In the area of first order prefix classes, most of the research was directed to prove the *existence* of models, without giving explicit *representations* of models. In particular, the existence of finite models for satisfiable formulas of a class Δ (such classes are called finitely controllable) implies the decidability of Δ. Thus showing finite controllability of classes is a key proof technique in the theory of decidable classes [DG79]. A characteristic of this technique is to compute a recursive bound $\alpha(F)$ for every F in Δ s.t. F is satisfiable iff there exists a model of domain size $\leq \alpha(F)$. The transformation of such a proof into an algorithmical method results in exhaustive search through all finite domain interpretations of size $\leq \alpha(F)$. But, even for small $\alpha(F)$, this is clearly inadequate for practical computing. In more recent time algorithmical finite model building based on theorem proving methods has been investigated by T. Tammet [Tam91],[Tam92], Manthey and Bry [MB88], Caferra & Zabel [CZ91],[CZ93] and J. Slaney [Sla92], [Sla93]. An earlier approach of S. Winker [Win82], although practically relevant and successful, did not define a general algorithmic method.

Tammet's approach, like ours, is based on resolution decision procedures. But his finite model building method applies to the monadic and Ackermann class only and is based on the termination of an ordering refinement. In the resulting model description the interpretation of the function symbols is given completely, but the interpretation of predicate symbols is only partial. Moreover, Tammet uses narrowing and works with equations on the object language level. In our approach the model building is based on termination sets for hyperresolution (which yield other decision classes). The finite model building method presented here is based on the transformation of Herbrand models; it does not use equality reasoning but filtration.

Caferra and Zabel define an (equational) extension of the resolution calculus, specifically designed for the purpose of finite model construction. They give complete representations of Herbrand models but don't specify finite models completely (only the interpretation of the predicate symbols is given). Their method is not based on "postprocessing" of termination sets of resolution but considerably alters the inference system itself. Moreover, the decidable classes are completely different from ours.

Manthey and Bry describe a hyperresolution prover based on a model generation paradigm in [MB88]. Their method of model building, although similar in case of the class PVD_+, differs from ours in several aspects. They essentially use splitting of positive ground clauses and backtracking, features that are avoided in our approach. Moreover, we can handle cases, where no ground facts are produced. While Manthey

and Bry construct Herbrand models only, the final products of our procedure are finite models.

Slaney [Sla92] devised the program *FINDER* that identifies finite models (of reasonably small cardinality) of clause sets whenever they exist. The algorithm is a clever variant of exhaustive search through all possible interpretations and is not referring to resolution or any other inference system. In [Sla93] Slaney investigates the advantages of combining the resolution based theorem prover *OTTER* [McC90] with *FINDER*. Although the methods underlying this approach are quite different form ours its main point—demonstrating the usefulness of model building in Automated Deduction—serves well as motivation also for our work.

We present a fully algorithmical method for the construction of models out of termination sets of positive hyperresolution provers. Even more importantly, we develop a formal framework for model building by resolution based on resolution operators (inluding condensation and subsumption) and orthogonalization. Our key concept is that of a (finite) atomic representation of an Herbrand model. We argue that such atomic representations are suitable and useful as model descriptions by demonstrating that it is decidable whether two given atomic representations are equivalent and that arbitrary clauses can effectively be evaluated w.r.t. the represented models. Motivated by special syntactical properties of the the investigated examples of decidable classes we focus on linear atomic representations, i.e. sets of atoms in which a variable occurs at most once. In particular, we show that linear atomic respresentations can be "orthogonalized". this fact is exploited in order to project the corresponding Herbrand models into finite models.

The basic idea of our approach to automated model building is the following: If the (refutationally complete) theorem prover stops on a set of clauses $\mathcal{C}$ without deriving $\Box$ (contradiction) then $\mathcal{C}$ must be satisfiable. Having obtained the (finite) set $\mathcal{D}$ of derivable clauses, we start to construct a model out of $\mathcal{D}$ (i.e., we use the information provided by positive hyperresolution). The first step of model building consists in transforming the set of derived positive clauses $\mathcal{D}_+$ into an a finite set of unit clauses by iteratively applying hyperresolution operators. $\mathcal{U}$ is shown to be an atomic representation of an Herbrand model of $\mathcal{C}$. In general, this model is infinite since its domain is the Herbrand universe. The second step is of a different flavor: As already mentioned above, we present a method which, for a wide range of classes decidable by hyperresolution, constructs finite models out of the atomic representations generated by hyperresolution. Neither backtracking nor equality reasoning on the object level is required (the latter in contrast to [Tam91], [Tam92] and [CZ91], the former in contrast to [MB88] and [Sla92]).

Although the finite models obtained in this way are not minimal (w.r.t. domain size) in general, the method is clearly superior to exhaustive search, since *no kind of backtracking* is involved. However, it is limited to syntax classes admitting filtration on the termination sets (i.e., Herbrand models can be mapped into finite models under preservation of truth values).

As the method presented here is based on termination of positive hyperresolution with subsumption and condensing, the preprocessing step just consists of "ordinary" theorem proving. Only in case $\Box$ has not been derived, the proper model building algorithm is applied to the termination set. This is also a characteristic of Tammet's

work, while Caferra & Zabel start with a model building procedure at once. Together with Tammet's results on finite model building, this paper can be considered as a starting point for model building methods based on resolution decision procedures for first order classes.

All results mentioned here are presented and proved in [FL95].

2 Hyperresolution as Decision Procedure

All model building operations presented here are based on (positive) hyperresolution. This resolution refinement is used first in a preproccessing manner, in the sense that the model construction itself starts on finite termination sets, i.e. finite sets of clauses that are satisfiable and closed w.r.t. the resolution operator. But also the computation of atomic representations out of the termination sets essentially employs hyperresolution. Termination sets are familiar from resolution decision theory (see [FLTZ93]). In this section we formally define hyperresolution in terms of resolution operators. Moreover we present some classes of clause sets that can be decided by hyperresolution. These classes serve as concrete examples for our model building procedures, to be decribed in the section 3.

Definition 2.1 *Let C, D be condensed clauses, where D is positive. The condensation of a binary resovent of C and a factor of D is called a* PRF-resolvent. *(PRF abbreviates "positive, restricted factoring".)*

Remark. Throughout this paper we assume that clauses always appear in condensed form, mostly without mentioning this fact explicitly. Note that a clause $\mathcal{C}$ is called condensed if it does not contain a nontrivial factor which subsumes $\mathcal{C}$.

Definition 2.2 *Let C be a non-positive clause and let the clauses D_i, for $1 \leq i \leq n$, be positive. Then the sequence $\Gamma = (C; D_1, \ldots, D_n)$ is called a* clash sequence.

Let $C_0 = C$ and C_{i+1} be a PRF-resolvent of C_i and D_{i+1} for $i < n$. If C_n is positive then it is called a clash (or hyper)resolvent *defined by Γ.*

Hyperresolution examplifies the principle of macro inference. It only produces positive clauses or the empty clause $\Box$. In variance to the standard definition of hyperresolution we have included a restriction on factoring. The concept of "semi-factoring" is investigated in [Nol80], where—inter alia—it is shown that positive hyperresolution based on PRF-resolution is complete.

Below, we do not need to refer to hyperresolution deductions themselves but rather are interested in the set of derived clauses. For this purpose the following operator based description of hyperresolution seems most adequate.

Definition 2.3 *Let $\mathcal{C}$ be a set of clauses. By $\rho_H(\mathcal{C})$ we denote the set of all clash resolvents definable by clash sequences of clauses in $\mathcal{C}$. The hyperresolution operator R_H and its closure R_H^* is defined by:*

$$R_H(\mathcal{C}) = \mathcal{C} \cup \rho_H(\mathcal{C}),$$

$$R_H^0(\mathcal{C}) = \mathcal{C} \text{ and } R_H^{i+1}(\mathcal{C}) = R_H(R_H^i(\mathcal{C})) \text{ for } i \leq 0.$$
$$R_H^*(\mathcal{C}) = \bigcup_{i \geq 0} R_H^i(\mathcal{C}).$$

Combining hyperresolution with subsumption we get:

Definition 2.4 *Let $\mathcal{C}$ be a set of clauses and sub be a subsumption reduction operator. Then we have*

$$R_{HS}(\mathcal{C}) = sub(\mathcal{C} \cup \rho_H(\mathcal{C})),$$
$$R_{HS}^0(\mathcal{C}) = sub(\mathcal{C}) \text{ and } R_{HS}^{i+1}(\mathcal{C}) = R_{HS}(R_{HS}^i(\mathcal{C})) \text{ for } i \leq 0.$$
$$R_{HS}^*(\mathcal{C}) = \bigcap_{i \geq 0} \bigcup_{j \geq i} R_H^j(\mathcal{C}).$$

R_{HS}^* enjoys two important properties:

(1) R_{HS} is complete, i.e. $\Box \in R_{HS}^*(\mathcal{C})$ whenever $\mathcal{C}$ is unsatisfiable. Note that, by definition of the subsumption operator, $\Box \in R_{HS}^*(\mathcal{C})$ implies $R_{HS}^*(\mathcal{C}) = \{\Box\}$. (Since $\Box$ subsumes all clauses.)

(2) If there exisits an i s.t. $R_{HS}^i(\mathcal{C}) \leq_{sub} R_{HS}^{i+1}(\mathcal{C})$, i.e. if $R_{HS}^i(\mathcal{C})$ is a fixed point w.r.t. R_{HS}, then $R_{HS}^*(\mathcal{C}) = R_{HS}^i(\mathcal{C})$ and $R_{HS}^*(\mathcal{C})$ is finite. (Of course, $\mathcal{C}$ is assumed to be finite.)

It is well known that hyperresolution remains complete when combined with subsumption. The crucial fact here is that $R_{HS}^*(\mathcal{C}) \leq_{sub} R_H^*(\mathcal{C})$. That condensation preserves completeness is shown in [FLTZ93].

If for all $\mathcal{C}$ in a class of clause sets Γ there exists an i s.t. $R_{HS}^*(\mathcal{C}) = R_{HS}^i(\mathcal{C})$ then the computation of the finite fixed point $R_{HS}^*(\mathcal{C})$ defines a decision procedure for Γ; we say that R_{HS} *decides* Γ. Note that $R_{HS}^*(\mathcal{C}) \subseteq R_H(\mathcal{C})$, and therefore R_{HS} decides Γ whenever R_H decides Γ.

We now define two classes which can be decided by R_{HS}.

Definition 2.5 PVD_+ *is the set of all sets of clauses $\mathcal{C}$ s.t. for all $C \in \mathcal{C}$:*

(1) $V(C_+) \subseteq V(C_-)$, and

(2) for all $x \in V(C_+)$: $\tau_{\max}(x, C_+) \leq \tau_{\max}(x, C_-)$.

(Observe, that (1) implies that all positive clauses in $\mathcal{C}$ must be ground.)

PVD_+ is a subclass of PVD, a class that has been demonstrated to be decidable in [Lei93] and [FLTZ93]. But there is no loss of generality in investigating PVD_+ instead of PVD, since the decision procedure for PVD can be easily reduced to that for PVD_+: A clause set $\mathcal{C}$ is in PVD iff there is a renaming η of the signs of the literals s.t. $\eta(\mathcal{C}) \in \mathrm{PVD}_+$. To get a resolution decision procedure for PVD we first construct an adequate renaming η and then apply R_{HS} to $\eta(\mathcal{C})$. Instead of transforming $\mathcal{C}$ to $\eta(\mathcal{C})$ one may also apply a different semantical setting (see [Lov78]) to $\mathcal{C}$ itself.

PVD_+ can be considered as a generalization of DATALOG to clause forms containing function symbols. The decidability of PVD_+ restricted to Horn was shown in [Lei90].

Definition 2.6 $\mathrm{OCC1N}_+$ *is the set of all sets of clauses* $\mathcal{C}$ *s.t. for all* $C \in \mathcal{C}$*:*

(1) $OCC(x, C_+) = 1$ *for all* $x \in V(C_+)$*, and*

(2) $\tau_{\max}(x, C_+) \leq \tau_{\min}(x, C_-)$ *for all* $x \in V(C_+) \cap V(C_-)$*:*

Like in the case of PVD_+, $\mathcal{C} \in \mathrm{OCC1N}$ iff there exists a renaming η of the signs of the literals s.t. $\eta(\mathcal{C}) \in \mathrm{OCC1N}_+$. OCC1N is shown to be decidable in [FLTZ93]; the decidability of $\mathrm{OCC1N}_+$ restricted to Horn clauses is proven in [Fer90].

PVD_+ is even decidable by positive hyperresolution without condensing (see [Lei93]). However, in deciding $\mathrm{OCC1N}_+$ condensing is essential (but subsumption is not needed, see [FLTZ93]). Clearly, if a resolution procedure terminates (i.e., if only finitely many resolvents are produced) then this remains true if condensing and subsumption is added. Therefore the cited results imply that R_{HS} decides $\mathrm{PVD}_+ \cup \mathrm{OCC1N}_+$. We shall see that the reduction of the clause sets w.r.t. condensing and subsumption is essential to the model building procedure.

In the following sections the sets $R^*_{HS}(\mathcal{C})$ will serve as "raw material" for model building. All results described here hold for the decidable classes PVD_+ and $\mathrm{OCC1N}_+$, but many of them are more general. A wider range of decidable classes (generalizations of PVD) was obtained in [Lei93] using the concept of abstract measures of atom complexity. All these classes are decidable by hyperresolution and admit the finite model building procedure introduced in Section 3.

3 Automated Model Building

Observe that speaking of algorithms that construct models of clause sets is actually a slight misuse of language. The output of a computer program can, of course, only be a formal description or *representation* of a model. As long as we only deal with finite models there is hardly a point in insisting on the difference between the model (as abstract mathematical entity) and its representation (e.g., in form of multiplication tables). However, in dealing with infinite structures the difference gets essential. Here we are interested in Herbrand models (which are infinite in all but trivial cases) und thus have to present a formal language for the representation of these structures. It should be clear that there is no single formalism capable of representing *all* Herbrand models (over a signature that contains at least one function symbol) in a finite manner, since there are non-denumerably many.

Our aim is to specify an algorithm that, given a satisfiable clause set $\mathcal{C}$ in a class that is decidable by hyperresolution as indicated in Section 2, constructs a representation of *some* Herbrand model of $\mathcal{C}$. We shall show that these models allow for a particularily simple and elegant representation in form of finite sets of atoms.

Definition 3.1 *Let* $\mathcal{A}$ *be a finite set of atoms and* H *be an Herbrand universe containing* $H(\mathcal{A})$*. We say that* $\mathcal{A}$ *is an* atomic representation *of the set* $INT_H(\mathcal{A})$ *of all* H*-instances of* $\mathcal{A}$*. Note that we identify Herbrand models with the set of true ground atoms; therefore* $\mathcal{A}$ *may be considered as an atomic representation of the Herbrand model* $INT_H(\mathcal{A})$*. We abbreviate this by saying that* $\mathcal{A}$ *is an ARM (w.r.t. to* H*).*

Observe that whenever only finitely many ground atoms are true in an Herbrand model $\mathcal{I}$ then $\mathcal{I}$ (as a set of atoms) and its atomic representation coincide. However, many more interesting Herbrand models admit atomic representations as well.

Example 3.1 . Let $\mathcal{C} = \{P(f(a)),\ P(x) \vee P(f(x)),\ \neg P(a)\}$. Then $\mathcal{I} = \{P(f(t)) \mid t \in H(\mathcal{C})\}$ is an Herbrand model of $\mathcal{C}$. The set $\mathcal{A} = \{P(f(x))\}$ is obviously an atomic representation of $\mathcal{I}$ w.r.t. $H(\mathcal{C})$, i.e. $\mathcal{I} = INT_{H(\mathcal{C})}(\mathcal{A})$.

Of course, there are (even quite simple) Herbrand models that do not have an atomic representation. E.g., if we augment $\mathcal{C}$ by the clause $\neg P(x) \vee \neg P(f(x))$ then $\mathcal{I}' = \{P(f^{(2n+1)}(a)) \mid n \geq 0\}$ is the only Herbrand model of the new clause set. But $\mathcal{I}'$ cannot be represented by a finite number of atoms in the sense of Definition 3.1.

Whether a formalism is adequate for the representation of models not only depends on which type of models we want to describe, but also on the intended applications. However, even if we abstract from specific applications, there are some features which seem desirable for any type of representation formalism:

- It should be (computationally) easy to decide whether a given ground atom is true or false in the represented model. (E.g., an algorithm that evaluates ground atoms in polynomial time w.r.t. the size of the atom and the repesentation of the model would be adequate.)
- It should be possible to decide whether two given representations specify the same model or not.
- There should be an effective method for the evaluation of arbitrary clauses w.r.t. the represented model. I.e., there should be an algorithm that decides whether a given clause is true or false in the model.

Observe that in the case of finite models the standard representation by multiplication tables almost trivially fulfills these requirements of adequacy. However, as soon as we want to represent infinite models it is not clear at all how to evaluate clauses or how to test the equivalence of representations. For atomic representations of Herbrand models it is clear that arbitrary ground atoms can be evaluated efficiently: Since a ground atom B is true in $INT_H(\mathcal{A})$ iff there is some $A \in \mathcal{A}$ s.t. B is an instance of A, the evaluation consists in a linear number of instance checks und thus is at most of quadratic time complexity. Indeed also the other features from the above list hold. We do not know of any other representation mechanism for reasonable classes of infinite models that shares these desirable properties.

Observe, that the possibility to evaluate effectively arbitrary clauses is a basic requirement if one wants to use model constructing algorithms to find counter models during proof search (e.g., in the frame of model elimination procedures.) Recently, there is a lot of interest in this type of model based search pruning (see e.g. [Sla93]), which is usually employed in a purely heuristic, ad hoc manner and still seems to lack a theoretical foundation. Our results also contribute to this line of research. Another application for model representation mechanisms that allow for the evaluation of clauses (and literals) consists in the possibility to use more sophisticated interpretations than simple "settings" in the context of semantic resolution (see [Lov78]).

The decidable classes introduced in the last section not only allow for the construction of atomic representation of models but also guarantee a very simple term structure of the representing atoms. The resulting representations are called linear.

Definition 3.2 *An expression E is called* linear *if each variable in E occurs only once, i.e. $OCC(x, E) = 1$ for all $x \in V(E)$. A set of expressions is said to be linear if all its elements are linear.*

In [FL95] it is shown that every linear representation can algorithmically be transformed into a finite sets of atoms that represents the same Herbrand model in an "orthogonal" manner and consists of instances of the original atoms. These orthogonal representations can be applied to project the represented Herbrand models into finite models.

W.l.o.g. we speak of terms only in the following definitions. The generalization to atoms is obvious.

Definition 3.3 *Let T be a finite set of non-variable linear terms and H an Herbrand universe containing $H(T)$. T* orthogonally represents *the set of its ground H-instances $G_H(T)$ if for all $s \in G_H(T)$ there is exactly one $t \in T$ s.t. s is an instance of t.*

T is an orthogonal extension *of a set of terms T' (w.r.t. H) if T orthogonally represents $G_H(T')(= G_H(T))$ and if for each $t' \in T'$ there is an H-instance t of t' in T.*

Example 3.2. $T = \{a,\ f(x, f(u, v)),\ f(a, y),\ f(x, a)\}$ represents $G_H(T)$, where H is the set of all terms built up from a and f only. Indeed, $G_H(T) = H$. T is not an orthogonal representation since, e.g., $f(a, f(a, a))$ is an instance of both, $f(x, f(u, v))$ and $f(a, y)$. $T_1 = \{a, f(x, y)\}$ orthogonally represents H but it is not an orthogonal extension of T since, e.g., no instance of $f(a, y)$ occurs in T_1. However, $T_2 = \{a,\ f(x, f(u, v)),\ f(a, a),\ f(f(x, y), a)\}$ is an orthogonal extension of T w.r.t. H.

Remark. Observe that T is an orthogonal representation iff any two different terms in T are not unifiable.

In [FL95] an algorithm is presented that constructs for any finite set T of linear terms an orthogonal extension of T.

Above we have suggested to represent Herbrand models by (finite) sets of atoms. This representation not only seems to be very natural, but also is also close to termination sets of hyperresolution. Observe, that if R_{HS} terminates on a set $\mathcal{C}$ of Horn clauses then it produces an atomic representation of an Herbrand model of $\mathcal{C}$. This is trivially so, since positive hyperresolution only produces positive unit clauses (or $\Box$). Therefore the union over all hyperresolvents (i.e., the derivable positive clauses) constitutes an atomic representation of a (minimal) Herbrand model of $\mathcal{C}$. In this section we extend the method to non-Horn termination sets of hyperresolution.

Definition 3.4 *A set of clauses $\mathcal{C}$ is* (R_{HS}-)stable *if $R_{HS}(\mathcal{C}) = \mathcal{C}$.*

Observe that the deductive closure $R^*_{HS}(\mathcal{C})$ of $\mathcal{C}$ is always stable. Our investigations mainly focus on finite stable sets, i.e., finite fixed points of R_{HS}.

Definition 3.5 *We call $\mathcal{C}$ positively decomposed (and write $\mathcal{C} \in PDC$) iff $R^*_H(\mathcal{C})$ is finite and all clauses in $P(R^*_H(\mathcal{C}))$ are decomposed.*

PDC contains all sets of Horn clauses $\mathcal{C}$ where R^*_H is finite. But also PVD_+ and $\mathrm{OCC1N}_+$ are subsets of PDC and all classes that are shown to be decidable by hyperresolution in [FLTZ93] and [Lei93].

The following lemma provides the key technique for the reduction of a set of positive clauses to an atomic representation. It states that, for stable sets of clauses, positive clauses can be replaced by proper subclauses under preservation of satisfiability.

Lemma 3.1 *Let $\mathcal{C} \in PDC$ s.t. $\mathcal{C}$ is stable and contains a positive non-unit clause D. For any atom P in D we have*

(1) $(\mathcal{C} - \{D\}) \cup \{P\}$ is satisfiable iff $\mathcal{C}$ is satisfiable, and

(2) $(\mathcal{C} - \{D\}) \cup \{P\}$ implies $\mathcal{C}$.

Remark. (1) and (2) together guarantee that, for satisfiable $\mathcal{C}$, there exists a model of $(\mathcal{C} - \{D\}) \cup \{P\}$ which is also a model of $\mathcal{C}$.

Remark. Lemma 3.1 relies essentially on the stability of the clause set $\mathcal{C}$. It is clearly wrong for non-stable clause sets: E.g., take $\mathcal{C} = \{\neg A, A \vee B\}$. $\mathcal{C}$ is satisfiable; but if we replace $A \vee B$ by A we obtain $\mathcal{C}' = \{\neg A, A\}$, which is unsatisfiable. Clearly, $\mathcal{C}$ is not stable; but its deductive closure $R^*_{HS}(\mathcal{C}) = \{\neg A, B\}$ obviously represents an Herbrand model of $\mathcal{C}$.

The replacement of $\mathcal{C}$ by $(\mathcal{C} - \{D\}) \cup \{P\}$ can be described by an operator α which selects a clause D in $\mathcal{C}$ and a literal P in D. The model construction consists in alternately applying transformation α and computing the deductive closure (w.r.t. R^*_{HS}) of the new set of clauses. (Note that $(\mathcal{C} - \{D\}) \cup \{P\}$ need not be stable, so we have to apply R_{HS} after selecting a literal.) We actually need an operator T transforming a stable set into a new stable set in which some non-unit positive clauses are replaced by unit clauses. We define

$$T(\mathcal{C}) = R^*_{HS}(\alpha(\mathcal{C})) \text{ for a stable set } \mathcal{C} \in PDC, \text{ and}$$

$$T^0(\mathcal{C}) = \mathcal{C}, \quad T^{i+1}(\mathcal{C}) = T(T^i(\mathcal{C})) \text{ for } i \geq 0.$$

In [FL95] it is shown that for all $\mathcal{C} \in PDC$ there is a finite fixed point of T. I.e., there exists an i s.t. $T^{i+1}(\mathcal{C}) = T^i(\mathcal{C})$. Then the positive unit clauses of $T^i(\mathcal{C})$ represent an Herbrand model of $\mathcal{C}$. The computation of this fixed point $T^*(\mathcal{C})$ is purely "iterative" and does not require any backtracking.

It is proved in [FL95] that the equivalence of arbitrary atomic representations can be decided. Moreover there exists an algorithm which evaluates the truth values of arbitrary clauses (of adequate signature) over atomic representations. This algorithm itself is based on hyperresolution [FL95]. Therefore atomic representations meet all the requirements stated in the beginning of this section. Note that this extends the

traditional range of model-based resolution, as it gives us a method to evaluate clauses over *infinite* models.

Therefore not only (finite) multiplication tabels but also atomic representations of infinite Herbrand models enjoy properties that make them suitable tools for various applications. We have also shown that for the classes PVD_+ and $\mathrm{OCC1N}_+$ we can algorthimically construct such model representations. Still, it would be interesting to know whether these classes are finitely controllable, i.e. whether there is a model with finite domain for every satisfiable clause set in these classes. (See [DG79] for a legacy of model theoretic results on finite controllability of classes of first order formulas.) Indeed all the classes mentioned above not only admit the construction of Herbrand models but also of finite models.

However, from the computer science point of view, we are not only interested in finite controllability itself but strive for feasible and simple algorithms for the construction of finite models. In particular, we want to avoid exhaustive search up to some fixed limit for the cardinality of the models (as, implictly, suggested by [DG79]) and also do not want to introduce for this purpose equational reasoning methods on the object level (as done e.g. in [CZ91] and [Tam91]). We directly rather want to use the data generated by terminating hyperresolution procedures. To this aim backtracking free algorithm is developed in [FL95] that "extracts" from any finite set of linear atoms $\mathcal{A}$ a finite model that is equivalent to $INT_H(\mathcal{A})$ (w.r.t. any H containing $H(\mathcal{A})$). This is essentially achieved by truth value preserving projections of the Herbrand base into finite subsets of it.

In order to specify the domain of our finite models we need the orthogonal extensions of sets of terms occurring in the atomic representation.

Definition 3.6 *Let $\mathcal{A}$ be a linear atomic representation of an Herbrand model w.r.t. the Herbrand universe H (containing $H(\mathcal{A})$). We introduce an additional constant d not occurring in H. Let γ_d be the substitution that assigns d to all occurring variables; i.e. $\forall x \in V(\mathcal{A}) : \gamma_d(x) = d$. Moreover, let $T'(\mathcal{A})$ be an orthogonal extension of the set $T(A)$ of all non-variable terms that occur in some atom in $\mathcal{A}$. Then*

$$D_{\mathcal{A}} = \{d\} \cup \{t\gamma_d \mid t \in T'(\mathcal{A})\}.$$

(d is intended to represent all ground terms that are not represented by any term in $T'(\mathcal{A})$.)

$D_{\mathcal{A}'}$ serves as domain of a finite model that assigns the same truth values to atoms and clauses as $INT_H(\mathcal{A})$.

Note that we can always compute an orthogonal extension of the set $T(\mathcal{A})$ of terms occurring in a linear atomic representation $\mathcal{A}$ of the Herbrand model $INT_H(\mathcal{A})$. This allows us to define the following function that assigns some element of $D_{\mathcal{A}}$ to each term of the corresponding Herbrand universe.

Definition 3.7 *Let $\mathcal{A}$ be a linear atomic representation of an Herbrand model of a clause set $\mathcal{C}$. Let $H = H(\mathcal{C})$ and let $T'(\mathcal{A})$ be an orthogonal extension of $T(\mathcal{A})$ w.r.t. H. Then for each $t \in H \cup \mathcal{D}_{\mathcal{A}}$ we define $\Phi_{\mathcal{A}}(t)$ as follows:*

(i) If t is an instance of some $s \in T'(\mathcal{A})$ then $\Phi_{\mathcal{A}}(t) = s\gamma_d$ (where γ_d is defined as in Definition 3.6).

(ii) Otherwise, let $\Phi_{\mathcal{A}}(t) = d$.

The definition of $\Phi_{\mathcal{A}}$ enables us to specify concisely the interpretation of the function symbols in the finite model. We are now in the position to present the complete specification of a finite model corresponding to an $INT_H(\mathcal{A})$.

Definition 3.8 *Let $\mathcal{A}$ be a linear atomic representation of an Herbrand model of some clause set $\mathcal{C}$. Then the (finite) interpretation $\mathcal{FM}_{\mathcal{A}} = \langle D, \varphi \rangle$ is defined as follows:*

(i) $D = D_{\mathcal{A}}$.

(ii) $\varphi(c) = c$ for all constants $\in D_{\mathcal{A}}$. For all other constants occurring in $\mathcal{C}$ we define $\varphi(c') = d$.

(iii) $\varphi(f)(t_1, \ldots, t_n) = \Phi_{\mathcal{A}}(f(t_1, \ldots, t_n))$ for all n-ary function symbols f occurring in $\mathcal{C}$.

(iv) $\langle t_1, \ldots, t_n \rangle \in \varphi(P)$ iff $P(t_1, \ldots, t_n)$ is an instance of some $A \in \mathcal{A}$. for all n-ary predicate symbols P occurring in $\mathcal{C}$ and all terms $t_1, \ldots t_n \in D_{\mathcal{A}}$.

Of course, instead of using the set of terms $D_{\mathcal{A}}$ as domain of discourse, we could map $D_{\mathcal{A}}$ bijectively into any domain of the same cardinality and define φ accordingly. It is shown in [FL95] that , given an orthogonal representation of a Herbrand model, the above definition always yields a finite model.

We remark that if there is an H–model $\mathcal{M}$ for some clause set $\mathcal{C}$, s.t. $\mathcal{M} = \{A \mid v_{\mathcal{M}}(A) = \mathbf{true}\}$ is finite then we have a simple subcase of our approach: In that case $\mathcal{M}$ itself may be conceived as a finite set of linear atoms. Moreover, the set $T(\mathcal{M})$ of terms occuring in $\mathcal{M}$ is an orthogonal representation of itself. Therefore $\mathcal{FM}_{\mathcal{M}}$ is a model of cardinality $|T(\mathcal{M})+1|$ (the additional element d represents all ground terms that are not in $T(\mathcal{M})$). Observe that, in general, this model is not minimal w.r.t. the cardinality of the domain.

By the methods described above we obtain finite models for all decision classes of hyperresolution mentioned in section 2. This might suggest that termination of hyperresolution without deriving contradiction implies the finite model property – a conjecture formulated in [FL93]. However this conjecture is easily falsified by the following set of clauses (communicated by M. Baaz):

$$\mathcal{C} = \{P(x,x),\ \neg P(f(x), f(y)) \vee P(x,y),\ \neg P(c, f(x))\}.$$

Hyperresolution terminates on $\mathcal{C}$ giving the atomic model representation $\mathcal{A} = \{P(x,x)\}$. But $\mathcal{C}$ does not have finite models (note that $\mathcal{A}$ is not linear!). This example shows that our methods of model construction and evaluation surpass the range of finitely controllable classes.

References

[CZ91] R. Caferra and N. Zabel, *Extending Resolution for Model Construction.* In: Logics in AI (JELIA '90). Springer Verlag, LNCS 478 (1991), pp. 153–169.

[CZ93] R. Caferra and N. Zabel, *A method for simultaneous Search for Refutations and Models by Equational Constraint Solving.* J. Symbolic Computation 13 (1992) , pp. 613–641.

[DG79] B. Dreben and W.D. Goldfarb, *The Decision Problem.* Addison-Wesley, Massachusetts 1979.

[Fer90] C.G. Fermüller, *Deciding some Horn Clause Sets by Resolution.* In: Yearbook of the Kurt-Gödel-Society 1989, Vienna 1990, pp. 60–73.

[Fer91] C.G. Fermüller, *Deciding Classes of Clause Sets by Resolution.* Ph.D. Thesis, Technical University Vienna, 1991.

[FL93] 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.

[FL95] C.G. Fermüller and A. Leitsch, *Hyperresolution and Automated Model Building.* To appear in the Journal of Logic and Computation.

[FLTZ93] C.G. Fermüller, A. Leitsch, T. Tammet, and N. Zamov, *Resolution Methods for the Decision Problem.* Springer Verlag, LNAI 679 (1993).

[Joy76] W.H. Joyner, *Resolution Strategies as Decision Procedures.* J. ACM 23,1 (July 1976), pp. 398-417.

[Lei90] A. Leitsch, *Deciding Horn Classes by Hyperresolution.* In: CSL'89. Springer Verlag, LNCS 440 (1990), pp. 225–241.

[Lei93] A. Leitsch, *Deciding Clause Classes by Semantic Clash Resolution.* Fundamenta Informaticae 18 (1993), pp. 163–182.

[Lov78] D. Loveland, *Automated Theorem Proving — A Logical Basis.* North Holland Publ. Comp. 1978.

[MB88] R. Manthey and F. Bry, *SATCHMO: A theorem prover implemented in Prolog.* In: 9^{th} Conference on Automated Deduction. Springer Verlag, LNCS 310 (1988), pp. 415–434.

[McC90] W. McCune, *Otter 2.0 Users Guide.* Argonne National Laboratory, Argonne (Ill.), 1990.

[Nol80] H. Noll, *A Note on Resolution: How to Get Rid of Factoring Without Loosing Completeness.* In: 5^{th} Conference on Automated Deduction. Springer Verlag, LNCS 87 (1980), pp. 250–263.

[Tam91] T. TAMMET, *Using Resolution for Deciding Solvable Classes and Building Finite Models.* In: Baltic Computer Science. Springer Verlag, LNCS 502 (1991), pp. 33–64.

[Tam92] T. TAMMET, *Resolution Methods for Decision Problems and Finite Model Building.* Dissertation, Department of Computer Science, Chalmers University of Technology. Chalmers/Göteb0rg, 1992.

[Sla92] J. SLANEY, *FINDER (Finite Domain Enumerator): Notes and Guide.* Technical report TR-ARP-1/92, Australian National University Automated Reasoning Project, Canberra, 1992.

[Sla93] 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.

[Win82] S. WINKER, *Generation and Verification of Finite Models and Counterexamples Using an Automated Theorem Prover Answering Two Open Questions.* J. of the ACM, Vol. 29/2, April 1982, pp. 273–284.