Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

329

E. Börger H. Kleine Büning
M. M. Richter (Eds.)

CSL '87

1st Workshop on Computer Science Logic
Karlsruhe, FRG, October 12–16, 1987
Proceedings

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo

Table of Contents

Proving Finite Satisfiability of Deductive Databases

François Bry and Rainer Manthey
ECRC, Arabellastr. 17, D - 8000 München 81, West Germany

ABSTRACT *It is shown how certain refutation methods can be extended into semi-decision procedures that are complete for both unsatisfiability and finite satisfiability. The proposed extension is justified by a new characterization of finite satisfiability. This research was motivated by a database design problem: Deduction rules and integrity constraints in definite databases have to be finitely satisfiable.*

1. Introduction

When designing deductive databases, deduction rules and integrity constraints have to be checked for various well-formedness properties in order to prevent deficiencies at update or query time. Current research in deductive databases is focussing mainly on databases with definite deduction rules. A necessary well-formedness property for definite databases is the finite satisfiability (i.e., the existence of a finite model) of the set of all deduction rules and integrity constraints (considered as first-order formulas). A method able to detect finite satisfiability of formulas is therefore highly desirable, e.g. as part of an automated design system for definite databases.

Though finite satisfiability is undecidable [TRAC 50], it is at least semi-decidable (like, e.g., unsatisfiability). Therefore checking methods guaranteed to terminate for every finitely satisfiable input may exist. Finite satisfiability has been studied by logicians only indirectly. Since Hilbert's dream of a solution to the decision problem and Church's proof of its unsolvability, various special classes of formulas for which decision procedures may exist have been investigated. Many of these so-called *solvable classes* are in fact *finitely controllable* (a term introduced in [DG 79]), i.e., satisfiability and finite satisfiability coincide for these classes. [DG 79] provides a systematic and unified study of solvable classes in general and of finitely controllable classes in particular. However, decision methods for most of the finitely controllable classes are not known. Furthermore, these classes are characterized by means of rather strong syntactical restrictions which are too stringent for being acceptable in a database context.

Dreben and Goldfarb in addition provide a *finite model lemma* characterizing finitely satisfiable sets of formulas in general by means of term-mappings. In most cases, these mappings don't have any direct practical relevance either, as they are defined on the whole (usually infinite) Herbrand universe. We therefore give a new characterization of finite satisfiability in terms of Herbrand levels and special term-mappings of these finite subsets of the Herbrand universe. This characterization gives rise to extending refutation procedures based on the Herbrand's theorem - i.e., based on a model-theoretic paradigm - into semi-decision procedures for both, unsatisfiability as well as finite satisfiability. When applied to sets of formulas in a finitely controllable class, this extension is a decision procedure for the respective class.

Although it is well-known that direct implementations of Herbrand's theorem are inherently inefficient - as they are based on exhaustive instantiation - a treatment of finite satisfiability in the context of a Herbrand procedure provides valuable insight into the principle techniques on which efficient procedures may rely. Such a more efficient implementation of an instantiation-based proof procedure and its extension into a semi-decision procedure for finite satisfiability have been developped by the authors. They are documented in [MB 87, BDM 88]. In many cases this approach is competitive even if compared with sophisticated resolution-based techniques [MB 88].

This article consists of six sections. Section 1 is this introduction. Section 2 provides a more elaborate motivation of the relevance of finite satisfiability for databases. In Section 3, the Herbrand's theorem and the Herbrand procedure are recalled. Section 4 contains the above-mentioned characterization of finite satisfiability and the corresponding extension of the Herbrand procedure. In Section 5 the extended method is improved. It is combined in Section 6 with a model building approach to deciding propositional satisfiability. Section 7 is a conclusion.

Terminology and Notations

Where appropriate, we consider clauses instead of formulas. We assume that all function symbols denote Skolem functions. Skolemizing (i.e., replacing existentially quantified variables by Skolem terms) does not preserve logical equivalence. A formula F and one of its Skolem forms Sk(F) do not have the same interpretations, since interpretations of Sk(F) assign functions to Skolem function symbols, while interpretations of F ignore these symbols. However, interpretations of Sk(F) induce interpretations of F, and interpretations of F extend into interpretations of Sk(F): Skolemization preserves satisfiability. The proof of this result (see, e.g., [LOVE 78 ,p. 41]) can easily be adapted to proving that skolemization also preserves finite satisfiability.

The character S will always be used for denoting a *finite* set of clauses. H_S denotes the Herbrand universe of S, $H_S{}^i$ the i^{th} level of H_S. $\Delta_S{}^i$ denotes the difference set $H_S{}^i \setminus H_S{}^{i-1}$. Given a subset T of H_S, T[S] denotes the saturation of S over T, i.e., the set of all ground clauses obtained by instantiating variables in clauses in S with terms in T.

Given a clause C and a set of pairs of (possibly ground) terms $\sigma=\{(t_1,u_1), (t_2,u_2), ..., (t_i,u_i), ...\}$ the clause $C\sigma$ is obtained by replacing simultaneously for all i each occurrence of a u_i in C by t_i. E.g., $p(f(a), x)\{(a,f(a))\} = p(a, x)$. The set σ is called a *substitution*.

If a set A is the union of two disjoint sets B and C, we write A = B + C. For other notions, refer to [MEND 69, LOVE 78].

2. Databases and Finite Satisfiability

A deductive database can be formalized in logic [GMN 84, REIT 84] as a triple DB = (F,DR,IC) where:

1. F is a finite set of variable-free atomic formulas.
   (The set of *facts*, or *extensional database*.)
2. DR is a finite set of closed first-order fromulas, used to derive new facts from F.
   (The set of *deduction rules*, or *intentional database*.)
3. IC is a finite set of closed first-order formulas expressing conditions imposed on the extensional as well as intentional databases.
   (The set of *integrity constraints*.)

If DR is empty, DB is a conventional relational database.

In order to preclude derivation of irreducible disjunctive formulas - a formula $F_1 \vee F_2$ is irreducible if neither F_1 nor F_2 are provable - the class of *definite* deduction rules has been defined [KUHN 67]. A formula is *definite* if:

1. all its variables are universally quantified
2. each conjunct of its conjunctive normal form contains exactly one non-negated atom.

A *definite deductive database* is a database the deduction rules of which are definite. In a definite database DB, F$\cup$DR is necessarily satisfiable (a set of definite formulas is always satisfiable). Since F$\cup$DR contains only formulas of the Bernays-Schoenfinkel class, it is even *finitely satisfiable*.

A database DB *satisfies its integrity constraints* if F$\cup$DR $\vdash$ IC, i.e., if all models of F$\cup$DR are models of IC. Therefore, finite satisfiability of IC and moreover of DR$\cup$IC is a necessary condition for definite deductive database [BM 86]. The importance of finite satisfiability for conventional as well as definite deductive databases has already been explicitly mentioned in [FV 84], implicitly in [NG 78].

3. The Herbrand Procedure

Most refutation procedures are justified by means of the following result:

Theorem 1: [Herbrand's Theorem]
S is unsatisfiable iff there is a Herbrand level $H_S{}^i$ such that $H_S{}^i[S]$ is unsatisfiable.

This version of the Herbrand's theorem induces a basic refutation procedure - called the Herbrand procedure - that successively generates the level-saturations $H_S{}^i[S]$ and checks them for propositional unsatisfiability (which is a decidable property). If an unsatisfiable saturation is found, the procedure terminates: Unsatisfiability of S has been shown. In case all Skolem terms in S are constants (i.e., S corresponds to a formula of the Beranys-Schoenfinkel class), H_S is finite and all $H_S{}^i$ are identical. In this case, satisfiability of $H_S{}^0$ implies finite satisfiability of S. Otherwise there are infinitely many levels to be considered, and the Herbrand procedure runs forever if S is satisfiable. All procedures introduced in the following are based on the Herbrand procedure:

Herbrand Procedure:

1. **Initialization**

   i := 0,
   if $H_S{}^0$ is unsatisfiable
   then report unsatisfiability of S and stop
   else if $H_S{}^0 = H_S{}^1$
       then report finite satisfiability of S and stop
       else goto 2.

2. **Unsatisfiability Check**

   i := i+1,
   if $H_S{}^i[S]$ is unsatisfiable
   then report unsatisfiability of S and stop
   else goto 2.

4. A Characterization of Finite Satisfiability

The Herbrand procedure detects finite satisfiability only if the Herbrand universe H_S is finite. There are, however, finitely satisfiable sets of clauses with infinite Herbrand universe. Proposition 3 characterizes these sets by means of the concept of *term-mapping* we first define:

Definition 2:

Let T be a subset of H_S.

A *term-mapping* σ *of* T is a surjective function from T onto T.

A term-mapping σ of a set $T \subseteq H_S$ induces a substitution $\{(\sigma(t),t) \mid t \in T\}$. This substitution is also denoted by σ.

Proposition 3: [Finite Model Lemma]

S is finitely satisfiable iff there is a term-mapping σ of H_S such that $\sigma(H_S)$ is finite and $H_S[S]\sigma$ is satisfiable.

This is the the characterization by Dreben and Goldfarb mentioned in the introduction. A method able to detect finite satisfiability must necessarily provide a feature that corresponds to the search for a term-mapping with finite range. Instead of searching for term-mappings of the Herbrand universe as a whole, we can restrict attention to special mappings of Herbrand levels only.

Definition 4:

A term-mapping σ of an Herbrand level $H_S{}^i$ is *regular* iff

1. $\sigma(H_S{}^i)$ is subterm-closed
   (i.e., if $t \in \sigma(H_S{}^i)$, then all subterms of t are in $\sigma(H_S{}^i)$ as well)
2. $\sigma(t) = t$ for all $t \in \sigma(H_S{}^i)$

Proposition 5:

S is finitely satisfiable iff there is a Herbrand level $H_S{}^i$ and a regular term-mapping σ of $H_S{}^{i+1}$ such that $\sigma(H_S{}^{i+1}) \subseteq H_S{}^i$ and $H_S{}^i[S]\sigma$ is satisfiable.

[**Proof:** (sketched) Necessary condition: If a regular term-mapping of $H_S{}^{i+1}$ is given, it extends naturally into a mapping of H_S. By Proposition 3, S is finitely satisfiable.

Sufficient condition: Assume S is finitely satisfiable. Consider a term-mapping σ of H_S such that $\sigma(H_S)$ is finite and $H_S[S]$ is satisfiable, the existence of which follows from Proposition 3.

Let $<$ be a total order on H_S compatible with the Herbrand level hierarchy, i.e., such that:

1. $t^i < t^j$ if $t^i \in H_S{}^i$, $t^j \in H_S{}^j$, and $i \leq j$
2. $f(t_1{}^1,...,t_n{}^1) < f(t_1{}^2,...,t_n{}^2)$ if f is an n-ary function symbol, the $t_k{}^l$ are in H_S and $(t_1{}^1,...,t_n{}^1) <_L (t_1{}^2,...,t_n{}^2)$ in the lexicographical order $<_L$ induced by $<$.

Let $\sim$ the equivalence relation on H_S defined from σ by $t^1 \sim t^2$ iff $\sigma(t^1) = \sigma(t^2)$.

Since $\sigma(H_S)$ is finite, $H_S/_\sigma$ is finite. Let $H_S/_\sigma = \{C_1,...,C_n\}$ and let c_k be the $<$-smallest element of C_k, for all $k = 1,...,n$. Define a term-mapping τ by $\tau(t) = c_k$, if $t \in C_k$. Let i be the smallest integer such that $H_S{}^i$ contains all c_k; i exists because there are only finitely many c_k.

• τ is regular:

$\tau(H_S{}^i)$ is subterm-closed because of the definiton of $<$.

$\tau(t) = t$ for all $t \in \tau(H_S{}^{i+1})$ by definition of τ and since $\tau(H_S{}^{i+1}) = \{c_1,...,c_k\}$.

• $\tau:(H_S^{i+1}) \subseteq H_S^i$:

by definition of i.

• $H_S^i[S]\tau$ is satisfiable:

By definition of i and ~, $H_S^i[S]$ is isomorphic to $H_S/_\sim[S]$. $H_S^i[S]\tau$ is therefore isomorphic to $H_S/_\sim[S]\sigma$. Since the quotient of H_S by ~ preserves σ, $H_S/_\sim[S]\sigma$ is satisfiable like $H_S[S]\sigma$. $H_S{}^i[S]\tau$ is therefore satisfiable as well.]

Proposition 5 motivates the following first extension of the Herbrand procedure.

<u>Procedure 1:</u>

1. **<u>Initialization</u>**

   i := 0,
   if H_S^0 is unsatisfiable
   then report unsatisfiability of S and stop
   else if $H_S^0 = H_S^1$
       then report finite satisfiability of S and stop
       else goto 2.

2. **<u>Finite Satisfiability Check</u>**

   for each regular term-mapping σ of H_S^{i+1} such that $\sigma(H_S^{i+1}) \subseteq (H_S^i)$
       if $H_S^i[S]\sigma$ is satisfiable
       then report finite satisfiability of S and stop,
   goto 3.

3. **<u>Unsatisfiability Check</u>**

   i := i+1,
   if $H_S^i[S]$ is unsatisfiable
   then report unsatisfiability of S and stop
   else goto 2.

Procedure 1 stops for finitely satisfiable as well as for unsatisfiable S and runs forever iff S is an axiom of infinity. In the following sections, we propose optimizations of this finite satisfiability check.

5. An Improved Procedure

The efficiency of Procedure 1 can be considerably improved, if an optimization technique is applied that is often called Δ-optimization, for example in recursion theory [BAYE 85]. Since a Herbrand level $H_S{}^i$ contains every smaller level it can be computed recursively according to:

$$H_S{}^i = H_S{}^{i-1} + \Delta_S{}^i$$

Regular term-mappings of Herbrand levels may be obtained recursively as well. Consider a term-mapping σ^i of $H_S{}^i$. By definition of regularity the restriction σ^{i-1} of σ^i to $H_S{}^{i-1}$ is regular, too. Therefore a regular term-mapping δ^i: $\Delta_S{}^i \rightarrow H_S{}^i$ exists such that:

$$\sigma^i = \sigma^{i-1} + \delta^i$$

This equation serves as the basis of an optimized finite satisfiability check where regular term-mappings on level i are systematically constructed by augmenting mappings that have already been constructed on level i-1.

However, not all possible augmentations are acceptable. Consider Herbrand levels $H_S{}^1$ = {a, f(a), g(a)} and $H_S{}^2$ = {a, f(a), g(a), f^2(a), f(g(a)), g(f(a)), g^2(a)} and the following regular term-mapping σ^1 of $H_S{}^1$:

$a \rightarrow a$
$f(a) \rightarrow a$
$g(a) \rightarrow g(a)$

The fact that σ^1 replaces f(a) by a and leaves g(a) unchanged already "predetermines" the assignments of a to f^2(a), and of g(a) to g(f(a)) in any acceptable extension of σ^1. Therefore only replacements for the remaining terms g^2(a) and f(g(a)) have to be chosen when constructing a possible δ^2. We say that δ^2 has to be σ^1-*compatible.* This property is formally defined as follows:

Definition 6:

Let σ^{i-1} be a term-mapping of $H_S{}^i$ and δ^i a term-mapping of $\Delta_S{}^i$.
δ^i is σ^i-*compatible* iff for every term $t = f(t_1,\dots,t_n) \in \Delta_S{}^i$, we have:

$$t' = f(\sigma^{i-1}(t_1),\dots,\sigma^{i-1}(t_n)) \in H_S{}^{i-1} \quad \Rightarrow \quad \delta^i(t) = \sigma^{i-1}(t')$$

The previous remarks lead to the following procedure sound and complete for both unsatisfiability and finite satisfiability. (A term-mapping σ is expressed as a set of pairs (σ(t),t).)

Procedure 2:

1. **Initialization**

   M^0 :- $\{(t,t) \mid t \in H_S{}^0\}$,
   $i := 0$,
   if $H_S{}^0$ is unsatisfiable
   then report unsatisfiability of S and stop
   else if $H_S{}^0 = H_S{}^1$
       then report finite satisfiability of S and stop
       else goto 2.

2. **Finite Satisfiability Check**

   $M^{i+1} := \varnothing$,
   for each $\sigma^i \in M^i$
       for each σ^i-compatible and regular term-mapping δ^{i+1}: $\Delta_S{}^{i+1} \rightarrow H_S{}^{i+1}$
           $\sigma^{i+1} := \sigma^i + \delta^{i+1}$,
           if σ $H_S{}^i[S]$ is satisfiable
           then report finite satisfiability of S and stop
           else $M^{i+1} := M^{i+1} \cup \{\sigma^{i+1}\}$,
   goto 3.

3. **Unsatisfiability Check**

   $i := i+1$,
   if $H_S{}^i[S]$ is unsatisfiable
   then report unsatisfiability of S and stop
   else goto 2.

This procedure, as opposed to the Herbrand procedure, is no longer linear: It performs in fact a breadth-first search of a tree the nodes of which are the members of the sets M^i.

6. Finite Satisfiability Checking on the Basis of g-Models

Saturations over Herbrand levels are sets of ground clauses and may therefore be checked for unsatisfiability by means of an appropriate decision procedure for propositional calculus. Most of the refutation procedures prior to resolution, such as the tableaux method [BETH 59, SMUL 68] and the Davis-Putnam method [DP 60] are based on the notion of g-model [LOVE 78].

<u>**Definition 7:**</u>

A *g-model* M of a set S of *ground* clauses is a set of unit clauses such that:

1. M does not contain complementary literals.
2. Every clause in S contains a literal in M.

Combining level saturation and construction of g-models leads to a characterization of unsatisfiability that can be seen as a corollary to the Herbrand's theorem.

<u>**Proposition 8:**</u>

S is unsatisfiable iff some level saturation $H_S{}^i[S]$ has no g-model.

In a similar way we characterize finite satisfiability by applying regular term-mappings directly to the g-models of level saturations:

<u>**Proposition 9:**</u>

S is finitely satisfiable iff there is a level $H_S{}^i$, a g-model g^i of $H_S{}^i[S]$, and a regular term-mapping σ of $H_S{}^{i+1}$ such that $\sigma(H_S{}^{i+1})\subseteq H_S{}^i$ and $g^i\sigma$ is satisfiable.

It can easily be checked whether application of σ to g^i preserves satisfiability: $g^i\sigma$ is unsatisfiable iff it contains two complementary unit clauses. Furthermore, g-models may also be obtained recursively: Every g-model g^i of $H_S{}^i[S]$ is the union of a g-model g^{i-1} of $H_S{}^{i-1}[S]$ and of a g-model d^i of $H_S{}^i[S]\backslash H_S{}^{i-1}[S]$.

We conclude this section with a semi-decision procedure for finite satisfiability and unsatisfiability based on g-models.

<u>**Procedure 3:**</u>

1. <u>**Initialization**</u>

M^0 :- $\{(t,t) \mid t\in H_S{}^0\}$,
i := 0,
G^0 := set of all g-models of $H_S{}^0[S]$,
if $G^0 = \varnothing$
then report unsatisfiability of S and stop
else if $H_S{}^0 = H_S{}^1$
  then report finite satisfiability of S and stop
  else goto 2.

2. Finite Satisfiability Check

$M^{i+1} := \emptyset$,
for each $g^i \in G^i$
 for each $\sigma^i \in M^i$
  for each σ^i-compatible and regular term-mapping $\delta^{i+1}: \Delta_S^{i+1} \to H_S^{i+1}$
   $\sigma^{i+1} := \sigma^i + \delta^{i+1}$,
   if $\sigma^{i+1}(H_S^{i+1}) \subseteq H_S^i$ and $g^i\sigma^{i+1}$ is satisfiable
   then report finite satisfiability of S and stop
   else $M^{i+1} := M^{i+1} \cup \{\sigma^{i+1}\}$,
goto 3.

3. Unsatisfiability Check

$i := i+1$,
$G^i := \emptyset$,
for each $g^{i-1} \in G^{i-1}$
 for each g-model d^i of $H_S^i[S] \setminus H_S^{i-1}[S]$
  $g^i := g^{i-1} \cup d^i$,
  if g^i is satisfiable
  then $G^i := G^i \cup \{g^i\}$,
if $G^i = \emptyset$
then report unsatisfiability of S and stop
else goto 2.

As an example consider the following set S of clauses, expressing that every human has an ancestor, nobody is his own ancestor, and there are humans ('a' is a Skolem constant, 'f' is a Skolem function):

not anc(X,X)
not human(X) human(f(X))
not human(X) anc(X,f(X))
human(a)

M^0 consists of the identity mapping on H_S^0, and G^0 consists of the only g-model of H_S^0

$$g^0 = \{\text{not anc}(a,a), \text{human}(a), \text{human}(f(a)), \text{anc}(a,f(a))\}$$

There are two possible ways how to map $\Delta_S^1 = \{f(a)\}$ on $H_S^1 = \{a, f(a)\}$. Both are compatible with the mapping in M^0; thus M^1 contains $\sigma_1^1 = \{(a,a),(f(a),a)\}$ and $\sigma_2^1 = \{(a,a),(f(a),f(a)\}$. The first of these mappings is not yet sufficient because $g^0\sigma_1^1$ contains the complementary unit clauses 'anc(a,a)' and 'not anc(a,a)'. The second is not sufficient as $\sigma_2^1(H_S^1) = H_S^1$ is not a subset of H_S^0. There is only one possibility how to extend g^0 into

$$g^1 = g^0 \cup \{\text{not anc}(f(a),f(a)), \text{human}(f^2(a)), \text{anc}(f(a),f^2(a))\}$$

Now σ_2^1 can be extended into $\sigma_1^2 = \{(a,a), (f(a),f(a)), (f^2(a),a)\}$. As $\sigma_1^2(H_S^2) = H_S^1$, and $g^1\sigma_1^2 = \{$not anc(a,a), human(a), human(f(a)), anc(a,f(a)), anc(f(a),a), not anc(f(a),f(a))$\}$ is satisfiable, finite satisfiability has been confirmed: $g^1\sigma_1^2$ constitutes a finite model of S.

7. Conclusion

In this article, we have proposed an extension of the Herbrand procedure into a semi-decision method complete for both finite satisfiability and unsatisfiability. This procedure is justified by a new characterization of finite satisfiability based on the finite model lemma of Dreben and Goldfarb. It is shown how the additional feature we propose combines with model building refutation procedures. A practical procedure following this approach is described in [BDM 88]. It has been implemented in Prolog as a component of a database design system. The present paper provides a theoretical justification for this implementation.

Other approaches to finite satisfiability checking are possible. One of them, outlined in [BM 86], relies on resolution. Its main difference with the procedures described here is that it delays instantiation as much as possible as opposed to the "instantiation-first" strategy of Herbrand-like procedures. However the resolution-based approach cannot rely on a linear strategy, but requires a saturation strategy in order to reach all finite models.

Whatever approach is chosen, one has to be aware that the capability to detect finite satisfiability necessarily introduces a severe overhead in case the set of formulas under consideration turns out to be unsatisfiable.

8. Acknowledgement

The authors are grateful to Jean-Marie Nicolas for his patience and support during this research, and to Hervé Gallaire and all ECRC for providing such a stimulating environment.

9. References

[BAYE 85] Bayer, R.
Query Evaluation and Recursion in Deductive Database Systems.
Technical Report TUM-18503, Technical University of Munich, 1985.

[BDM 88] Bry, F., Decker, H. and Manthey, R.
A Uniform Approach to Constraint Satisfaction and Constraint Satisfiability in Deductive Databases.
In *Advanced in Database Technology - Proc. EDBT '88.* Mar., 1988.
Springer-Verlag, LNCS 303.

[BETH 59] Beth, E.W.
The Foundation of Mathematics.
North Holland, Amsterdam, 1959.
cited in [SMUL 68].

[BM 86] Bry, F. and Manthey, R.
Checking Consistency of Database Constraints: A Logical Basis.
In *Proc. 12th Int. Conf. on Very Large Data Bases (VLDB '86).* Aug. 25-28, 1986.

[DG 79] Dreben, B. and Goldfarb, W.
The Decision Problem - Solvable Classes of Quantified Formulas.
Addison-Wesley, Reading, Massachussetts, 1979.

[DP 60] Davis, M. and Putnam, H.
A Computing Procedure for Quantification Theory.
J. of the ACM 7(3), Jul., 1960.

[FV 84] Fagin, R. and Vardi M.Y.
The Theory of Data Dependencies - An Overview.
In *Proc. of ICALP.* 1984.

[GMN 84] Gallaire, H., Minker, J. and Nicolas, J.-M.
Logic and Databases: A Deductive Approach.
ACM Computing Surveys 16(2), Jun., 1984.

[KUHN 67] Kuhns, J.L.
Answering Questions by Computer - A Logical Study.
Rand Memo RM 5428 PR, Rand Corp., Santa Monica, Calif., 1967.

[LOVE 78] Loveland, D.
Automated Theorem Proving: A Logical Basis.
North-Holland, Amsterdam, 1978.

[MB 87] Manthey, R. and Bry, F.
A Hyperresolution Proof Procedure and its Implementation in Prolog.
In *Proc. 11th German Workshop on Artificial Intelligence (GWAI '87).* Sep. 28-Oct. 2, 1987.
Springer-Verlag, IF 152.

[MB 88] Manthey, R. and Bry, F.
SATCHMO: A Theorem Prover Implemented in Prolog.
In *Proc. 9th Conf. on Automated Deduction (CADE '88).* May 23-26, 1988.
Springer-Verlag, LNCS.

[MEND 69] Mendelson, E.
Introduction to Mathematical Logic.
Van Nostrand Reinhold, New York, 1969.

[NG 78] Nicolas, J.-M. and Gallaire, H.
Database: Theory vs. Interpretation.
In *Logic and Databases.* Plenum, New York, 1978.

[REIT 84] Reiter, R.
Towards a Logical Reconstruction of Relational Database Theory.
In *On Conceptual Modelling.* Springer-Verlag, Berlin and New York, 1984.

[SMUL 68] Smullyan, R.
First-Order Logic.
Springer-Verlag, New York, 1968.

[TRAC 50] Trachtenbrot, B.A.
Impossibility of an Algorithm for the Decision Problem in Finite Classes.
Dokl. Acad. Nauk. SSSR 70 , 1950.
in Russian, translated into English in Amer. Soc. Trans., Series 2, 23, 1963.