Chapter 1

INTRODUCTION

It is a central problem of logic and computer science to determine whether a formula in predicate logic is satisfiable. Its roots can be traced back to Leibnitz's brave vision of a calculus ratiocinator [Leib], which would allow to settle arbitrary problems by purely mechanical computation, once they have been translated into a special formalism. The actual history of the modern understanding of this project starts around 1900 when Hilbert defined the problem to find an algorithm which decides the validity of first order predicate logic formulas [Hil01]. He called this decision problem the "fundamental problem of mathematical logic". At that time a positive solution seemed to be merely a question of wealth of mathematical invention.

Although no precise concept of "algorithm" was available during the first three decades of our century, some decidable classes of predicate logic were found during this time. The algorithms provided for these classes were clearly effective in any plausible intuitive sense of the word. One of the first results was the decidability of the monadic class (i.e. the class of first order formulas containing only one-place predicate symbols) [Löw15]. In the same paper Löwenheim showed that dyadic logic gives a reduction class, i.e. a class of first order formulas effectively "encoding" full predicate logic; a decision algorithm for such a reduction class could thus be transformed into a decision algorithm of full predicate logic.

In 1936 A. Church demonstrated rigorously that a positive solution of the decision problem is impossible [Chu36], which means that no algorithm can decide the validity (or the satisfiability) of arbitrary first order formulas. From this result the undecidability of the reduction classes follows immediately. However the search for decidable classes has been continued (indeed it gained additional practical and epistemological importance).

The decision problem with respect to the prefix structure of prenex normal forms is solved completely (see [DG79] and [Lew79]). There are much less results, however, for classes with full functional structure; one of the first results on functional classes was the decidability of the satisfiability problem for formulas with prefix $\forall$ and arbitrary function symbols in the matrix [Gur73]. Recently functional classes have been investigated in papers of the authors ([Fer90], [Lei90], [Tam90]). But rather than finding new decidable classes only, the purpose of this monograph is to analyse the decision methods themselves; as basic technique we use resolution and its refinements and variants. While the book of Dreben and Goldfarb on decidable classes [DG79] is based on model theoretical techniques, our approach follows the proof theoretical tradition. Before Robinson's famous paper on the resolution principle [Rob65] was published, S. Maslov succeeded to prove the decidability of the $\exists^*\forall^*\exists^*$ - Krom class (the so called Maslov class) by proof theoretical means [Mas64]; he used the inverse method which can be considered as a special version of the resolution method based on sequent calculus. One of the authors used a variant of this method to prove the decidability of Maslov's K-class [Zam72], [Zam89]. A typical feature of this method (and of resolution) is the essential use of most general unifiers; Instead on ground level inference takes place on the predicate logic level with variables (Herbrand's theorem is used only for completeness results - not for decision methods). Further results on decidable classes, based on this method, were obtained by Zamov and Sharonov in 1974 [ZS74].

Independently Joyner showed in his thesis how the resolution principle can be adapted to decide some of the classical prefix classes [Joy73]. His method consists in defining specific A-ordering refinements of resolution, which are complete and produce only finitely many resolvents on certain clause classes (the classes are Skolemized conjunctive normal forms of prefix classes). Although Joyner used A-ordering methods only, his basic idea is quite general: Find a complete resolution refinement which is terminating (i.e. it produces only finitely many resolvents) on a class of sets of clauses Γ. Then this refinement decides Γ, because either a contradiction is derived or the deduction process stops "without success"; in the latter case we have found that the set of clauses is satisfiable (without defining a model). This is the principle on which most of the methods and results of this monograph are based (although for some classes refinements are defined which are complete on the corresponding decision class only).

In chapter 3 (written by A.Leitsch) semantic clash resolution as decision procedure is investigated. Here some earlier results on decidable Horn classes [Fer90], [Lei90] are generalized to clause classes with arbitrary propositional structure. The classes are characterized by term depth and variable-occurrence properties, but there are no restrictions on function symbols (thus the classes cannot be obtained by Skolemization of function-free classes). Instead of fixing a specific semantic refinement, a prover generator is defined, which produces terminating setting refinements out of the syntactical structure of the clause sets. By this technique the classes PVD and OCC1 are shown to be decidable. OCC1 is defined by occurrence restrictions to variables in the (semantically) positive part of the clause; for its decision restricted factoring and condensing are required. PVD is a strong extension of DATALOG and is "sharp" w.r.t. undecidability (a small change in the definition of the class yields the representation of a word problem in an arbitary equational theory). The decision procedure for PVD is shown to give an efficient resolution decision procedure for the Bernays - Schönfinkel class. A subchapter is devoted to a subclass of the Horn clause implication problem; here positive hyperresolution with ordered rule clauses provides a basis for the decision algorithm.

Chapter 4 (written by T.Tammet) gives a detailed treatment of ordering refinements and their completeness properties. Various types of ordering principles (a-priori and a-posteriori orderings) are compared and general completeness conditions for π - orderings are presented. It is shown that all ordering refinements are compatible with backward subsumption (but not with forward subsumption). A-ordering methods appear as specific π -refinements, where the full deletion method is allowed. A short overview (examples) of using ordering refinements as decision algorithms is (are) presented. Finally an ordering, which is not a π -ordering (called v-ordering), is used to decide the class E^+ (containing the one-variable functional clause class). While the v-ordering refinement is shown to terminate on E^+, its completeness (on E^+) is still an open problem. However it is proved that the v-ordering is complete on a subclass of E^+ by a locking technique.

Chapter 5 (written by C. Fermüller) describes various decision procedures based on A-ordering refinements and saturation. To make the chapter more self-contained, there is a detailed presentation of semantic-tree-completeness, A-orderings and splitting principles. Via the concept of covering terms several A-ordering refinements are shown to terminate on specific clause classes. One of the classes, being decidable this way, is the class E1 which properly contains the extended Ackermann class. For the class E^+ (which is also discussed in chapter 4) an (a-posteriori) ordering is defined which is not an A-ordering. Similar to the v-ordering refinement in chapter 4 this ordering refinement is shown to terminate on E^+. But by combining it with a saturation technique one gets a refinement which also is complete on E^+ (this yields the first full proof of decidability for E^+). Combining A-ordering with saturation gives a decision procedure for the class S^+ which is a generalization of the initially extended Skolem class (and thus contains the Gödel class). A pure saturation method, combined with unrestricted resolution is introduced as decision procedure for a class containing Maslovs $\exists^*\forall^*\exists^*$-Krom class; while the completeness of the method is independent of the propositional clause form, the Krom-property is needed for termination.

In chapter 6 (written by N. Zamov) a decision procedure for Maslov's K-class based on an ordering refinement is presented. The K-class is very general and contains the initially extended Skolem class (but it is not comparable with the S^+-class in chapter 5). A detailed analysis of the behaviour of most general unifiers and ordered resolvents of the class is perfomed by the concept of regular terms and by the domination relation among literals. A specific π-ordering is defined on the ground level which then is lifted to an ordering on the general level. The termination of the resulting ordering refinement on K is proved (the property of regularity is preserved under the ordering refinement) and the completeness of the method is inferred by the completeness results in chapter 4.

In chapter 7 (written by T.Tammet) methods of resolution and narrowing are applied to automatical finite model building. Rather than just testing satisfiability of clause sets by termination of resolution refinements, a method is presented which constructs finite models for the union of the initially extended Ackermann- and the essentially monadic class. First the set of clauses is transformed into a set containing literals only of the essentially monadic type. Then, by use of narrowing on the set of ground terms of the Herbrand universe, a finite domain interpretation of the function symbols is constructed, which then can be extended to an interpretation of the predicate symbols. By its proof theoretical nature, based on tools of computational logic, this method of building finite models is much more efficient than exhaustive search through finite domain interpretations up to some (recursively computable) domain size. A similar method invented by the author [Tam90] was tested on several examples and proved to be very efficient.

In chapter 8 several applications of resolution decision methods are discussed. Consistency is of central importance to terminological logics, such as KL-ONE. It is shown that the language ALC (which can be considered as a basic language for KL-ONE systems) can be translated into the class S^+ (defined in chapter 5); thus A-ordering + saturation gives an algorithm testing consistency of formulas from ALC (there is also an apriori ordering refinement deciding the translation of ALC). It is shown that a variant of ALC allowing arbitrary function symbols and ground equalities can be translated into the one-variable class (which is a subclass of E^+ discussed in the chapters 4 and 5). The following sections are spent on descriptions of experiments with resolution decision procedures as theorem provers. The prover generator, defined for deciding the class PVD in chapter 3 and based on semantic clash resolution, is tested on 3 examples. On the first two examples the PVD-decision method proved to be strongly superior to a semantic clash prover with fixed setting; example 3 could not be handled by any refinement described in the monograph. Also in the case of ordering refinements decision procedures are very efficient theorem provers; so formulas in the classical book of Church [Chu56] could be proved much faster than by other theorem provers (the latter even failed on some examples). The experiments thus indicate that it pays out to incorporate decision methods into general theorem proving. A way to do this, is to design an expert system classifying a set of clauses S before deduction; if it turns out that S is contained in a decidable class Γ then apply the resolution decision procedure for Γ. Although this method cannot always work (by the undecidability of clause logic) it can be quite useful for practical purposes.