Automated Model Building

Ricardo Caferra and Alexander Leitsch and Nicolas Peltier

Applied Logic Series 31, Kluwer, 2004

November 11, 2004

Preface

On the history of the book:

In the early 1990s several new methods and perspectives in automated deduction emerged. We just mention the superposition calculus, meta-term inference and schematization, deductive decision procedures, and automated model building. It was this last field which brought the authors of this book together. In 1994 they met at the Conference on Automated Deduction (CADE-12) in Nancy and agreed upon the general point of view, that semantics and, in particular, construction of models should play a central role in the field of automated deduction. In the following years the deduction groups of the laboratory LEIBNIZ at IMAG Grenoble and the University of Technology in Vienna organized several bilateral projects promoting this topic. This book emerged as a main result of this cooperation. The authors are aware of the fact, that the book does not cover all relevant methods of automated model building (also called model construction or model generation); instead the book focusses on deduction-based symbolic methods for the construction of Herbrand models developed in the last 12 years. Other methods of automated model building, in particular also finite model building, are mainly treated in the final chapter; this chapter is less formal and detailed but gives a broader view on the topic and a comparison of different approaches.

How to read this book:

The chapters 3 (resolution based methods) and 4 (constraint-based methods), which are largely based on former scientific work of the authors, can be considered as the core of the book; they depend on each other and should be read in this order. Chapter 2 (preliminaries), which should be read by need only, gives the basic concepts and makes the book accessible to graduate students without a firm background in first-order logic and automated deduction. Chapter 5 (on model representation and evaluation) essentially depends on the chapters 3 and 4. Chapter 6 on finite model building in largely independent of the former chapters and gives a broader view on the topic as a whole.

Acknowledgments

We would like to thank Chris Fermüller for several substantial and fruitful discussions in the early stage of the book; important parts of the book are based on his research carried out together with one of the authors. Our special thanks go to Joseph Goguen for his encouragement and his constructive criticism; his remarks on the history, the general methodology and the philosophical background helped to improve the conceptual level of the book. He also gave us very valuable information about several other deductive methods, resulting in a clearer and more profound presentation our approach within the field of deduction as a whole. Finally we would like to thank Dov Gabbay for promoting and supporting our work and for making its publication possible.

Chapter 1

Introduction

1.1 Automated Deduction

This book is on Automated Model Building. Certain keywords and domains are immediately evoked by this title. We shall consider here three of them that seem to be the most important: *model, model theory* and *automated deduction.*

The concept of *model* is a very deep one and represents a challenge to some of our strongest intellectual abilities. It is used in different fields with different intended meanings (see for example [118]). Maybe the oldest scientific meaning of "model" is that of a mathematical (not necessarily logical) object allowing to explain observational data (see for example [70]). It is important to point out that the idea of non-uniqueness is implicitly accepted here: better models explain more observations more precisely. This, for example, is the concept of model used by physicists and biologists.

The other widespread meaning is that used in mathematical logic. The relationship among different uses of the notion "model" in mathematical logic and in empirical sciences did not receive very much attention in scientific research. But in [203] the different concepts of model in various scientific fields (like logic, physics, social sciences) are carefully analyzed and the logical Tarskian notion is shown to be adequate to empirical sciences. In the framework of Artificial Intelligence the model building process in non-deductive inference, for example in abduction, is quite close to that of empirical sciences (and to natural language understanding). The difficulty of checking consistency of a proposed explanatory hypothesis with an existing theory and observed facts is one of the main impediments to the mechanization of the logic of discovery (see for example [176, 110]).

The importance of models in the systematization of the art of reasoning was recognized very early. It is well known to logicians (see for example [145]) that some explanations relating major, middle and minor premisses in Aristotle's syllogistic figures are correct only for certain concrete terms (i.e. with respect to certain *models*) and can be also be proven wrong for other concrete terms (representing *counter-models*).

The content of this book is hardly related to classical model theory, which, according to [44], is "the branch of mathematical logic which deals with the relation between a formal language and its interpretations, or models", but is much closer to Automated Deduction.

Therefore we are naturally compelled to "review" automated deduction, its goals, what it is missing, ... as well as to talk about some historical and philosophical matters.

At this time Automated Deduction is about 45 years old. The field is reaching maturity and its theoretical and practical results (some of them quite striking) ensure it a firm place somewhere between Logic, Computer Science and Artificial Intelligence. Maybe the best witness of maturity is that, today, we have a much clearer view of the feasibility (or non-feasibility) of the goals implicitly defined at the beginning. Still most of the problems, identified as central already at the very beginning, lack satisfactory solutions; we just mention proof planning, the use of analogy, learning and *constructing counter-examples* (see e.g. [20, 212, 9, 186]).

In the attempt to describe the field of Automated Deduction two basic methodological questions arise quite naturally:

- What is automated deduction?
- What should (and could) automated deduction be?

A scholar way to answer the first question is to have a look at the bibliography for corresponding definitions or, at least for attempts to characterize more or less precisely the field proposed by its practitioners. Surprisingly, there are only a few informal "definitions" around automated deduction (often merged with automated theorem proving), some of them rather tautological; but all of them partially address the second point above, at least implicitly. We just mention four of them given by renowned scientists in the domain of deduction, logic and mathematics.

> *By automated theorem proving we mean the use of a computer to prove non-numerical results, i.e. determine their truth (validity). Often (but not always, e.g. decision procedures) we may*

also require human readable proofs rather than a simple statement: "proved". Two modes of operation are used: fully automatic proof searches and man-machine interaction ("interactive") proof searches. The label ATP will cover both. D. Loveland [144].

And by automated deduction I am thinking of a field broader than automatic theorem proving including also automatic processing of proofs. D. Prawitz [180].

The subject of automated deduction deals with computerizing the logical aspect of mathematics, while the subject of symbolic computation deals with computerizing the computational aspects. M. Beeson [14]. [1]

The subject of automated reasoning is concerned with using computers to help the humans discover and write formal proofs. R. Constable [63] .

The second and fourth characterizations are the more interesting ones: they take into account features with increasing importance in reasoning systems. It should be remarked that the fourth defines automated *reasoning* (note that reasoning is a broader concept *including* deduction). [2]

The study of automated reasoning (instead of automated deduction) as well as the cooperation of systems for theorem proving and symbolic computation (compared in Beeson's characterization) should not be forgotten when answering the question about the very nature of automated deduction.

A more technical answer to the question, what automated deduction actually is, can be given by a historical analysis of the subjects that have been treated by the researchers in the field.

A few survey papers have been published ([68, 144, 213, 69]). A compilation of early papers that have founded automated deduction in [194, 195] is also a reference in the field. [20] gives the state of the art when the domain was 25 years old (see also [**?**]).

The recent handbook [**?**] bears witness of the maturity of the field.

Concerning applications, some major impacts of automated deduction on science, economy and society deserve to be mentioned: e.g. discovering of errors in chips and the role of resolution in (constraint) logic programming.

---

[1] Beeson proposes the equation Mathematics = Logic + Computation, by admitting that it is an overstatement, since there are other aspects of mathematics, as for instance the visual aspect.

[2] The term *automated reasoning* was not introduced until 1980 (see [**?**], page 114).

Recently new important potential application have emerged. One of them is deduction in the internet, where there is a growing need for powerful inference [3] engines, a consequence of its generalized use in more and more fields and of the user's expectations of new facilities.

Directly related to the goal of the present work is the fact that, in the whole set of the surveys and handbook articles on automated deduction mentioned above, only a few pages in [91] are devoted to the subject of model building. In [91], page 1828 it is written:

> Automated model building (sometimes also called model generation) is becoming a discipline on its own and one of the more fascinating applications of automated deduction.

1.2 Formal and Informal Proofs

In defining the field of automated deduction, we might observe that only a few attempts have been made to build bridges between automated deduction and the deep works on the notion of "proof" by mathematicians and philosophers of mathematics. But, clearly, this notion lies at the very heart of automated deduction; its analysis seems unavoidable once the state of the art allows for treating "big" proofs. In particular *proof presentation*, *proof schemata* and *explanation of proofs* become important issues in the domain. We just mention [186] and [102] (see also [32]).

At this point the question what is a proof? naturally arises.

The notion of proof is a very deep one, especially if "real" mathematical proofs are considered. The conditions required for accepting an object as a proof of a fact were different ones in different times. This also applies to mathematics; just consider proofs in analysis before and after Weierstrass. Etymologically, to prove is related to ensuring quality (note that probus means: of good quality, honest, faithful).

The statement below is probably supported by most mathematicians:

> The process of deducing some sentences from others is the most important part of the mathematical work. [181], page 179.

Historically, the importance of proofs has been recognized first in ancient Babylon:

[3] Inference, of course, *includes* deduction.

> ...More important than the technical algebra of these ancient Babylonians is their recognition -as shown by their work- of the necessity of *proof* in mathematics.
>
> Until recently it has been supposed that the Greeks were the first to recognize that proof is demanded for mathematical propositions. This was one of the most important steps ever taken by human beings. Unfortunately it was taken so long ago that it led nowhere in particular so far as our own civilization is concerned -unless the Greeks followed consciously, which they may well have done. They were not particularly generous to their predecessors. [16], page 18.

But the concept of a rigorous mathematical proof (i.e. using logical and non logical axioms and inference rules) [4] is due to the Greeks; it is the most distinctive feature between Greek and Babylonian mathematics.

It is generally acknowledged that Parmenides was the first to propose proofs as deductive reasoning starting with irrefutable statements and using rigorous chains of deductions. He also proposed dividing proofs into parts, the premises of one part being the conclusions of previous ones (this corresponds to *structuring proofs in lemmata*), see for example [141]. The most important works in Greek science addressing the concept of proof are Aristotle's Organon and Rhetorics, and of course Euclid's Elements.

It is not surprising that, among the "intelligent" activities handed over to computers, finding proofs in a formal system was a most appealing one. Indeed, the Logic Theory Machine [160], able to produce proofs in propositional calculus, is usually considered as the first Artificial Intelligence program.

With the proof of the 4-colour theorem [2] computer aided theorem proving became "public domain". The computer's task in the solution of the famous problem basically consisted in testing a very high number of cases. In fact Appel and Haken tested 1482 graphs representing all possible map configurations (the computation took more than 1000 hours).

The use of computers for proving theorems inspired mathematicians, logicians and philosophers of science. Their reflections on this matter might help us in developing a better understanding of automated deduction as it is and as it should be.

In a well known paper [207], the author writes:

> I will argue that computer-assisted proofs introduce experimental methods into pure mathematics.

[4] The Greeks used other names: postulates, common notions.

Other authors formulated similar theses (see for example [103, 104]), sometimes in a more radical way [5].

Tymoczko asks the question What is a proof? and gives an answer by identifying 3 main characteristics:

- Proofs are convincing
- Proofs are surveyable
- Proofs are formalizable

The first point addresses one of the basic requirements of proof theory; indeed, arguments have to be convincing to be accepted as a proof. [6]

The second point is of particular relevance when 'big' proofs are investigated. It is also reflected in the opinion of J.P Serre (see [124]):

> Tout résultat qui est obtenu par un moyen humainement invérifiable n'est pas une démonstration.

In this context it is interesting to look at the thoughts of mathematicians concerning their own proofs (e.g. see [134, 135]). Lam's analysis started with the title of an article in a newspaper: Is a math proof a proof if no one can check it?. The alluded proof was the proof of a conjecture (by K. F. Gauss) about the projective plan of order 10 and consumed 3000 hours of a CRAY-1A. [7] Lam proposes the term computed result instead of proof for this kind of "proofs" and he contends that, in case of computed-based results, correctness is not absolute , but just almost sure. [8]

The third point corresponds to the definition of proof in a formal system and the difficulties it raises are essentially technical ones. It is the underlying idea in logical frameworks, in particular in the AUTOMATH project (see below).

The authors think that the relation between human proofs and computer-based proofs is made clearer by the distinction (due to Martin

[5]The first statement of [104] is: Mathematics is a natural science.

[6]The need to capture human's reasoning features was present in Gentzen's natural deduction. Gentzen wanted *"to set up a formal system that came as close as possible to actual reasoning"*.

[7]At this time it was estimated that there were non detected hardware errors approximately every 1000 hours !!

[8]The author forgets that there have been "false proofs", some of them proposed by great mathematicians, long time before computers were invented.

Löf, see [15]) between *proofs* and derivations. A *proof* contains the computational information needed by a computer in order to verify a proposition. A derivation is an object that convinces us of the truth of a proposition. Thus what is found in text books are derivations. [9]

1.3 Proofs and Automated Deduction

Above we gave a brief description of the notion of "proof" and of the views of mathematicians, logicians and philosophers concerning computer-based proofs. But how did researchers in automated deduction look at proofs? Different approaches exist since the very beginning of research in the field. They can be characterized by the following classification (the main principles guiding the approaches are characterized by informal key phrases). Many works in automated deduction can be assigned to other categories as well (e.g. if we use the techniques as a criterion), a general problem appearing in taxonomy. Here we classify the works according to their *aim* and relegate techniques to a secondary position.

1. (**"As fast as possible"**). Designing and implementing first-order provers based on a single calculus (resolution, tableaux, model elimination,...). Research efforts are concentrated on strategies and implementation techniques. Good examples and powerful systems are very well known; we just mention the systems

    - Otter (http://www-unix.mcs.anl.gov/AR/otter/)
    - Setheo (http://www4.informatik.tu-muenchen.de/ letz/setheo/)
    - Spass (http://spass.mpi-sb.mpg.de/)
    - Vampire (http://www.cs.man.ac.uk/ riazanoa/Vampire/)
    - Gandalf (http://www.math.chalmers.se/ tammet/gandalf/)

    Though they are not based on logical calculi, theorem provers in elementary geometry deserves to be included in this category. Algebraic methods have allowed to prove easily a lot of theorems for which resolution theorem provers fail, as well as to solve some open questions (see for ex. [49]).

    A particularly important technique, namely **rewriting**, must be mentioned here. Rewriting is considered as a *specific* domain of reasoning,

[9] One of the goals of the AUTOMATH project was a complete formal checking of the proofs (derivations) in Landau's "Grundlagen der Analysis" (see [72]).

closely linked to the independent research field of *Symbolic Computation*. We just mention the work of B. Buchberger [30] and the forthcoming book of J. Goguen [101].

2. (**"Induction"**). Designing and implementing provers dedicated to induction. The use of induction is unavoidable in mathematical reasoning (H. Poincaré even considered it as the mathematical reasoning *par excellence*). Special induction provers have been developed (the most popular being that of Boyer and Moore [28]) producing some remarkable practical results.

3. (**"As general as possible"**). Designing and implementing higher-order logical frameworks allowing the user to define any logic and any calculus and to construct and verify proofs in it (AUTOMATH ([72, 71]), LCF ([146, 158]), nuPRL ([81]), Coq ([64, 80]),ISABELLE ([162]), KUMO (http://www.cs.ucsd.edu/groups/tatami)...). This approach is related to approaches defining consequence relations, logics, inference rules in full generality ([3, 11, 157, 87]).

4. (**"As close as possible human theorem proving"**). Designing and implementing programs in which human-style heuristics are combined with proof systems (sometimes replacing them). In general these provers are not complete (see [18, 9, 19, 21] and references therein). The logical languages used in this approach are first-order logic and subsets of second-order logic (see [19]).

In our classification above we did not mention provers for classical propositional logic because they have only played a pioneering role (Logic Theory Machine [160]) in automated deduction; today the most essential theoretical and practical work on this logic is focused on the SAT problem (and SAT solvers) (see also chapter **??**).

The work on non-classical logics can be included in 1 above. This holds for the *direct* approach (specialized and also parameterized provers have been implemented for different logics), and for *translation* approach, where the problem in non-classical logic is translated to a problem in a fragment of first-order logic (see [42, 161]).

1.4 Model Building in Automated Deduction

It is worth looking for more abstract classification criteria: We think that *decidability* is a good one[10]. Two kinds of problems have been mainly attacked so far: decidable– and semidecidable (but undecidable) ones.

It is revealing that the first published work reporting on a non-propositional theorem prover was that by M. Davis dealing with the mechanization of a decidable fragment of arithmetic [67]. Gilmore made the first step in the mechanization of a semidecidable problem [99] (that of validity in first-order logic). In this book we propose a *non–semidecidable* problem, namely that of building first-order models, as a standard topic in theorem proving [11].

It is a trivial remark that, in studying conjectures, the ability to prove or to disprove them are equally important. Some theoretical limits should be recalled when trying to automatize this process (we talk indifferently of valid formulas/countermodels (or counterexamples) or unsatisfiable formulas/models).

As the set of satisfiable first-order formulas is not recursively enumerable there can be no "universal" model building procedure. Even in this formal sense *automated model building* is substantially more complex than just proving theorems (the set of valid formulas is recursively enumerable). But despite this barrier, as we will point out below, automated model building is by no means a hopeless and futile enterprise. On the contrary, the authors believe that a more concentrated investigation of satisfiable (non-valid) problems within deduction should lead to a deeper understanding of inference as a whole, enlarging the scope of automated deduction.

Proving theorems and constructing (counter-) models for theories (i.e. set of formulas) are activities which lie at the very heart of mathematics and even of science in general.

Maybe the best example illustrating the importance of model building is the famous problem concerning the "axiom of parallel" and the discovery of non-Euclidean geometry. The inventors of these geometries proposed interpretations that are models of the other axioms and counter-models of the axiom of parallel.

This example is interesting from different points of view, mathematical,

[10]This topic is of course related to that of *computation*. To enlighten more and more the relationships between deduction and computation (as for example in [114]) is of the highest importance in mechanizing deduction.

[11]Obviously, algorithmic methods will only work for particular subclasses of this problem.

historical and philosophical ones.

Long time before the discovery of non-Euclidean geometry the status of the axiom of parallel (or 5th postulate) was surrounded by some "mystery" and the understanding of its nature in the structure of geometry was intriguing. It is therefore not exaggerated to say, in a modern terminology, that this axiom was an "open problem" since many centuries. In fact, already in his "Comments" to the book I of Euclid the Greek mathematician and astronomer Proclus promoted the idea of the formulation of the axiom which is most popular nowadays.

A continuous investigation of this axiom can also be found in the Arabic tradition (from the 9th through the 13th century). The mathematician and poet O. al Khayyam (9th - 12th century) and the astronomer at-Tusi were among the most important Arabic scientist who contributed to this matter. Though they did not contribute, strictly speaking, to non-Euclidean geometry they should not be forgotten from a historical point of view: al Khayyam and at-Tusi used in their research the quadrilateral used later by G. Saccheri and J.H. Lambert (18th century), who first considered the possibility of negating the 5th postulate. The work of at-Tusi was translated in Europe at the end of the 17th century.

More or less simultaneously with A-M. Legendre, C.F. Gauss analyzed this problem in depth, but did not publish anything about his hyperbolic geometry (according to F. Klein). Independently N.I. Lobatchevski and J. Bolyai discovered hyperbolic geometry too; but the importance of their work was not recognized at their time. Indeed, Gauss seems to be the only person appreciating the work of Lobatchevski.

The other possible geometry (anticipated by Saccheri and Lambert), the so-called elliptic geometry, was defined later by B. Riemann. Concerning the history of the problem of parallels see e.g. [205].

The discovery of non-Euclidean geometries has been considered by the great philosopher and logician H. Putnam in his book *Mathematics, Matter and Method* as the most important event in the history of science from an epistemological point of view, because it "shows" that mathematics and empirical sciences are not truly disjoints domains, or stated differently, the mathematical statements are not true in a pure analytical way. [12] Putnam's opinion is mentioned because this view is relevant to automated deduction when approaches like [124, 134, 135, 104, 207, 79] are taken into account.

[12] A statement (judgement) is true in a pure analytical way if only logical analysis of the concepts involved in it is necessary to show its truth. Logical analysis does not suffice to determine the truth of synthetic judgements (as those coming from experience).

Putnam's position evolved in time from *metaphysical realism* towards (what he called) *internal or pragmatic realism.* It is interesting to mention here [100], page 133: *But nobody has ever seen the tail of a model, models are ideal (usually infinite) objects.* This kind of statement can be conciliated with a more realistic view of things. In fact, the ideal objects Girard talks about can perfectly correspond to the way our minds apprehend (what is called) the external reality.

In 20th century mathematics, several further important contributions of this type were made, e.g. the development of nonstandard analysis and Gödel's consistency proof of the continuum hypothesis.

Not only the construction of single models but also proving model-theoretic properties became an important technique in 20th century mathematical logic: if a first-order class has the finite model property, then it is decidable (i.e., the satisfiability problem is decidable); a decision procedure can be obtained by simultaneously applying a complete first-order refutation procedure and an enumeration of finite domain interpretations (the interested reader may consult [31]). Using essentially this principle, Gödel proved the decidability of the $\forall\forall\exists$-class. Gödel's result also yields an algorithmic (though inefficient) method of constructing models of the $\forall\forall\exists$-class.

In [183] the number of *intended models* is used to give a very abstract and deep classification of mathematical theories:

> We may distinguish two types of theories, characterizing them by their intent. In the first type, exemplified by Euclidean geometry, arithmetic and set theory there is a single intended interpretation, a standard model. ...In the second type of axiomatic theory, exemplified by group theory and point-set topology, the purpose and power of the theory lie in the large number of intended models. [183], page 125.

The usefulness of (counter-)models can be also exemplified in simpler topics, as for example, in showing the non-truth-functionality of modal connectives such as possibility and necessity. In order to prove that some reasoning patterns valid in classical logic are not applicable in non-classical logics, examples (i.e. particular situations or models) are exposed in which, if subsentences (of a given sentence) are replaced by other sentences with the same truth value, the overall truth value of the original sentence changes.

Epistemic logic furnishes another example ("...Initially, a player may consider possible all deals consistent with the cards in her hand...", [88], page 16).

The list of examples from different domains could be easily enlarged.

Roughly speaking counter-examples do not have the same reputation as proofs, because they do not deal with the "general case", but their mathematical and pedagogical value cannot be overestimated. [13]

In particular, counter-examples serve the purpose to convince everybody of the *necessity* of each hypothesis in the statement of a theorem. One major goal of counter-examples is their irreplaceable role in the correction of wrong intuitions (see for example [111]) or in testing conjectures (see A. Wiles' opinion in [124]). This capability is undoubtedly needed for a *deep* understanding of proofs (and therefore of mathematics), and surely is missing in present day theorem provers.

The concept of model is at the heart of the Tarskian notion of *logical consequence* (see for example [82] and references therein). It is (implicitly) also of central importance to probability, for example in the notion of *possible event*, i.e. an event that can be *realized* in an experience, for example to have 3 points throwing a dice [14].

Already in the early days of automated deduction, *model-based inference* was recognized as a powerful tool in proof–search; we just mention the GTM geometry prover of Gelernter et al. [98] and J. Slagle's semantic resolution [196]. Loveland [144] remarks, that already in 1956, "Minsky made the observation that the diagram that traditionally accompanies plane geometry problems is a simple model of the theorem that could greatly prune the proof search."

Thus besides serving as counterexamples (as in non-Euclidean geometry) models play an important role in inference itself: they allow introducing *semantics* into a basically syntactic process. Despite the prominent role of model-based techniques in the early days of automated deduction, the following phase of research in this field was characterized by optimization of refinements and the corresponding completeness proofs. Theorem provers were mainly understood as inference engines producing proofs for provable sentences; the problem of dealing with nonderivable sentences (represented by satisfiable sets of clauses in resolution theorem proving), received considerably less attention. Some striking results have been obtained on model building using techniques exploiting exclusively *deductive* capabilities of theorem provers [210].

That inference systems can do more than just produce proofs was demon-

---

[13] Concerning the pedagogical importance of studying topics such as *consistency, non-consequence, independence*, the interested reader can consult [12].

[14] An impossible event, on the contrary, cannot be realized in an experience, for example, to have 7 points throwing a dice.

strated by S.Y. Maslov [152] and later by W.J. Joyner [128].[15] Maslov constructed a computational calculus capable not only of proving theorems but also of deciding (the validity problem of) first-order sentences belonging to specific first-order classes. He even proved the decidability of the K-class by this method, thus obtaining a strong and new mathematical result. On formulas of the K-class, the calculus, commonly called the inverse method, produces only finitely many derivations; if none of them is a proof then the sentence under investigation is not provable at all. Indeed, an inference system can be used to prove that *sentences are **not** derivable!!* Although, in case of nonderivability, Maslov's method yields the existence of countermodels, *no such model is actually constructed*.

Joyner realized the same idea within the resolution calculus; on some decidable clause classes (among them clause forms of well-known prenex classes) specific ordering refinements of resolution terminate and thus provide decision procedures for these classes. If the refinement terminates without producing the empty clause, then the original set of clauses is satisfiable; but, as in the case of the inverse method, it does not produce a model.

Not only the inverse method and resolution, but also some Gentzen-type calculi were designed to provide decision procedures, e.g. Kleene's G3 for intuitionistic propositional calculus [131].

So the next step in the "development" of inference systems can be defined as that of model construction. It is the main purpose of this book to present and analyze such inference systems and to demonstrate their value to theorem proving and to science in general.

In the same sense that proofs are more than just provability, models are more than the fact of satisfiability. Indeed, both proofs and models provide *evidence*, i.e., they show *why* statements hold or do not. This underlines the conceptual value of model construction in general. The problem remains, to which extent the construction of models can be calculized and automatized at all. As already pointed out, it is impossible, even theoretically, to realize "universal" model builders for first-order logic.

However this does not imply that algorithmic model construction is pointless *a priori*. On the contrary our aim is to show that automated model building is a reasonable and even realistic task. Sometimes this task can be carried out by ordinary resolution theorem provers, in other cases suitable extensions of the calculi are necessary.

[15] An obviously much simpler use of inference systems is proposed in [41] to *deduce* non-theorems in *classical propositional calculus*. The author gives a Hilbert system for *non-theorems*. The idea is related to that of "non consequence" (see Chapter **??**).

Basically we distinguish two types of methods for model construction:

- *enumeration (or verification) based methods:* ground instances are sequentially generated and checked.
- *deduction based methods:* new, in general nonground, formulae are inferred in the model building process. We include here consequence relations (in the standard sense) as well as others (which we call weakly sound).

We call the second type of methods *symbolic* (with the meaning – among other ones – of representing general patterns). In contrast, the enumeration methods are called *nonsymbolic*.

Enumeration can be considered, of course, as a very particular case of deduction; sometimes a combination of both approaches is the most effective one (e.g. EQMC method [**?**, 37], and hyper-linking [50, 51]) [16].

Enumerative model construction corresponds to exhaustive finite domain search for models. On the other hand, deductive methods are based on calculi producing syntactic model representations (mostly of Herbrand models) in some logical language. In this context we mainly present resolution, paramodulation and the RAMC-calculus as well as extended versions of tableaux; all these calculi are also "ordinary" refutational calculi but are modified in a way to *extract more logical information*. We do not claim that, so far, automated model building produced spectacular results; neither do we suggest that the field is fully developed and established. But we hope to demonstrate that building models is *as important as proving theorems* and that there exist reasonably efficient algorithmic methods which can and should be used in the practice of automated theorem proving. We are convinced that tools for model construction will be part of any standard theorem prover in the near future. Already in principle, constructing models and, more generally, model–based inference are crucial to any intelligent deductive method; the investigation of these mechanisms presents a major challenge to computer science and logic.

Perhaps the most natural, systematic, and elegant way of building models (counter-examples) is defined by the tableau method (due to Beth, Smullyan, Hintikka, . . . see for example [200, 95]), but in practice only trivial models can be constructed using tableaux (due to nontermination).

[16] This is in some sense foreseeable: in order to obtain general patterns some particular features are put aside. When treating a specific problem, taking into account its particularities may greatly facilitate the solution.

In order to compare tableaux with the methods presented in this work, it is worth mentioning that finite models can be found by enumeration. This feature is particularly relevant when we are interested in effective mechanization.

The book is organized as follows:

The chapters **??** and **??**, on deductive and symbolic model building methods, form the real heart of this book.

In our first approach, we present traditional resolution provers as decision procedures on particular first-order classes, where model building takes place as a postprocessing procedure. In particular we present new versions of the hyperresolution method of C. Fermüller and A. Leitsch and their extension to equational clause logic (by integrating ordered paramodulation). The model building procedures are based on deductive closure and unit selection and do not require any form of backtracking. The objects produced by these inference procedures are finite set of atoms which can be interpreted as representations of Herbrand models. In many cases (which are syntactically characterized), these Herbrand models can be transformed to finite models. The corresponding transformations are based on the analysis of equivalence classes of the Herbrand universe and do not require any form of search.

Our second approach to symbolic model building is the constraint based one. First we present the constraint language, equational logic, in more detail. Then we show how equational formulae can be transformed to (so called) equational problems, and we define algorithmic methods (related to those of Comon and Lescanne) to solve these problems. The constraint language provides an extension of clause logic, the so called *c-clause logic*, where equational constraints restrict the sets of ground instances. We demonstrate that in c-clause logic, constraints are not only useful to inference, but are equally valuable in defining *disinference* rules, i.e. rules characterizing instances that *cannot* be inferred from given premises (like disresolution and dissubsumption). We present the method RAMC by R. Caferra and N. Zabel, which incorporates these disinference principles into a resolution calculus for c-clauses. The results produced by RAMC are either refutations of unsatisfiable sets of c-clauses, or else satisfiable sets of unit c-clauses. We define two versions of RAMC, one with a strong capability for redundancy deletion, and compare these methods to hyperresolution. The original version of RAMC, having stronger expressive means for model construction (via the additional use of constraints), suffers from weaker termination (compared to hyperresolution). It is shown how both methods (which are incomparable) can be combined for the construction of more powerful model building algorithms.

The principle underlying constrained resolution and disinference rules can be naturally applied to tableaux. We describe the method RAMCET developed by Caferra, Zabel [40] and Peltier [166], which uses the constraint language for reducing redundancies. It is based on the use of a semantic strategy for pruning the search space and on inductive generalization for detecting potentially infinite branches. An interesting property of RAMCET, besides its soundness and refutational completeness, is its completeness w.r.t. the set of models that can be represented by constrained atoms.

In Chapter **??** we address a deep and central problem of symbolic model building, namely model representation. In enumerative finite model building this problem is trivial, at least from a theoretical point of view, as models can be represented by finite tables. In symbolic model building, a *formal expression* is needed to define a (generally infinite) model. In case of hyperresolution the expression is a conjunction of (closed) atoms, in case of RAMC a conjunction of constrained atoms. We prove that these representations enjoy all required computational properties, including algorithmic clause evaluation and decidability of the equivalence problem; in particular we present a generalized version of H-subsumption and of the corresponding evaluation algorithm by Fermüller and Leitsch [89]. However, atomic and constrained atomic representations are not strong enough to cover some quite simple cases of model specification. Here we mention alternative representation mechanisms, like regular tree grammars and term schematizations, and show how far the formalism can be pushed under preservation of desirable computational properties. In particular, we focus on the evaluation problem for clauses for different symbolic representations, which is an important feature of every model building formalism.

The last chapter is devoted to finite model building.

The main theoretical results enlightening conceptual differences between finite and infinite models are recalled. Different fields are mentioned in which techniques for finite model building are an important tool. We suggest the study of a particular class of problems that shed some light on the limits of enumeration methods and underline the importance of one of the key concepts studied in the present work, i.e. model representation formalisms. Several of the most representative approaches to finite model building are described. Finally, practical results attained with running systems for model building are evoked.

The areas in which the most important results have been obtained are Mathematics and Logic. Applications of model building to the semantic guiding of theorem provers are mentioned. The usefulness of model building as a help in correcting programs is illustrated by examples coming from programming teaching and simple program verification. Both domains show the potentials as well as the present limits of model building systems.

Bibliography

[1] A. Amaniss, M. Hermann, and D. Lugiez. Etude comparative des méthodes de schématisation de séquences infinies de termes du premier ordre. Research Report 93-R-114, Centre de Recherche en Informatique de Nancy, 1993.

[2] K. Appel and W. Haken. The Four-color Problem. In L. A. Steen, editor, *Mathematics Today, Twelve Informal Essays*, pages 153–180. Springer-Verlag, 1978.

[3] A. AVRON. Simple consequence relations. *Information and Computation*, 92:105–139, 1991.

[4] F. Baader and W. Snyder. Unification theory. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 8, pages 445–532. Elsevier Science, 2001.

[5] M. Baaz, 1996. Personal communication.

[6] M. Baaz, U. Egly, and A. Leitsch. Normal form transformations. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 5, pages 273–333. Elsevier Science, 2001.

[7] M. Baaz and A. Leitsch. Complexity of resolution proofs and function introduction. *Annals of Pure and Applied Logic*, 20:181–215, 1992.

[8] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 1(4):217–247, 1994.

[9] A. Ballantyne and W. Bledsoe. On generating and using examples in proof discovery. *Machine Intelligence*, 10:3–39, 1982.

[10] H. P. Barendregt. *The Lambda calculus. Its Syntax and Semantics*, volume 103 of *Studies in Logic and The Foundations of Mathematics*. North Holland, 1984.

[11] J. Barwise and S. F. (editors). *Model-theoretic logics*. Perspectives in Mathematical Logic. Springer-Verlag, 1985.

[12] J. Barwise and J. Etchemendy. *Hyperproof*, volume 42 of *CSLI Lecture Notes No. 42*. CSLI Publications, 1994.

[13] P. Baumgartner, U. Furbach, and I. Niemelä. Hyper-tableaux. In *Logics in AI, JELIA'96*. Springer, 1996.

[14] M. Beeson. Computerizing mathematics: Logic and computation. In R. Herken, editor, *The Universal Turing Machine, A Half-Century Survey*, pages 191–225. Oxford University Press, 1988.

[15] M. J. Beeson. *Foundations of constructive mathematics*, volume 6 of *Modern surveys in Mathematics*. Springer-Verlag, 1985.

[16] E. T. Bell. *Men of Mathematics. The lives and achievements of the great mathematicians from Zeno to Poincaré*. Simon and Schuster, 1986.

[17] E. Birnbaum and E. L. Lozinskii. The good old Davis-Putnam procedure helps counting models. *Journal of Artificial Intelligence Research*, 10:457–477, 1999.

[18] W. Bledsoe. Non-resolution theorem proving. *Artificial Intelligence*, 9:1–35, 1977.

[19] W. Bledsoe. Some automatic proofs in analysis. In W. W. Bledsoe and D. W. Loveland, editors, *Automated Theorem Proving: After 25 Years*, chapter 89–118, pages 1–45. American Mathematical Society, 1984.

[20] W. Bledsoe and D. Loveland. *Automated Theorem Proving after 25 years*, volume 29 of *Contemporary Mathematics*. American Mathematical Society, providence, RI, USA, 1984.

[21] W. W. Bledsoe. Some thoughts on proof discovery. In *Proc. of the 1986 Symposium on Logic Programming*, pages 2–10, 1986. The Computer Society of the IEEE.

[22] B. Bogaert and S. Tison. Equality and disequality constraints on direct subterms in tree automata. In *Symposium on Theoretical Aspects of Computer Science*, pages 436–449, 1992. LNCS 577.

[23] G. Boolos. Trees and Finite Satisfiability: Proof of a Conjecture of Burgess. *Notre Dame Journal of Formal Logic*, 25(3):193–197, July 1984.

[24] G. Boolos and R. Jeffrey. *Computability and logic*. Cambridge University Press, 1974.

[25] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.

[26] T. Boy de la Tour. On the complexity of finite sorted algebras. In R. Caferra and G. Salzer, editors, *Automated Deduction in Classical and Non-Classical Logics*, Lecture Notes in Artificial Intelligence 1761, pages 95–108. Springer Verlag, 2000.

[27] T. Boy de la Tour. Some techniques of isomorph-free search. In *Artificial Intelligence and Symbolic Computation, Internationa l Conference AISC'2000*, pages 240–252. Springer Verlag, 2000. Lecture Notes in Artificial Intelligence 1930.

[28] R. S. Boyer and J. S. Moore. *A computational logic*. Academic Press, 1979.

[29] F. Bry and A. Yahya. Minimal model generation with positive unit hyper-resolution tableaux. In *Proceeding of Tableaux'96*, LNAI 1071, pages 143–159. Springer, 1996.

[30] B. Buchberger. Grbner-bases: An algorithmic method in polynomial ideal theory. In N. Bose, editor, *Multidimensional Systems Theory*, pages 184–232. Reidel Publ. Comp., 1985.

[31] A. Bullock and H. Schneider. On Generating the Finitely Satisfiable Formulas. *Notre Dame Journal of Formal Logic*, 14(3):373–376, July 1973.

[32] R. Caferra and M. Herment. A generic graphic framework for combining inference tools and editing proofs and formulae. *Journal of Symbolic Computation*, 19(2):217–243, 1995.

[33] R. Caferra and N. Peltier. Model Building and Interactive Theory Discovery. In *Proceeding of Tableaux'95*, LNAI 918, pages 154–168. Springer, 1995.

[34] R. Caferra and N. Peltier. Combining inference and disinference rules with enumeration for model building. Workshop on model-based reasoning. IJCAI'97, 1997.

[35] R. Caferra and N. Peltier. A new technique for verifying and correcting logic programs. *Journal of Automated Reasoning*, 19(3):277–318, 1997.

[36] R. Caferra and N. Peltier. *Disinference rules, model building and abduction.* "Logic at work. Essays dedicated to the memory of Helena Rasiowa" (Part 5:Logic in Computer Science, Chap. 20). Physica-Verlag, 1998.

[37] R. Caferra and N. Peltier. Combining enumeration and deductive techniques in order to increase the class of constructible infinite models. *Journal of Symbolic Computation*, 29:177–211, 1999.

[38] R. Caferra and N. Peltier. The Connection Method, Constraints and Model Building. In S. Holldöbler, editor, *Intellectics and Computational Logic, papers in honor of Wolfgang Bibel*, pages 67–84. Kluwer Academic Publishers, 2000. Applied Logic Series, Volume 19.

[39] R. Caferra and N. Zabel. A method for simultaneous search for refutations and models by equational constraint solving. *Journal of Symbolic Computation*, 13:613–641, 1992.

[40] R. Caferra and N. Zabel. Building models by using tableaux extended by equational problems. *Journal of Logic and Computation*, 3:3–25, 1993.

[41] X. Caicedo. A Formal System for the Non-Theorems of the Propositional Calculus. *Notre Dame Journal of Formal Logic*, 19(1):147–151, January 1978.

[42] L. Catach. TABLEAUX: a general theorem prover for modal logic. *Journal of Automated Reasoning*, 7(4):489–510, 1991.

[43] S. Ceri, G. Gottlob, and L. Tanca. *Logic programming and databases.* Springer, 1990.

[44] C. Chang and H. J. Keisler. *Model Theory*, volume 73 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1990, Third edition.

[45] C. L. Chang. Theorem proving with variable-constrained resolution. *Information Sciences*, 4:217–231, 1972.

[46] C.-L. Chang and R. C.-T. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Computer Science and Applied Mathematics. Academic Press, 1973.

[47] H. Chen and J. Hsiang. Logic programming with recurrence domains. In *Automata, Languages and Programming (ICALP'91)*, pages 20–34. Springer, LNCS 510, 1991.

[48] H. Chen, J. Hsiang, and H. Kong. On finite representations of infinite sequences of terms. In *Conditional and Typed Rewriting Systems, 2nd International Workshop*, pages 100–114. Springer, LNCS 516, June 1990.

[49] S.-C. Chou. *Mechanical Geometry Theorem Proving*. Mathematics and its Applications. D. Reidel, 1988.

[50] H. Chu and D. Plaisted. Model finding in semantically guided instance-based theorem-proving. *Fundamenta Informaticae*, 21:221–235, 1994.

[51] H. Chu and D. A. Plaisted. Semantically guided first-order theorem-proving using hyper-linking. In *Proc. of CADE-12*, pages 192–206. Springer, 1994. LNAI 814.

[52] S. Cocco, O. Dubois, J. Mandler, and R. Mondasson. Au seuil de la complexité combinatoire. *Pour la Science*, (295):52–60, 2002.

[53] H. Comon. *Unification et Disunification. Théorie et Applications*. PhD thesis, INPG, Grenoble, 1988.

[54] H. Comon. Equational formulas in order-sorted algebra. In *ICALP'90*. Springer, July 1990.

[55] H. Comon. Disunification: a survey. In Jean-Louis Lassez, Gordon Plotkin, editors. Computational logic: Essays in Honor of Alan Robinson. MIT Press, 1991.

[56] H. Comon. On unification of terms with integer exponents. Technical report, LRI, Orsay, France, 1992.

[57] H. Comon. Complete axiomatizations of some quotient term algebras. *Theoretical Computer Science*, 118(2):167–191, September 1993.

[58] H. Comon. On unification of terms with integer exponents. *Mathematical System Theory*, 28:67–88, 1995.

[59] H. Comon and C. Delor. Equational formulae with membership constraints. *Information and Computation*, 112(2):167–216, August 1994.

[60] H. Comon, M. Haberstrau, and J.-P. Jouannaud. Syntacticness, cycle-syntacticness and shallow theories. *Information and Computation*, 111(1):154–191, May 1994.

[61] H. Comon and P. Lescanne. Equational problems and disunification. *Journal of Symbolic Computation*, 7:371–475, 1989.

[62] L. Console, D. T. Dupre, and P. Torasso. On the Relationship between Abduction and Deduction. *Journal of Logic and Computation*, 1(5):661–690, 1991.

[63] R. L. Constable. Formal theories and software systems: Fundamental connections between computer science and logic. Department of Computer Science Cornell University, April 1993.

[64] T. Coquand and G. Huet. The calculus of construction. *Information and Computation*, 76:95–120, 1988.

[65] P. D., H. Prawitz, and N. Vochera. A mechanical proof procedure and its realization in an electronic computer. *Journal of the Association for Computing Machinery*, 7(1-2):102–128, 1960.

[66] M. Davis. Eliminating the irrelevant from mechanical proofs. In *Proc. Symposia of Applied Mathematics, Vol. 15*, pages 15–30. American Mathematical Society, 1963.

[67] M. Davis. A computer program for Presburger arithmetic. In J. Siekmann and G. Wrighston, editors, *Automation of Reasoning 1*, pages 41–48. Springer, 1983.

[68] M. Davis. A computer program for Presburger's algorithm. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning, vol. 1*, pages 41–48. Springer-Verlag, 1983. Originally published in 1957.

[69] M. Davis. The early history of automated deduction. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, chapter 1, pages 3–18. North-Holland, 2001.

[70] P. J. Davis and R. Hersh. *L'Univers Mathématique*. Gauthier-Villars, 1985.

[71] N. G. de Bruijn. A plea for weaker frameworks. In G. Huet and G. Plotkin, editors, *Logical Frameworks*, pages 40–67. Cambridge University Press, 1991.

[72] N. G. deBruijn. A survey of the project AUTOMATH. In J. Seldin and J. Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 579–606. Academic Press, 1980.

[73] M. Denecker and D. de Schreye. On the duality of abduction and model generation in a framework for model generation with equality. *Theoretical Computer Science*, 122:225–262, 1994.

[74] B. Dreben and W. D. Goldfarb. *The Decision Problem, Solvable Classes of Quantificational Formulas.* Addison-Wesley, 1979.

[75] O. Dubois and R. M. (eds.). Phase transition in combinatorial problems. *Theoretical Computer Science*, 265(1,2), 2001.

[76] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Perspectives in Mathematical Logic. Springer, 1999. Second Revised and Enlarged Edition.

[77] E. Eder. *Relative Complexities of First-Order Logic Calculi.* Vieweg, 1990.

[78] H. Enderton. *A Mathematical Introduction to Logic.* Academic Press, 1972.

[79] D. Epstein and S. Levy. Experimentation and Proof in Mathematics. *Notices of the AMS*, 42(6):670–674, 1995.

[80] G. D. et al. The Coq proof assistant user's guide, version 5.6. Rapport Technique 134, INRIA, Décembre 1991.

[81] R. L. C. et al. *Implementing Mathematics with the Nuprl proof development system.* Prentice-Hall, 1986.

[82] J. Etchemendy. Tarski on truth and logical consequence. *The Journal of Symbolic Logic*, 53(1):51–79, March 1988.

[83] R. Fagin. Monadic generalized spectra. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 21:89–96, 1975.

[84] R. Fagin. A spectrum hierarchy. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 21:123–134, 1975.

[85] R. Fagin. Probabilities on Finite Models. *Journal of Symbolic Logic*, 41(1):50–58, March 1976.

[86] R. Fagin. Finite model theory. a personal perspective. *Theoretical Computer Science*, 116:3–31, 1993.

[87] R. Fagin, J. Halpern, and M. Vardi. What is an inference rule? *The Journal of Symbolic Logic*, 57(3):1018–1045, September 1992.

[88] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. The MIT Press, 1995.

[89] C. Fermüller and A. Leitsch. Hyperresolution and automated model building. *Journal of Logic and Computation*, 6(2):173–203, 1996.

[90] C. Fermüller and A. Leitsch. Decision procedures and model building in equational clause logic. *Journal of the IGPL*, 6(1):17–41, 1998.

[91] C. Fermüller, A. Leitsch, U. Hustadt, and T. Tammet. Resolution decision procedures. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, chapter 25, pages 1791–1849. North-Holland, 2001.

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

[93] C. G. Fermüller and A. Leitsch. Model building by resolution. In *Computer Science Logic, CSL'92*, pages 134–148. Springer, LNCS 702, 1992.

[94] S. Fèvre and N. Peltier. Two problems in geometry solved by using automated model builders. AAR Newsletter 38, 1997.

[95] M. Fitting. *First-Order Logic and Automated Theorem Proving*. Texts and Monographs in Computer Science. Springer-Verlag, 1990.

[96] M. Fujita and Hasegawa. A model generation theorem prover in KL1 using a ramified stack algorithm. In *Proceedings of 8th International Conference Symp. Logic Programming*, pages 1070–1080, 1991.

[97] H. Gelernter. A note on syntactic symmetry and the manipulation of formal systems by machine. *Information and Control*, 2:80–89, 1959.

[98] H. Gelernter. Realization of a Geometry-Theorem Proving Machine. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning*, chapter Vol. 1, pages 99–122. Springer-Verlag, 1983. First published in 1959.

[99] P. Gilmore. A Proof Method for Quantification Theory: its Justification and Realization. *The IBM Journal*, pages 28–35, 1960.

[100] J.-Y. Girard. From foundations to Ludics. *The Bulletin of Symbolic Logic*, 9(2):131–168, June 2003.

[101] J. Goguen. *Theorem Proving and Algebra*. MIT-press. to appear.

[102] J. Goguen, K. Lin, G. Rosu, A. Mori, and B. Warinschi. An overview of the tatami project. In K. Futatsugi, editor, *Cafe: An Industrial-Strength Algebraic Formal Method*, pages 61–78. Elsevier, 2000.

[103] N. D. Goodman. The knowing mathematician. *Synthèse*, 60:21–38, 1984.

[104] N. D. Goodman. Mathematics as natural science. *The Journal of Symbolic Logic*, 55(1):182–193, March 1990.

[105] G. Gottlob and R. Pichler. Working with ARMs: Complexity results on atomic representations of Herbrand models. *Information and Computation*, 2001. 25 pages, to appear.

[106] G. Grätzer. *Universal Algebra*. Springer-Verlag, 1979. Second Edition.

[107] Y. Gurevich. Toward logic tailored for computational complexity. In *Computation and Proof Theory, M.M. Richter et al. eds.*, pages 175–216. Springer-Verlag, 1984. LNM 1104.

[108] Y. Gurevich. Logic and the challenge of computer science. In E. Börger, editor, *Current trends in theoretical computer science*, pages 1–57. Computer Science Press, 1988.

[109] T. Hailperin. A Complete Set of Axioms for Logical Formulas Invalid in Some Finite Domain. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 7:84–96, 1961.

[110] P. Hájek and T. Havránek. *Mechanizing hypothesis formation. Mathematical foundations for a general theory.* Springer-Verlag, 1978.

[111] B. Hauchecorne. *Les Contre-exemples en Mathématiques.* ellipses, 1988.

[112] L. He. I-SATCHMO: An improvement of SATCHMO. *Journal of Automated Reasoning*, 27(3):313–322, October 2001.

[113] L. Henschen and L. Wos. Unit Refutations and Horn Sets. *Journal of the Association for Computing Machinery*, 21(4):590–605, 1974.

[114] R. Herken. *The Universal Turing Machine, A Half-Century Survey.* Oxford University Press, 1988.

[115] M. Hermann. On the relation between primitive recursion, schematization, and divergence. In *Proceeding 3rd Conference on Algebraic and Logic Programming*, pages 115–127. Springer, LNCS 632, 1992.

[116] M. Hermann. Divergence des systèmes de réécriture et schématisation des ensembles infinis de termes. Habilitation, Université de Nancy I, and CRIN-CNRS Inria-Lorraine, Nancy, France, March 1994.

[117] M. Hermann and R. Galbavý. Unification of Infinite Sets of Terms schematized by Primal Grammars. *Theoretical Computer Science*, 176(1–2):111–158, 1997.

[118] G. Hermerén. Models. In S. Stenlund, editor, *Logical theory and semantic analysis, essays dedicated to Stig Kanger on his fiftieth birthday*, pages 175–191. D. Reidel, 1974.

[119] W. Hodges. Elementary Predicate Logic. In D. Gabbay and F. G. (eds.), editors, *Handbook of Philosophical Logic*, chapter I.1, pages 2–131. D. Reidel, 1983. Vol. 1, Elements of Classical Logic.

[120] W. Hodges. *Model theory*, volume 42 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, 1993.

[121] C. J. Hogger. *Introduction to Logic Programming.* Academic Press, 1984.

[122] J. Hooker. New methods for computing inferences in first order logic. *Annals of Operations Research*, 43:479–492, 1993.

[123] J. Hooker. Solving the incremental satisfiability problem. *The Journal of Logic Programming*, 15(1 2):177–186, 1993.

[124] J. Horgan. L'ordinateur en mathématiques. *Pour la Science*, (194):84–93, Décembre 1993.

[125] J. Hsiang and M. Rusinowitch. Proving refutational completeness of theorem proving strategies: The transfinite semantic tree method. *Journal of the ACM*, 38(3), July 1991.

[126] R. Jeroslow. Computation-oriented reductions of predicate to propositional logic. *Decision Support Systems*, 4:183–197, 1988.

[127] J. Jouannaud and C. Kirchner. Solving equations in abstract algebras: a rule based survey of unification. In J.-L. Lassez and G. Plotkin, editors, *Essays in Honor of Alan Robinson*, pages 91–99. The MIT-Press, 1991.

[128] W. Joyner. Resolution strategies as decision procedures. *Journal of the ACM*, 23:398–417, 1976.

[129] R. J. B. Jr. and J. Pehoushek. Counting models using connected components. In *Proc. of AAAI 2000*, 2000.

[130] R. Khardon and D. Roth. Reasoning with models. In *AAAI-94*, pages 1148–1153. The MIT Press, 1994.

[131] S. C. KLEENE. *Introduction to Metamathematics*. North-Holland, 1952.

[132] D. Knuth and P. Bendix. Simple word problems in universal algebra. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[133] B. Krishnamurthy. Short proofs for tricky formulas. *Acta Informatica*, 22:253–275, 1985.

[134] C. Lam. How reliable is a computer-based proof? *The Mathematical Intelligencer*, 12(1):8–12, 1990.

[135] C. Lam. The search for a finite projective plane of order 10. *The American Mathematical Monthly*, 98(4):305–318, April 1991.

[136] J. Lassez, M. Maher, and K. Marriott. Unification Revisited. In *Foundations of Logic and Functional Programming*, pages 67–113. Springer, 1986. LNCS 306.

[137] S. Lee and D. Plaisted. Problem solving by searching for models with a theorem prover. *Artificial Intelligence*, 69:205–233, 1994.

[138] S.-J. Lee and D. Plaisted. Eliminating duplication with the hyperlinking strategy. *Journal of Automated Reasoning*, 9:25–42, 1992.

[139] A. Leitsch. Deciding clause classes by semantic clash resolution. *Fundamenta Informaticae*, 18:163–182, 1993.

[140] A. Leitsch. *The resolution calculus.* Springer. Texts in Theoretical Computer Science, 1997.

[141] G. Lloyd. *Origines et Développement de la Science Grecque*, volume 365 of *Champs*. Flammarion, 1990.

[142] D. Loveland, D. Reed, and D. S. Wilson. SATCHMORE: SATCHMO with RElevancy. *J. of Automated Reasoning*, 14(2):325–351, April 1995.

[143] D. W. Loveland. *Automated Theorem Proving: A Logical Basis*, volume 6 of *Fundamental Studies in Computer Science*. North Holland, 1978.

[144] D. W. Loveland. Automated theorem proving: a quarter century review. In W. W. Bledsoe and D. W. Loveland, editors, *Automated Theorem Proving: After 25 Years*, pages 1–45. American Mathematical Society, 1984.

[145] J. Lukasiewicz. *La Syllogistique d'Aristote*. Philosophies pour l'âge de la science. Armand Colin, 1972.

[146] A. M. M. Gordon and C. Wadsworth. *Edinburgh LCF: A mechanized Logic of Computation.* Springer, LNCS 78, New York (NY, USA), 1979.

[147] A. Mal'cev. *The Metamathematics of Algebraic Systems: Collected Papers 1936–1967*, chapter Axiomatizable classes of locally free algebra of various type, pages 262–281. Benjamin Franklin Wells editor, North Holland, 1971. chapter 23.

[148] R. Manthey and F. Bry. SATCHMO: A theorem prover implemented in Prolog. In *Proc. of CADE-9*, pages 415–434. Springer, LNCS 310, 1988.

[149] K. Marriott and J. Stuckey. *Programming with Constraints, an Introduction.* The MIT Press, 1998.

[150] A. Martelli and U. Montanari. Unification in linear time and space: a structured presentation. Technical Report B76-16, Istituto di elaborazione della informazione, 1976.

[151] P. Martin-Löf. Truth of a proposition, evidence of a judgement, validity of a proof. *Synthèse*, 73:407–420, 1987.

[152] S. Maslov. The inverse method for establishing deducibility for logical calculi. *Proc. Steklov Inst. Math.*, 98:25–96, 1968.

[153] R. Matzinger. Computational representations of Herbrand models using grammars. In *Computer Science Logic, CSL'96*, 1997.

[154] R. Matzinger. *Computational Representations of Models in First-Order Logic.* PhD thesis, TU-Vienna, 2000.

[155] W. McCune and R. Padmanabham. *Automated Deduction in Equational Logic and Cubic Curves*, volume 1095 of *LNAI*. Springer, 1996.

[156] R. Mckenzie. On spectra, and the negative solution of the decision problem for identities having a finite nontrivial model. *The Journal of Symbolic Logic*, 40(2):186–196, June 1975.

[157] J. Meseguer. General logics. In *Proc. of Logic Colloquium'87*, pages 275–330. Nort-Holland, 1989.

[158] R. Milner. The use of machines to assist in rigorous proof. In C. Hoare and E. J.C. Shepherdson, editors, *Mathematical Logic and Programming Languages*, pages 77–88. Prentice-Hall, 1985.

[159] J. Mongy. *Transformation de noyaux reconnaissables d'arbres. Forêts RATEG.* PhD thesis, Laboratoire d'Informatique Fondamentale de Lille, Villeneuve d'Ascq, France, 1981.

[160] A. Newell, J. Shaw, and H. Simon. Empirical explorations with the logic theory machine. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning. Classical papers on computational logic*, pages 49–73. Springer-Verlag, 1983. Originally published in 1957.

[161] H. Ohlbach, A. Nonnegart, M. de Rijke, and D. M. Gabbay. Encoding two-valued nonclassical logics in classical logic. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, chapter 21, pages 1403–1486. North-Holland, 2001.

[162] L. C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5(3):363–398, September 1989.

[163] N. Peltier. Increasing the capabilities of model building by constraint solving with terms with integer exponents. *Journal of Symbolic Computation*, 24:59–101, 1997.

[164] N. Peltier. *Nouvelles Techniques pour la Construction de Modèles finis ou infinis en Déduction Automatique.* PhD thesis, Institut National Polytechnique de Grenoble, 1997. ftp://ftp.imag.fr/pub/Mediatheque.IMAG/theses/97-Peltier.Nicolas/.

[165] N. Peltier. *Nouvelles Techniques pour la Construction de Modèles finis ou infinis en Déduction Automatique.* PhD thesis, Institut National Polytechnique de Grenoble, 1997. ftp://ftp.imag.fr/pub/Mediatheque.IMAG/theses/97-Peltier.Nicolas/.

[166] N. Peltier. Simplifying formulae in tableaux. Pruning the search space and building models. In *Proceeding of Tableaux'97*, pages 313–327. Springer LNAI 1227, 1997.

[167] N. Peltier. Tree automata and automated model building. *Fundamenta Informaticae*, 30(1):59–81, 1997.

[168] N. Peltier. System description: an equational constraints solver. In *Proceedings of CADE-15*. Springer, 1998.

[169] N. Peltier. Pruning the search space and extracting more models in tableaux. *Logic Journal of the IGPL*, 7(2):217–251, 1999.

[170] N. Peltier. A General Method for Using Terms Schematizations in Automated Deduction. In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR'01)*, pages 578–593. Springer LNCS 2083, 2001.

[171] N. Peltier. On the decidability of the PVD class with equality. *Logic Journal of the IGPL*, 9(4):601–624, 2001.

[172] R. Pichler. Algorithms on atomic representations of Herbrand models. In *Logic in AI, JELIA'98*, pages 199–215. Springer, 1998. LNAI 1489.

[173] R. Pichler. Solving equational problems efficiently. In H. Ganzinger, editor, *Automated Deduction – CADE-16, 16th International Conference on Automated Deduction*, LNAI 1632, pages 97–111, Trento, Italy, July 7–10, 1999. Springer-Verlag.

[174] R. Pichler. *H-Subsumption in Automated Deduction*. PhD thesis, TU-Vienna, 2000.

[175] D. A. Plaisted and Y. Zhu. Ordered semantic hyperlinking. *Journal of Automated Reasoning*, 25(3):167–217, October 2000.

[176] D. Plotkin. A Further Note on Inductive Generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, pages 101–124. Edinburgh University Press, 1971.

[177] G. Polya. *How to Solve It, a New Aspect of Mathematical Method*. Princeton University Press. Second Edition, 1973.

[178] D. Prawitz. An improved proof procedure. *Theoria*, 26:102–139, 1960.

[179] D. Prawitz. Advances and problems in mechanical proof procedure. *Machine Intelligence*, 4:59–71, 1969.

[180] D. Prawitz. Proof theory and the processing of proofs. Summary draft of talk at the Fifth Conference on Automated Deduction, July 1980.

[181] H. Rasiowa and R. Sikorski. *The mathematics of metamathematics*, volume 41 of *monografie Matematyczne*. PWN Polish Scientific Publishers, 1970.

[182] S. Reeves. Semantic tableaux as framework for automated theorem-proving. In *AISB Conference*, pages 125–139, 1987.

[183] F. Richman. Intuitionism as Generalization. *Philosophia Mathematica*, 5(1/2):124–128, 1990.

[184] G. Robinson and L. Wos. Paramodulation and theorem-proving in first-order theories with equality. In D. Michie and R. Meltzer, editors, *Machine Intelligence*, volume IV, pages 135–150. Edinburg U. Press, 1969.

[185] J. Robinson. Automatic deduction with hyperresolution. *Intern. Journal of Computer Math.*, 1:227–234, 1965.

[186] J. Robinson. Proof = Guarantee + Explanation. In S. Holldöbler, editor, *Intellectics and Computational Logic, papers in honor of Wolfgang Bibel*, pages 277–294. Kluwer Academic Publishers, 2000. Applied Logic Series, Volume 19.

[187] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. Assoc. Comput. Mach.*, 12:23–41, 1965.

[188] M. Rose and K. Wilkinson. Application of model search to lattice theory. *J. Association for Automated Reasoning*, 52:3–7, August 2001.

[189] J. Rosen. *Symmetry in Science. An Introduction to the General Theory.* Springer-Verlag, 1995.

[190] D. Roth. On the hardness of approximate reasoning. *J. Artificial Intelligence*, 82(1-2):273–302, April 1996.

[191] T. Rudlof. SHR tableaux - A Framework for Automated Model Generation. *Journal of Logic and Computation*, 10(6):107–155, 2000.

[192] G. Salzer. The unification of infinite sets of terms and its applications. In *Logic Programming and Automated Reasoning (LPAR'92)*, pages 409–429. Springer, LNAI 624, July 1992.

[193] G. Salzer. Primal grammar and unification modulo a binary clause. In *Proc. of CADE-12*, pages 72–86. Springer, 1994. LNAI 814.

[194] J. Siekmann and G. Wrightson. *Automation of Reasoning. Classical Papers on Computational Logic 1957-1966*, volume 1. Springer-Verlag, 1983.

[195] J. Siekmann and G. Wrightson. *Automation of Reasoning. Classical Papers on Computational Logic 1967-1970*, volume 2. Springer-Verlag, 1983.

[196] J. R. Slagle. Automatic theorem proving with renamable and semantic resolution. *Journal of the ACM*, 14(4):687–697, October 1967.

[197] J. Slaney. Finder (finite domain enumerator): Notes and guides. Technical report, Australian National University Automated Reasoning Project, Canberra, 1992.

[198] J. Slaney. SCOTT: a model-guided theorem prover. In *Proceedings IJCAI-93*, volume 1, pages 109–114. Morgan Kaufmann, 1993.

[199] C. Smoryński. *Logical number theory I. An Introduction.* Springer-Verlag, 1991.

[200] R. M. Smullyan. *First-Order Logic.* Springer, 1968.

[201] S. Stenlund. *Combinators, λ-terms and proof theory.* Reidel, 1971.

[202] F. Stolzenburg. Loop-detection in hyper-tableaux by powerful model generation. *Journal of Universal Computer Science*, 5(3):135–155, 1999.

[203] P. Suppes. A Comparison of the Meaning and Uses of Models in Mathematics and Empirical Sciences. *Synthèse*, 12:287–301, 1966.

[204] T. Tammet. Using resolution for deciding solvable classes and building finite models. In *Baltic Computer Science*, pages 33–64. Springer, LNCS 502, 1991.

[205] R. Taton. *La science moderne de 1450 à 1800*, volume 187 of *Quadrige*. Presses Universitaires de France, 1995.

[206] B. A. Trahtenbrot. Impossibility of an Algorithm for the Decision Problem in Finite Classes. *American Mathematical Society Translations*, 23(Series 2):1–5, 1963. First published (Russian): Doklady Akadémii Nauk SSSR, vol. 70 (1950), pp. 569–572.

[207] T. Tymoczko. The four-color problem and its philosophical significance. *The Journal of Philosophy*, 76(2), February 1979.

[208] Vorobyov. An improving lower bound for the elementary theories of trees. In *Proc. of CADE-13, LNAI 1104*, pages 275–287. Springer, 1996.

[209] C. Weidenbach. Combining superposition, sorts and splitting. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 27, pages 1965–2013. Elsevier Science, 2001.

[210] S. Winker. Generation and verification of finite models and counterexamples using an automated theorem prover answering two open questions. *Journal of the ACM*, 29(2):273–284, April 1982.

[211] S. W. Winker, L. Wos, and E. L. Lusk. Semigroups, antiautomorphisms, and involutions: a computer solution to an open problem, I. *Mathematics of Computation*, 37(156):533–545, October 1981.

[212] L. Wos. *Automated Reasoning, 33 Basic Research Problems*. Prentice Hall, 1988.

[213] L. Wos and et al. An overview of automated reasoning and related fields. *Journal of Automated Reasoning*, 1(1):5–48, 1985.

[214] J. Zhang. Problems on the generation of finite models. In *Proc. of CADE-12*, pages 753–757. Springer, 1994. LNAI 814.

[215] J. Zhang and H. Zhang. SEM: a system for enumerating models. In *Proc. IJCAI-95*, volume 1, pages 298–303. Morgan Kaufmann, 1995.

[216] W. Zhang. Number of models and satisfiability of sets of clauses. *Theoretical Computer Science*, 155(1):277–288, February 1996.