Abstracts of the Journal of Automated Reasoning

Journal of Automated Reasoning, Vol. 21, no. 2 (October 1998)

Larry Wos
Automating the Search for Elegant Proofs
Vol. 21, no. 2 (1998) 135-175

Abstract: Research reported in this article was spawned by a colleague's request to find an elegant proof (of a theorem from Boolean algebra) to replace his proof consisting of 816 deduced steps. The request was met by finding a proof consisting of 100 deduced steps. The methodology used to obtain the far shorter proof is presented in detail through a sequence of experiments. Although clearly not an algorithm, the methodology is sufficiently general to enable its use for seeking elegant proofs regardless of the domain of study. In addition to (usually) being more elegant, shorter proofs often provide the needed path to constructing a more efficient circuit, a more effective algorithm, and the like. The methodology relies heavily on the assistance of McCune's automated reasoning program OTTER. Of the aspects of proof elegance, the main focus here is on proof length, with brief attention paid to the type of term present, the number of variables required, and the complexity of deduced steps. The methodology is iterative, relying heavily on the use of three strategies: the resonance strategy, the hot list strategy, and McCune's ratio strategy. These strategies, as well as other features on which the methodology relies, do exhibit tendencies that can be exploited in the search for shorter proofs and for other objectives. To provide some insight regarding the value of the methodology, I discuss its successful application to other problems from Boolean algebra and to problems from lattice theory. Research suggestions and challenges are also offered.

Keywords: automated reasoning, elegant proofs, Otter, Boolean algebra

Geoff Sutcliffe and Christian Suttner
The TPTP Problem Library: The CNF Release v.1.2.1
Vol. 21, no. 2 (1998) 177–203

Abstract: This paper provides a detailed description of the CNF part of the TPTP Problem Library for automated theorem proving systems. The library is available via the Internet and forms a common basis for development and experimentation with automated theorem provers. This paper explains the motivations and reasoning behind the development of the TPTP (thus implicitly explaining the design decisions made) and describes the TPTP contents and organization. It also provides guidelines for obtaining and using the library, summary statistics about release v1.2.1, and an overview of the tptp2X utility program. References for all the sources of TPTP problems are provided.

Keywords: TPTP, problem library, experimental evaluation

Andrei Voronkov
Proof-Search in Intuitionistic Logic with Equality,
or Back to Simultaneous Rigid E-Unification
Vol. 21, no. 3 (1998) 205–231

Abstract: We characterize provability in intuitionistic logic with equality in terms of a constraint calculus. This characterization uncovers close connections between provability in intuitionistic logic with equality and solutions to simultaneous rigid E-unification. We show that the problem of existence of a sequent proof with a given skeleton is polynomial-time equivalent to simultaneous rigid E-unifiability. This gives us a proof procedure for intuitionistic logic with equality modulo simultaneous rigid E-unification. We also show that simultaneous rigid E-unifiability is polynomial-time reducible to intuitionistic logic with equality. Thus, any proof procedure for intuitionistic logic with equality can be considered as a procedure for simultaneous rigid E-unifiability. In turn, any procedure for simultaneous rigid E-unifiability gives a procedure for establishing provability in intuitionistic logic with equality.

Keywords: intuitionistic logic, equality, rigid E-unification

Jacques Chazarain and Serge Muller
Automated Synthesis of Recursive Programs from a $\forall\exists$ Logical Specification
Vol. 21, no. 2 (1998) 233–275

Abstract: The specification of a function is often given by a logical formula, called an AE-formula, of the following form. In this paper, we consider the case in which E is an equational theory. It is divided into two parts. In the first part, we develop a theory for the automated proof of such judgments in the initial model of E. The validity in the initial model means that we consider not only equational theorems but also inductive ones. From our theory we deduce an automated method for the proof of a class of such judgments. In the second part, we present an automated method for program synthesis. We show how the previous proof method can be used to generate a recursive program for a function f that satisfies a specific judgment. We illustrate our method with the automated synthesis of some recursive programs on domains such as integers and lists. Finally, we describe our system LEMMA, which is an implementation in Common Lisp of these new methods.

Keywords: automated reasoning, program synthesis, theorem proving, test set, inductive reasoning, term rewriting systems

Journal of Automated Reasoning,
SPECIAL ISSUE ON COMPUTER ALGEBRA and AUTOMATED THEOREM PROVING,
Vol. 21, no. 3 (December 1998)

Deepak Kapur and Dongming Wang
Preface

Vol. 21, no. 3 (1998) 277–278

J. Harrison and L. Thery
A Skeptic's Approach to Combining HOL and Maple
Vol. 21, no. 3 (1998) 279–294

Abstract: We contrast theorem provers and computer algebra systems, pointing out the advantages and disadvantages of each, and suggest a simple way to achieve a synthesis of some of the best features of both. Our method is based on the systematic separation of search for a solution and checking the solution, using a physical connection between systems. We describe the separation of proof search and checking in some detail, relating it to proof planning and to the complexity class NP, and discuss different ways of exploiting a physical link between systems. Finally, the method is illustrated by some concrete examples of computer algebra results proved formally in the HOL theorem prover with the aid of Maple.

Keywords: proof checking, automated theorem proving, computer algebra, complexity theory

Andrej Bauer, Edmund Clarke, and Xudong Zhao
Analytica – An Experiment in Combining Theorem Proving
and Symbolic Computation
Vol. 21, no. 3 (1998) 295–325

Abstract: Analytica is an automatic theorem prover for theorems in elementary analysis. The prover is written in the Mathematica language and runs in the Mathematica environment. The goal of the project is to use a powerful symbolic computation system to prove theorems that are beyond the scope of previous automatic theorem provers. The theorem prover is also able to deduce the correctness of certain simplification steps that would otherwise not be performed. We describe the structure of Analytica and explain the main techniques that it uses to construct proofs. Analytica has been able to prove several nontrivial theorems. In this paper, we show how it can prove a series of lemmas that lead to the Bernstein approximation theorem.

Keywords: computer algebra, Analytica, symbolic computation, Mathematica, theorem proving

Manfred Kerber, Michael Kohlhase, and Volker Sorge
Integrating Computer Algebra into Proof Planning
Vol. 21, no. 3 (1998) 327–355

Abstract: Mechanized reasoning systems and computer algebra systems have different objectives. Their integration is highly desirable, since formal proofs often involve both of the two different tasks proving and calculating. Even more important, proof and computation are often interwoven and not easily separable. In this article we advocate an integration of computer algebra into mechanized reasoning systems at the proof plan level. This approach allows us to view the

computer algebra algorithms as methods, that is, declarative representations of the problem-solving knowledge specific to a certain mathematical domain. Automation can be achieved in many cases by searching for a hierarchic proof plan at the method level by using suitable domain-specific control knowledge about the mathematical algorithms. In other words, the uniform framework of proof planning allows us to solve a large class of problems that are not automatically solvable by separate systems. Our approach also gives an answer to the correctness problems inherent in such an integration. We advocate an approach where the computer algebra system produces high-level protocol information that can be processed by an interface to derive proof plans. Such a proof plan in turn can be expanded to proofs at different levels of abstraction, so the approach is well suited for producing a high-level verbalized explication as well as for a low-level, machine-checkable, calculus-level proof. We present an implementation of our ideas and exemplify them using an automatically solved example.

Keywords: mechanized reasoning, computer algebra, hierarchical proof planning, proof checking

Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning
A New Approach for Automatic Theorem Proving in Real Geometry
Vol. 21, no. 3 (1998) 357–380

Abstract: We present a new method for proving geometric theorems in the real plane or higher dimension. The method is derived from elimination set ideas for quantifier elimination in linear and quadratic formulas over the reals. In contrast to other approaches, our method can also prove theorems whose complex analogues fail. Moreover, the problem formulation may involve order inequalities. After specification of independent variables, nondegeneracy conditions are generated automatically. Moreover, when trying to prove conjectures that—apart from nondegeneracy conditions—do not hold in the claimed generality, missing premises are found automatically. We demonstrate the applicability of our method to nontrivial examples.

Keywords: real quantifier elimination, real geometry, automatic theorem proving over the reals

Jinzhao Wu and Zhuojun Liu
Well-Behaved Inference Rules for First-Order Theorem Proving
Vol. 21, no. 3 (1998) 381–400

Abstract: The concept of well-behaved inference rules is developed in first-order polynomial-based theorem proving. It is shown that well-behaved inference rules are complete for both the set of support strategy and a linear strategy. Furthermore, two concrete inference rules that are well behaved are presented, and two other strategies for them are described.

Keywords: inference rule, first-order polynomial, set-of-support deduction, linear deduction