07401 Abstracts Collection
Deduction and Decision Procedures
— Dagstuhl Seminar —

Franz Baader[1], Byron Cook[2], Jürgen Giesl[3] and Robert Nieuwenhuis[4]

[1] TU Dresden, DE
baader@tcs.inf.tu-dresden.de
[2] Microsoft Research, Cambridge, GB
[3] RWTH Aachen, DE
giesl@informatik.rwth-aachen.de
[4] UPC Barcelona, ES
roberto@lsi.upc.es

Abstract. From 01.10. to 05.10.2007, the Dagstuhl Seminar 07401 "Deduction and Decision Procedures" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar are put together in this paper.

Keywords. Decision Procedures, Deduction, Boolean Satisfiability, First-Order Logic, Integer Arithmetic, Combination of Theories, Satisfiability Modulo Theories, Rewrite Systems, Formal Verification, Model Finding

07401 Executive Summary – Deduction and Decision Procedures

Formal logic provides a mathematical foundation for many areas of computer science. Significant progress has been made in the challenge of making computers perform non-trivial logical reasoning. be it fully automatic, or in interaction with humans.

In the last years it has become more and more evident that theory-specific reasoners, and in particular decision procedures, are extremely important in many applications of such deduction tools. General-purpose reasoning methods such as resolution or paramodulation alone are not efficient enough to handle the needs of real-world applications.

For this reason, the focus of this seminar was on decision procedures, their integration into general-purpose theorem provers, and the application of the integrated tools in computer science.

Keywords: Formal Logic, Deduction, Artificial Intelligence

Joint work of: Baader, Franz; Cook, Byron; Giesl, Jürgen; Nieuweinhuis, Robert

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2007/1251

Improvements in Formula Generalization

Markus Aderhold (TU Darmstadt, D)

For proofs by induction it is often necessary to generalize statements to strengthen the induction hypotheses. We present improved heuristics to generalize away subterms, unnecessary conditions and function symbols in a formula. This resolves shortcomings that we encountered within an experimental evaluation of generalization heuristics from the literature. Our generalization method has been implemented in the verification tool VeriFun. An evaluation with examples from the literature as well as several case studies of our own demonstrates the success of our development.

Keywords: Induction, Proof Planning, Generalization, Disproving

Full Paper:
http://dx.doi.org/10.1007/978-3-540-73595-3_16

See also: Markus Aderhold. Improvements in Formula Generalization. In Frank Pfenning, editor, Proceedings of the 21st Conference on Automated Deduction (CADE-21), Lecture Notes in Artificial Intelligence, vol. 4603, pp. 231-246, Springer-Verlag, 2007

Axiom Pinpointing in the Description Logic $\mathcal{EL}$

Franz Baader (TU Dresden, D)

Description Logics (DL) are a successful family of logic-based knowledge representation languages, which can be used to represent the conceptual knowledge of an application domain in a structured and formally well-understood way. They are employed in various application domains, such as natural language processing, databases, the semantic web, and biomedical ontologies. As the size of DL knowledge bases grows, tools that support improving their quality become more important. Standard DL reasoning can be used to compute implicit consequences such as inconsistencies and inferred subsumption relationships, but it does not explain the reasons for a given consequence.

Axiom pinpointing is a first step towards providing such an explanation.

Given a DL knowledge base and a consequence, it computes minimal (maximal) subsets of the KB that have the consequence (do not have the consequence). In the talk, I will review recent results on pinpointing in the DL $\mathcal{EL}$. Though of limited expressive power, $\mathcal{EL}$ is used in large biomedical ontologies such as Snomed and the Gene Ontology.

Full Paper:
http://lat.inf.tu-dresden.de/research/papers/2007/BaaPenSun-KI07.pdf

See also: Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. Pinpointing in the Description Logic $\mathcal{EL}^+$. In Proceedings of the 30th German Conference on Artificial Intelligence (KI2007), LNAI 4667, Osnabrück, Germany, 2007. Springer-Verlag.

Boosting Verification by Automatic Tuning of Decision Procedures

Domagoj Babic (University of British Columbia - Vancouver, CA)

Parametrized heuristics abound in computer aided-design and verification, and manual tuning of the respective parameters is difficult and time-consuming. Very recent results from the artificial intelligence (AI) community suggest that this tuning process can be automated, and that doing so can lead to significant performance improvements; furthermore, automated parameter optimization can provide valuable guidance during the development of heuristic algorithms.

Such an AI approach can improve a state-of-the-art decision procedure for large, real-world bounded model-checking and software verification instances. The resulting, automatically-derived parameter settings yielded runtimes on average 4.5 times faster on bounded model checking instances and 500 times faster on software verification problems than extensive hand-tuning of the decision procedure. Furthermore, the availability of automatic tuning influenced the design of the prover, and the automatically-derived parameter settings provided a deeper insight into the properties of problem instances.

Keywords: Decision Procedures, Boolean Satisfiability, Search Parameter Optimization, Bit-vector Arithmetic

Joint work of: Babic, Domagoj; Hutter, Frank; Hoos, Holger; Hu, Alan

Full Paper:
http://www.cs.ubc.ca/~babic/papers/fmcad07.pdf

Theories in Context - Model Evolution Modulo Integer Arithmetic

Peter Baumgartner (NICTA - Canberra, AU)

Some applications of automated deduction require reasoning modulo some form of integer arithmetic. Unfortunately, theory reasoning support for the integers in current theorem provers is sometimes too weak for practical purposes. In particular, the family of Satisfiability Modulo Theories solvers [NOT06] lack support for quantifiers and resort to incomplete or inefficient heuristics to deal with quantified formulas. Also, theory reasoning techniques developed within first-order theorem proving are often impractical as they require enumeration of complete sets of theory unifiers (in particular those in the tradition of Stickel's Theory Resolution [Sti85]) or feature only weak or no redundancy criteria (for instance, Bürckert's Constraint Resolution [Buer90]).

In this paper we propose a novel calculus for first-order logic modulo Linear Integer Arithmetic (LIA) that avoids these problems. The calculus requires a decision procedure for the $\forall\exists$ fragment of LIA instead of a complete LIA-unification procedure, and is amenable to strong redundancy criteria. It is derived from the Model Evolution calculus [BT03], a first-order logic version of the propositional DPLL procedure. The main data structure of that calculus is the context, which provides a compact representation of certain Herbrand interpretations. The new calculus builds in its core on a version of contexts modulo LIA. More precisely, the contexts (and the calculus) deal with conservative extensions of the integers structure by free predicate and constant symbols.

In the talk we present the basic ideas of the calculus and its theoretical properties, restricted for now to the case of free predicate and constant symbols ranging over finite integer domains.

Keywords: Model Evolution, Integer Arithmetic

Joint work of: Baumgartner, Peter; Tinelli, Cesare

A Dynamic Logic for Deductive Verification of Concurrent Programs

Bernhard Beckert (Universität Koblenz-Landau, D)

We present an approach aiming at full functional deductive verification of concurrent Java programs, based on symbolic execution. We define a Dynamic Logic and a deductive verification calculus for a restricted fragment of Java with native concurrency primitives. Even though we cannot yet deal with non-atomic loops, employing the technique of symmetry reduction allows us to verify unbounded systems.

The method relies on decision procedures for proving properties of data structures and the applicability of symmetry reductions.

The calculus has been implemented within the KeY system, and we demonstrate it by verifying a central method of the StringBuffer class from the Java standard library.

Joint work of: Beckert, Bernhard; Klebanov, Vladimir

The Calculus of Computation

Aaron Bradley (University of Colorado, USA)

I will introduce the new textbook, "The Calculus of Computation: Decision Procedures with Applications to Verification," that I wrote with Zohar Manna.

We wrote the textbook for advanced undergraduate and beginning graduate students, as well as computer scientists who desire a detailed introduction to satisfiability decision procedures. My purpose in discussing it at this seminar is to bring it to the attention of colleagues who might use it in their courses.

Part I of the book provides a classical presentation of first-order logic, theories, induction, and program verification. Part II studies algorithmic reasoning, in particular satisfiability decision procedures and static analyses. Associated with the book is a freely available verifying compiler, "PiVC", which supports verification exercises in the book.

Keywords: First-Order Logic, Verification, Decision Procedures, Static Analysis

Joint work of: Bradley, Aaron; Manna, Zohar

Full Paper:
http://www.springer.com/978-3-540-74112-1

See also: Bradley, A. R. and Manna, Z. The Calculus of Computation: Decision Procedures with Applications to Verification. Springer, 2007

Computing predicate abstractions by integrating BDDs and SMT solvers

Alessandro Cimatti (ITC-irst - Trento, I)

The efficient computation of exact abstractions of a concrete program for a given set of predicates is key to the efficiency of Counter-Example Guided Abstraction-Refinement (CEGAR). Recent work propose the use of DPLL-based SMT solvers, modified into enumerators.

This technique has been successfully applied in the realm of software, where a control flow graph is available to direct the exploration. However this approach shows some limitations when the number of models grows: in fact, it intrinsically relies on the enumeration of all the implicants, which basically requires the enumerations of all the disjuncts in the DNF of the abstraction.

In this paper, we propose a new technique to improve the construction of abstractions. We complement SMT solvers with the use of BDDs, which enables us to avoid the model explosion. Essentially, we exploit the fact that BDDs are a DAG representation of the space that a DPLL-based enumerator treats as a tree. A preliminary experimental evaluation shows the potential of the approach.

Instance-based Theorem Proving and Approximating Models

Koen Claessen (Chalmers UT - Göteborg, S)

I will discuss the latest developments in Equinox, an instance-based theorem prover that not only produces proofs, but can also produce approximative counter models in case of a failed proof search.

I will argue for the usefulness of such models and present a number of practical examples.

Keywords: Instance-Based Theorem Proving, Model Finding

From Non-Disjoint Combination to Satisfiability and Model-Checking of Infinite-State Systems

Silvio Ghilardi (Università di Milano, I); Enrica Nicolini (CNRS - Nancy, F)

In the first part of our contribution, we review recent results on combined constraint satisfiability for first order theories in the non-disjoint signatures case: this is done mainly in view of the applications to temporal satisfiability and model-checking covered by the second part of our talk, but we also illustrate in more detail some case-study where non-disjoint combination arises. The first case deals with extensions of the theory of arrays where indexes are endowed with a Presburger arithmetic structure and a length expressing 'dimension' is added; the second case deals with the algebraic counterparts of fusion in modal logics. We then recall the basic features of the Nelson-Oppen method and investigate sufficient conditions for it to be complete and terminating in the non-disjoint signatures case: for completeness we rely on a model-theoretic T_0-compatibility condition (generalizing stable infiniteness) and for termination we impose a noetherianity requirement on positive constraints chains. We finally supply examples of theories matching these combinability hypotheses.

In the second part of our contribution, we develop a framework for integrating first-order logic (FOL) and discrete Linear time Temporal Logic (LTL). Manna and Pnueli have extensively shown how a mixture of FOL and LTL is sufficient to precisely state verification problems for the class of reactive systems: theories in FOL model the (possibly infinite) data structures used by a reactive system while LTL specifies its (dynamic) behavior. Our framework for the integration is the following: we fix a theory T in a first-order signature Σ and consider as a temporal model a sequence $\mathcal{M}_1, \mathcal{M}_2, \ldots$ of standard (first-order) models of T and assume such models to share the same carrier (or, equivalently, the domain of the temporal model to be 'constant'). Following [Plaisted 86], we consider symbols from a subsignature Σ_r of Σ to be *rigid*, i.e. in a temporal model $\mathcal{M}_1, \mathcal{M}_2, \ldots$, the Σ_r-restrictions of the $\mathcal{M}_i$'s must coincide. The symbols in $\Sigma \setminus \Sigma_r$ are called 'flexible' and their interpretation is allowed to change over time (free variables are similarly divided into 'rigid' and 'flexible'). For model-checking, the *initial states* and the *transition relation* are represented by first-order formulae, whose role is that of (non-deterministically) restricting the temporal evolution of the model.

In the quantifier-free case, we obtain sufficient conditions for decidability for both satisfiability and model-checking of safety properties *by lifting combination methods* for *non-disjoint* theories in FOL: noetherianity and T_0-compatibility (where T_0 is the theory axiomatizing the rigid subtheory) gives decidability of satisfiability, whereas T_0-compatibility and local finiteness give safety model-checking decidability. The proofs of these decidability results suggest how decision procedures for the constraint satisfiability problem of theories in FOL and algorithms for checking the satisfiability of propositional LTL formulae can be integrated. This paves the way to employ efficient Satisfiability Modulo Theories solvers in the model-checking of infinite state systems. We illustrate our techniques on some examples and discuss further work in the area.

Joint work of: Ghilardi, Silvio; Ranise, Silvio; Nicolini, Enrica; Zucchelli, Daniele

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2007/1247

Termination of Programs using Term Rewriting and SAT Solving

Jürgen Giesl (RWTH Aachen, D)

There are many powerful techniques for automated termination analysis of term rewrite systems (TRSs). However, up to now they have hardly been used for real programming languages. In this talk, we describe recent results which permit the application of existing techniques from term rewriting in order to prove termination of programs. We discuss two possible approaches:

1. One could translate programs into TRSs and then use existing tools to verify termination of the resulting TRSs.

2. One could adapt TRS-techniques to the respective programming languages in order to analyze programs directly.

We present such approaches for the functional language Haskell and the logic language Prolog. Our results have been implemented in the termination provers AProVE and Polytool. In order to handle termination problems resulting from real programs, these provers had to be coupled with modern SAT solvers, since the automation of the TRS-termination techniques had to improve significantly. Our resulting termination analyzers are currently the most powerful ones for Haskell and Prolog.

The talk is based on joint work with Danny De Schreye, Manh Thang Nguyen, Peter Schneider-Kamp, Alexander Serebrenik, Stephan Swiderski, and René Thiemann.

Keywords: Termination, Term Rewriting, Haskell, Prolog, SAT Solving

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2007/1248

Superposition for Finite Domains

Thomas Hillenbrand (MPI für Informatik - Saarbrücken, D)

Standard superposition is not a decision procedure for first-order finite domain problems.

One reason are inferences with the explicit finite domain clause $x \simeq 1 \vee \ldots \vee x \simeq n$; others are unbounded inferences from transitivity-like clauses, or literals with non-linear variable occurrences. Exploiting a stronger lifting argument, we present a more restrictive superposition calculus that actually constitutes a decision procedure for finite domain problems. In addition, we demonstrate that, in a framework with a sort discipline based on general monadic predicates, the benefits of this calculus can be transferred to finite domain sorts occurring together with potentially infinite sorts.

Joint work of: Hillenbrand, Thomas; Weidenbach, Christoph

Rewriting with Semantic Data Structures: Termination using Dependency Pairs

Deepak Kapur (University of New Mexico, USA)

Standard rewriting systems on free data structures have limited expressive power since semantic data structures like sets or multisets cannot be modeled elegantly. A class of rewrite systems that allows the use of semantic data structures is defined. Additionally, built-in natural numbers, including (dis)equality, ordering, and divisibility constraints, are supported for expressing rewrite rules.

The rewriting mechanism is a combination of normalized equational rewriting with validity checking of instantiated constraints. The framework is highly expressive and allows modeling of algorithms in a very natural way.

One of the most important properties of algorithms and rewrite based specifications of them is termination. The dependency pair framework is extended to be applicable to this class of rewrite systems, thus obtaining a flexible and powerful method for termination proving that can be automated effectively.

Keywords: Rewrite Systems, Semantic Data Structures, Normalized Rewriting, Termination, Constraints, Presburger Arithmetic

Joint work of: Falke, Stephan; Kapur, Deepak

Reasoning Algebraically About P-Solvable Loops

Laura Ildiko Kovacs (University of Linz, A)

By combining techniques from algorithmic combinatorics, symbolic summation, computer algebra and computational logic, a framework is developed for generating polynomial invariants for imperative programs operating on numbers.

A certain family of loops, called P-solvable, is defined for which the value of each program variable can be expressed as a polynomial of the initial values of variables, the loop counter, and some new variables where there are algebraic dependencies among the new variables. For such loops, a systematic method is developed for generating polynomial invariants.

Further, if the bodies of these loops consist only of assignments and conditional branches, and test conditions in the loop and conditionals are ignored, the method is shown to be complete for some special cases. By completeness we mean that it generates a set of polynomial invariants from which, under additional assumptions, any polynomial invariant can be derived.

Exploiting the symbolic manipulation capabilities of the computer algebra system Mathematica, these techniques are implemented in a new software package called Aligator.

Keywords: Loop Invariant, Polynomial Relations, Groebner Basis, Symbolic Summation

Architecting Solvers for SAT Modulo Theories: Nelson-Oppen with DPLL

Sava Krstic (Intel Hillsboro, USA)

We will discuss a transition system representing a high-level but detailed architecture for SMT solvers that combine a propositional SAT engine with solvers for multiple disjoint theories. The system captures succinctly and accurately all the major aspects of the solver's global operation: boolean search with across-the-board backjumping, communication of theory-specific facts and equalities between shared variables, and cooperative conflict analysis. Provably correct and prudently underspecified, our system is a usable ground for high-quality implementations of comprehensive SMT solvers.

Joint work of: Krstic, Sava; Goel, Amit

Full Paper:
http://www.springerlink.com/content/p664r371240m7500/

See also: Boris Konev, Frank Wolter (Eds.): Frontiers of Combining Systems, 6th International Symposium, FroCoS 2007, Liverpool, UK, September 10-12, 2007, Proceedings. Lecture Notes in Computer Science 4720 Springer 2007

Relating Context and Sequence Unification

Temur Kutsia (University of Linz, A)

In this work we study relations between sequence and context unification.

These problems represent two generalizations of word equations. Sequence unification is decidable, while decidability of context unification is still an open problem. We use "currying" transformation to encode sequence unification into a fragment of context unification whose solutions are restricted to have a special shape, called left-hole contexts. Extending this restriction to the entire context unification problem, we obtain a new decidable variant of context unification that we call left-hole context unification. Inverse "currying" transforms left-hole context unification into a new decidable extension of sequence unification. Besides, we obtain a new decidability proof of sequence unification.

Joint work of: Kutsia, Temur; Levy, Jordi; Villaret, Mateu

Full Paper:
http://www.springerlink.com/content/q201336248611175/fulltext.pdf

See also: Temur Kutsia, Jordi Levy, and Mateu Villaret. Sequence Unification Through Currying. In: Franz Baader, editor, Proceedings of the 18th International Conference on Rewriting Techniques and Applications (RTA'07), June 26-28, 2007. Paris, France. Volume 4533 of the Lecture Notes in Computer Science. Springer Verlag, 2007

SMELS: Satisfiability Modulo Equality with Lazy Superposition

Christopher Lynch (Clarkson University - Potsdam, USA)

We give a method for extending efficient SMT theorem provers to handle quantifiers, using Superposition inference rules.

In our method, the input formula is converted into CNF as in traditional first order logic theorem provers. The ground clauses are given to the SMT theorem prover, which runs a DPLL method to build partial models. The partial model is passed to a congruence closure procedure, as is normally done in SMT. The congruence closure calculates all reduced (dis)equalities in the partial model and passes them to a Superposition procedure, along with a justification. The Superposition procedure then performs an inference rule, which we call Justified Superposition, between the (dis)equalities and the nonground clauses, plus usual Superposition rules with the nonground clauses. Any resulting ground clauses are provided to the DPLL engine. We prove the completeness of this method, using a nontrivial modification of the Bachmair Ganzinger model generation technique. We believe this combination uses the best of both worlds, an SMT process to handle ground clauses efficiently, and a Superposition procedure which uses orderings to handle the nonground clauses.

We also present some ideas for making the Superposition part more efficient, where we compile the inferences using ideas from Schematic Paramodulation and techniques from databases and expert systems. An implementation based on these ideas is planned.

Keywords: SMT, Superposition

Joint work of: Lynch, Christopher; Tran, Duc-Khanh

KBO^3

Aart Middeldorp (Universität Innsbruck, A)

We present three different encodings (SAT, PBC, SMT) to prove termination of rewrite systems with the Knuth-Bendix order automatically.

The constraints for the weight function and for the precedence are encoded in a formula which is tested for satisfiability. Any satisfying assignment represents a weight function and a precedence such that the induced Knuth-Bendix order orients the rules of the encoded rewrite system from left to right. Comprehensive experimental results will be presented.

Keywords: Termination, Term Rewriting, SAT, PBC, SMT

Joint work of: Middeldorp, Aart; Zankl, Harald

Hard Satisfiable Clause Sets for Benchmarking SAT Solvers with Equivalence Reasoning Techniques

Ilkka Niemelä (Helsinki University of Technology, FIN)

A family of hard satisfiable instances in conjunctive normal form based on random regular graphs is introduced. Instances in the family have the structure of a system of linear equations. Schemes for introducing nonlinearity to the instances are developed, making the instances suitable for benchmarking solvers with equivalence reasoning techniques. An extensive experimental evaluation shows that the computational hardness of the instances increases faster than in other well-known families of hard satisfiable instances, with already small instances being very challenging.

Keywords: Boolean Satisfiability, Equivalence Reasoning, Hard Instances, Solver Evaluation

Joint work of: Niemelä, Ilkka; Haanpää, Harri; Järvisalo, Matti; Kaski, Petteri

Full Paper:
http://jsat.ewi.tudelft.nl/

See also: Harri Haanpää, Matti Järvisalo, Petteri Kaski, and Ilkka Niemelä. Hard Satisfiable Clause Sets for Benchmarking Equivalence Reasoning Techniques. Journal on Satisfiability, Boolean Modeling and Computation. Volume 2 (2006), 27-46.

Reflecting Quantifier Elimination: From Dense Linear Orders to Presburger Arithmetic

Tobias Nipkow (TU München, D)

This talk presents reflected quantifier elimination procedures for both integer and real linear arithmetic. Reflection means that the algorithms are expressed as recursive functions on recursive data types inside some logic (in our case HOL), are verified in that logic, and can then be applied to the logic itself.

After a brief overview of reflection we will discuss a number of quantifier elimination algorithms for the following theories:

– Dense linear orders without endpoints. We formalize the standard DNF-based algorithm from the literature.
– Linear real arithmetic. We present both a DNF-based algorithm extending the case of dense linear orders and an optimized version of the algorithm by Ferrante and Rackoff.
– Presburger arithmetic. Again we show both a naive DNF-based algorithm and the DNF-avoiding one by Cooper.

We concentrate on the algorithms and their formulation in Isabelle/HOL, using the concept of locales to allow modular definitions and verification.

This work will appear in the proceedings of the Marktoberdorf Summer School 2007.

Joint work of: Nipkow, Tobias; Chaieb, Amine

MiniMaxSat: an Efficient Weighted MaxSAT Solver

Albert Oliveras (TU of Catalonia - Barcelona, E)

Max-SAT is the optimization version of SAT where the goal is to satisfy the maximum number of clauses. It is considered one of the fundamental combinatorial optimization problems and many important problems can be naturally expressed as Max-SAT.

In this talk we introduce MiniMaxSat a new Max-SAT solver that is built on top of MiniSAT+. It incorporates the best current SAT and Max-SAT techniques. It can handle hard clauses (clauses of mandatory satisfaction as in SAT), soft clauses (clauses whose falsification is penalized by a cost) as well as pseudo-boolean objective functions and constraints. Its main features are: learning and backjumping on hard clauses; resolution-based and substraction-based lower bounding; and lazy propagation with the two-watched literals scheme. Our empirical evaluation comparing a wide set of solving alternatives on a broad set of optimization benchmarks indicates that the performance of MiniMaxSat is usually close to the best specialized alternative and, in some cases, even better.

Joint work of: Oliveras, Albert; Larrosa, Javier; Heras, Federico

Presburger arithmetic and reasoning about collections of elements

Ruzica Piskac (EPFL - Lausanne, CH)

We describe results on decision procedures for sets with cardinalities, as well some recent extensions. These extensions also involve investigations of properties of formulas in Presburger arithmetic and integer linear programming.

Keywords: Decision Procedures

Joint work of: Piskac, Ruzica; Kuncak, Viktor

Experiments with CPLEX as a linear SMT solver

Enric Rodríguez-Carbonell (TU of Catalonia - Barcelona, E)

So far, the development and implementation of decision procedures for linear arithmetic in the context of Satisfiability Modulo Theories (SMT) has been mostly independent from the technology used in Operations Research (OR). This situation has been justified given that the requirements for linear SMT solvers are different from those of OR linear programming tools for optimization: they have to handle disequalities and strict inequalities, and in order to guarantee correctness, infinite precision arithmetic instead of floating-point arithmetic must be used. In this talk we will describe ongoing experiments in which we have used the linear programming tool CPLEX as a linear SMT solver.

Keywords: Satisfiability Modulo Theories, Operations Research, Simplex Algorithm

Path Invariants

Andrey Rybalchenko (MPI für Software Systeme - Saarbrücken, D)

The success of software verification depends on the ability to find a suitable abstraction of a program automatically. We propose a method for automated abstraction refinement which overcomes some limitations of current predicate discovery schemes. In current schemes, the cause of a false alarm is identified as an infeasible error path, and the abstraction is refined in order to remove that path. By contrast, we view the cause of a false alarm —the spurious counterexample— as a full-fledged program, namely, a fragment of the original program whose control-flow graph may contain loops and represent unbounded computations. There are two advantages to using such path programs as counterexamples for abstraction refinement. First, we can bring the whole machinery of program analysis to bear on path programs, which are typically small compared to the original program. Specifically, we use constraint-based invariant generation to automatically infer invariants of path programs —so-called path invariants. Second, we use path invariants for abstraction refinement in order to remove not one infeasibility at a time, but at once all (possibly infinitely many) infeasible error computations that are represented by a path program. Unlike previous predicate discovery schemes, our method handles loops without unrolling them; it infers abstractions that involve universal quantification and naturally incorporates disjunctive reasoning.

Keywords: Formal Verification, Software Model Checking, Predicate Abstraction, Abstraction Refinement, Invariant Synthesis

Full Paper:
http://www.mpi-inf.mpg.de/~rybal/papers/pldi07-path-invariants.pdf

Handling Integer Arithmetic and Quantifiers in a Sequent Calculus

Philipp Rümmer (Chalmers UT - Göteborg, S)

We first introduce a calculus for handling ground integer arithmetic in first-order logic. The method is tailored to Java program verification and meant to be used both as a supporting procedure and simplifier during interactive verification and as an automated tool for discharging (ground) proof obligations. It is fully implemented as part of the KeY verification system. There are four main components: a complete procedure for linear equations, a complete procedure for linear inequalities, an incomplete procedure for nonlinear (polynomial) equations, and an incomplete procedure for nonlinear inequalities. The calculus is complete for the generation of counterexamples for invalid ground formulas in integer arithmetic.

As work in progress, we then discuss how the calculus can be extended to deal with quantifiers. The proposed approach is based on the work by Martin Giese on tableaux with incremental closure. In the theory of linear integer arithmetic, this naturally leads to a formulation of the Omega test. Incomplete procedures are obtained for more expressive fragments like nonlinear (polynomial) integer arithmetic or arithmetic together with uninterpreted function symbols.

Full Paper:
http://www.cs.chalmers.se/~philipp/publications/arithmetic-verify07.pdf

See also: Philipp Rümmer, 2007, A Sequent Calculus for Integer Arithmetic with Counterexample Generation, Proceedings of 4th International Verification Workshop (VERIFY'07), CEUR Workshop Proceedings (CEUR-WS.org)

Using Tableau to Decide Expressive Description Logics with Role Negation

Renate Schmidt (Manchester University, GB)

Mainstream description logics systems and ontology web languages provide a rich supply of concept operators, but there is currently little support for complex role operators. Role negation in particular cannot be handled by current tableau decision procedures and is not supported by the standard web ontology languages. We define a sound and complete tableau calculus for the description

logic ALBO and show that it provides a basis for decision procedures for this logic and numerous other description logics with full role negation. ALBO is the extension of the description logic ALC with the Boolean role operators, inverse of roles, domain and range restriction operators and it includes full support for nominals (individuals). ALBO is a very expressive description logic which subsumes Boolean modal logic and the two-variable fragment of first-order logic and reasoning in it is NExpTime-complete. An important novelty is the use of a generic, unrestricted blocking rule as a replacement for standard loop checking mechanisms implemented in description logic systems. An implementation of our approach exists in the MetTeL system.

Keywords: Automated reasoning, Decidability, Tableaux, Description Logic

See also: Schmidt, R. A. and Tishkovsky, D. (2007). Using Tableau to Decide Expressive Description Logics with Role Negation, The Semantic Web: ISWC 2007, Lecture Notes in Computer Science, Springer, To appear.

Implementing RPO and POLO using SAT

Peter Schneider-Kamp (RWTH Aachen, D)

Well-founded orderings are the most basic, but also most important ingredient to virtually all termination analyses. The recursive path order with status (RPO) and polynomial interpretations (POLO) are the two classes that are the most popular in the termination analysis of term rewrite systems. Numerous fully automated search algorithms for these classes have therefore been devised and implemented in termination tools.

Unfortunately, the performance of these algorithms on all but the smallest termination problems has been lacking. E.g., recently developed transformations from programming languages like Haskell or Prolog allow to apply termination tools for term rewrite systems to real programming languages. The results of the transformations are often of non-trivial size, though, and cannot be handled efficiently by the existing algorithms.

The need for more efficient search algorithms has triggered research in reducing these search problems into decision problems for which more efficient algorithms already exist. Here, we introduce an encoding of RPO and POLO to the satisfiability of propositional logic (SAT). We implemented these encodings in our termination tool AProVE.

Extensive experiments have shown that one can obtain speedups in orders of magnitude by this encoding and the application of modern SAT solvers.

The talk is based on joint work with Elena Annov, Mike Codish, Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, René Thiemann, and Harald Zankl.

Keywords: Termination, SAT, Recursive Path Order, Polynomial Interpretation

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2007/1249

Delayed Theory Combination

Roberto Sebastiani (Università di Trento, I)

Many approaches for Satisfiability Modulo Theory (SMT(T)) rely on the integration between a SAT solver and a decision procedure for sets of literals in the background theory T (T-solver). When T is the combination of two or more simpler theories, the approach is typically handled by means of Nelson-Oppen's combination procedure (NO), in which two specific T-solvers deduce and exchange (disjunctions of) interface equalities.

In recent papers we have proposed a new approach to SMT($\bigcup_i T_i$), called *Delayed Theory Combination* (DTC). Here part or all the (possibly very expensive) task of deducing interface equalities is played by the SAT solver itself.

In this talk I present and discuss the main features of DTC, and present a comparison with N.O. combination procedure.

Keywords: Satisfiability Modulo Theories, Combination of Theories, DPLL

See also: M. Bozzano, R. Bruttomesso, A. Cimatti, T. Junttila, P. van Rossum, S. Ranise, and R. Sebastiani. Efficient Satisfiability Modulo Theories via Delayed Theory Combination. In Proc. CAV 2005, volume 3576 of LNCS. Springer, 2005.

M. Bozzano, R. Bruttomesso, A. Cimatti, T. Junttila, P. van Rossum, S. Ranise, and R. Sebastiani. Efficient Theory Combination via Boolean Search. Information and Computation, 204(10), 2006.

R. Bruttomesso, A. Cimatti, A. Franzén, A. Griggio, and R. Sebastiani. Delayed Theory Combination vs. Nelson-Oppen for Satisfiability Modulo Theories: a Comparative Analysis. In Proc. LPAR 06, volume 4246 of LNAI. Springer, 2006.

Local Theory Extensions, Hierarchical Reasoning and Applications to Verification

Viorica Sofronie-Stokkermans (MPI für Informatik - Saarbrücken, D)

Many problems occurring in verification can be reduced to proving the satisfiability of conjunctions of literals in a background theory. This can be a concrete theory (e.g. the theory of real or rational numbers), the extension of a theory with additional functions (free, monotone, or recursively defined) or a combination of theories. It is therefore very important to have efficient procedures for checking the satisfiability of conjunctions of ground literals in such theories.

We present some new results on hierarchical and modular reasoning in complex theories, as well as several examples of application domains in which efficient reasoning is possible. We show, in particular, that various phenomena analyzed in the verification literature can be explained in a unified way using the notion of local theory extension.

Keywords: Automated Reasoning, Combinations of Decision Procedures, Verification

Joint work of: Ihlemann, Carsten; Jacobs, Swen; Sofronie-Stokkermans, Viorica

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2007/1250

Can linear hyperplane arrangements be relevant for deciding (disjunctive) linear arithmetic ?

Ofer Strichman (Technion - Haifa, IL)

We describe work in progress dedicated to a decision procedure for disjunctive linear arithmetic, based on traversing in various ways the linear hyperplane arrangement induced by the input formula's predicates.

We propose two methods:

1. a method by which theory propagation is approximated, with the use of hints (clauses that only affect decisions in the SAT solver, but do not participate in conflicts)

2. 'reversed DPLL(T)', by which the theory leads the search rather than the SAT solver. We show a concrete algorithm for traversing the cells in the linear hyperplane arrangement, which can be the base for such a procedure.

Keywords: Linear Arithmetic

Building Decision Procedures in the Calculus of Inductive Constructions

Pierre-Yves Strub (Ecole Polytechnique - Palaiseau, F)

It is commonly agreed that the success of future proof assistants will rely on their ability to incorporate computations within deduction in order to mimic the mathematician when replacing the proof of a proposition P by the proof of an equivalent proposition P' obtained from P thanks to possibly complex calculations.

We present a new version of the calculus of inductive constructions which incorporates arbitrary decision procedures into deduction via the conversion rule of the calculus. The novelty of the problem in the context of the calculus of inductive constructions lies in the fact that the computation mechanism varies along proof-checking: goals are sent to the decision procedure together with the set of user hypotheses available from the current context. Our main result shows that this extension of the calculus of constructions does not compromise its main properties: confluence, subject reduction, strong normalization and consistency are all preserved.

Decision Procedures for Loop Detection

René Thiemann (RWTH Aachen, D)

The dependency pair technique is a powerful modular method for automated termination proofs of term rewrite systems. We first show that dependency pairs are also suitable for disproving termination: loops can be detected more easily.

In a second step we analyze how to disprove innermost termination. Here, we present a novel procedure to decide whether a given loop is an innermost loop.

We implemented and evaluated our results in the automated termination prover AProVE.

Keywords: Non-Termination, Decision Procedures, Term Rewriting

Joint work of: Thiemann, René; Giesl, Jürgen; Schneider-Kamp, Peter

Combined Satisfiability Modulo Parametric Theories

Cesare Tinelli (University of Iowa, USA)

We give a fresh theoretical foundation for designing comprehensive solvers for satisfiability modulo theories (SMT). In contrast with the traditional Nelson-Oppen approach dealing with theories in single- or many-sorted first-order logic, we consider parametric theories—a restricted class of theories defined in the context of a higher-order logic with polymorphic types. Our main result is the combination theorem for decision procedures for disjoint parametric theories. Combinations of parametric theories conveniently express the “logic” of common datatypes, covering virtually all deeply nested structures (lists of arrays of sets of ...) that arise in verification practice. The vexatious stable infiniteness condition of the traditional approach is replaced in our setting with a milder flexibility condition. Our results do not subsume existing results, nor are subsumed by them, but they apply more widely because the datatypes relevant in applications are described by theories that satisfy our parametricity requirements without necessarily satisfying the stable infiniteness requirements of other combination methods.

Keywords: Nelson-Oppen Combination, Parametricity, SMT

Joint work of: Krstic, Sava; Goel, Amit; Grundy, Jim; Tinelli, Cesare

Integrating Linear Arithmetic into the Superposition Calculus

Andrei Voronkov (Manchester University, GB)

We point out several problems with the standard definition of combination of universal theories when additional first-order axioms are presented.

We give a new definition for the case of integrating first-order logic and rational arithmetic. After that we present a method of integrating linear rational arithmetic into the superposition calculus for first-order logic. We prove several incompleteness results and a completeness result under some finiteness assumptions.

Keywords: Combination of Theories

Joint work of: Korovin, Konstantin; Voronkov, Andrei

New developments in SPASS+T

Uwe Waldmann (MPI für Informatik - Saarbrücken, D)

SPASS+T is an extension of the superposition-based theorem prover SPASS that enlarges the reasoning capabilities of SPASS using both an arbitrary SMT procedure and built-in simplification rules and standard axioms for arithmetic.

We discuss new developments in SPASS+T, including an advanced coupling between SPASS and the SMT procedure, additional inference rules, and a new term ordering.

Keywords: Superposition, Theorem Proving, Satisfiability Modulo Theories

Automatic Analysis of LAN Infrastructures

Christoph Weidenbach (MPI für Informatik - Saarbrücken, D)

Important guarantees for LAN infrastructures include connectivity, error detection/recovery, robust changes and security. In the talk I will develop a LAN infrastructure model in first-order logic enabling automatic analysis of such properties. The model starts at the ethernet layer of the TCP/IP stack and ranges up to application protocols such as DHCP.

Keywords: Automated Protocol Verification Analysis

Joint work of: Hirth, Simon; Karl, Carsten; Weidenbach, Christoph

Multiplicative Arithmetic Revisited

Volker Weispfenning (Universität Passau, D)

Multiplicative arithmetic is the first-order theory T of natural numbers with multiplication. A famous theorem of Skolem (1930) asserts that T is decidable. Later proofs of this fact were given by Feferman-Vaught (1959) and Ferrante-Rackoff (1979).

The latter also established an asymptotic upper complexity bound for the decision problem. All proofs are based on the isomorphism of $(N, .)$ with $(N_0^*, +)$ and the decidability of Presburger arithmetic.

The talk investigates the problem of algorithmic quantifier elimination (QE) for T in a suitable extension language. In his Habilitationsschrift (1979) the author established a QE procedure for T in a large and rather unnatural extension language by very general principles. Here we present an efficient "restricted" QE for T can be performed in a "natural" extension language. The method yields also explicit parametric solutions for existential problems. We also discuss the asymptotic and the practical complexity of the method.

The Reduction Approach to Decision Procedures

Calogero G. Zarba (Saarland University, D)

Reduction procedures reduce the satisfiability of a formula over a complex theory to the satisfiability of a formula over a simpler theory.

In the first part of this talk, we present a reduction procedure that allows us to reduce the theory of sets to the theory of equality. As an added bonus, the procedure allows us to compute interpolants for the theory of sets.

In the second part of the talk, we present a method for combining reduction procedures. As an application, we obtain a decision procedure for a theory combining equality with uninterpreted function symbols, fixed-sized bit-vectors, integer linear arithmetic, lists, arrays, sets, and multisets. The decision procedure has been implemented in the SMT solver Caissa.

Joint work of: Kapur, Deepak; Zarba, Calogero G.