Report from Dagstuhl Seminar 23471

The Next Generation of Deduction Systems: From Composition to Compositionality

Maria Paola Bonacina*[1], Pascal Fontaine*[2], Cláudia Nalon*[3], Claudia Schon*[4], and Martin Desharnais†[5]

1 **University of Verona, IT.** `mariapaola.bonacina@univr.it`
2 **University of Liège, BE.** `pascal.fontaine@uliege.be`
3 **University of Brasília, BR.** `nalon@unb.br`
4 **Hochschule Trier, DE.** `c.schon@hochschule-trier.de`
5 **Max-Planck-Institut für Informatik Saarbrücken, DE.** `desharnais@mpi-inf.mpg.de`

Abstract

Deduction systems are computer procedures that employ inference or transition rules, search strategies, and multiple supporting algorithms, to solve problems by logico-deductive reasoning. They are at the heart of *SAT/SMT solvers*, *theorem provers*, and *proof assistants*. The wide range of successful applications of these tools shows how logico-deductive reasoning is well-suited for machines. Nonetheless, *satisfiability* and *validity* are difficult problems, and applications require reasoners to handle *large and heterogeneous knowledge bases*, and to generate *proofs* and *models* of increasing size and diversity. Thus, a vast array of techniques was developed, leading to what was identified during the seminar as a *crisis of growth*. This crisis manifests itself also as a software crisis, called *automated reasoning software crisis* at the seminar. Many deduction systems remain prototypes, while relatively few established systems resort to assemble techniques into *portfolios* that are useful for experiments, but do not lead to breakthroughs.

In order to address this *crisis of growth*, the Dagstuh Seminar "The Next Generation of Deduction Systems: From Composition to Compositionality" (23471) focused on the key concept of *composition*, that is, a combination where properties of the components are preserved. Composition applies to all building blocks of deduction: *rule systems*, *strategies*, *proofs*, and *models*. All these instances of compositions were discussed during the seminar, including for example composition of *instance-based and superposition-based inference systems*, and composition of modules towards *proof production* in SMT solvers. Other kinds of composition analyzed during the seminar include the composition of *reasoning and learning*, and the composition of *reasoning systems* and *knowledge systems*. Indeed, reasoners *learn* within and across derivations, while for applications, from *verification* to *robotics*, provers and solvers need to work with other knowledge-based components.

In order to address the *automated reasoning software crisis*, the seminar elaborated the concept of *compositionality*, as the engineering counterpart of what is composition at the theory and design levels. The seminar clearly identified *modularity* as the first step towards compositionality, proposing to decompose existing systems into *libraries of modules* that can be recomposed in new systems. The ensuing discussion led to the distinction between automated reasoners that are *industry powertools* and automated reasoners that are *pedagogical tools*. At the societal level, this distinction is important to counter the phenomenon whereby new students are either discouraged by the impossibility of competing with industry powertools, or induced to join only those research groups that work on industry powertools. In summary, the seminar fully succeeded in promoting the exchange of ideas and suggestions for future work.

Seminar November 19–24, 2023 – https://www.dagstuhl.de/23471
2012 ACM Subject Classification Theory of computation → Automated reasoning

* Editor / Organizer
† Editorial Assistant / Collector

The Next Generation of Deduction Systems: From Composition to Compositionality, *Dagstuhl Reports*, Vol. 13, Issue 11, pp. 130–150
Editors: Maria Paola Bonacina, Pascal Fontaine, Cláudia Nalon, and Claudia Schon
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases artificial intelligence, automated reasoning, compositionality, deduction, logic
Digital Object Identifier 10.4230/DagRep.13.11.130

1 Executive Summary

Maria Paola Bonacina
Pascal Fontaine
Cláudia Nalon
Claudia Schon

This report contains the program and outcomes of the Dagstuhl Seminar 23471 on *The Next Generation of Deduction Systems: From Composition to Compositionality* that was held at Schloss Dagstuhl, Leibniz Center for Informatics, during November 19–24, 2023. It was the fifteenth in a series of Dagstuhl Deduction seminars held biennially since 1993.

The motivation for this seminar was threefold:

1. Automated reasoning tools, including *SAT solvers*, *SMT solvers*, *theorem provers*, and *proof assistants*, are widely applied in fields as diverse as *analysis/verification/synthesis of systems*, *programming language design*, *knowledge engineering*, *computer mathematics*, *natural language processing*, and *robotics*. However, *satisfiability* and *validity* remain *fundamentally difficult computational problems*, so that reasoners may run out of time or memory returning "don't know" or may demand too much human labor.
2. After the low-hanging fruits have been picked, the *formalization* of problems require logics, formulas, theories, and knowledge bases that are increasingly complex, large, and heterogeneous. The *size* and *diversity* of the *proofs* and *models*, that reasoners produce to support their answers, increase accordingly.
3. Deduction offers a vast array of techniques, but many implementations of new techniques remain short-lived prototypes, and the transfer of the successful ones into more stable systems is uncertain. Relatively few systems gather most of the resources, but over time they may become too big, monolithic, and unwieldy for further development, or resort to assemble techniques into portfolios. A portfolio allows one to experiment and may win competitions, but it hardly leads to a conceptual synthesis and hence a breakthrough.

The Dagstuhl Seminar on *The Next Generation of Deduction Systems: From Composition to Compositionality* addressed these issues by challenging participants to reflect around the ideas of *composition* and *compositionality*.

A *composition* is a combination such that properties of the components (e.g., *soundness*, *completeness*, *termination*, *model-construction*) are *preserved*. Since different inference systems have different strengths, their composition is essential to meet Challenge (1). For example, the seminar participants presented and discussed research about the composition of equality reasoning by *superposition* with *instance-based* (e.g., *Inst-gen* – "Instance Generation") or *model-based* (e.g., *SCL* – "Simple Clause Learning") inference systems.

A major cause of "don't know" answers in *satisfiability modulo theories* (SMT) is the fact that most decision procedures inside SMT solvers are for quantifier-free fragments of theories, whereas applications require handling quantified formulas. Thus, the seminar addressed the fundamental problem of *composing quantifier reasoning and theory reasoning*. For example, the *QSMA* algorithm, where *QSMA* stands for "Quantified Satisfiability Modulo Assignment," offers a novel solution for quantifier reasoning in a complete theory (e.g., arithmetic).

Historically, *proof generation* was deemed unproblematic in automated theorem provers, whereas *model generation* was deemed unproblematic in SMT solving. This is why recent research has focused on proofs in SMT and models in first-order theorem proving. The seminar reflected these trends. Several talks presented advanced research on *proof production* in SMT, involving *composition of proofs*, both within the SMT solver, as in composition of proofs from different theories, and at the interface of the SMT solver (e.g., CVC5) with a proof assistant (e.g., Lean, Isabelle/HOL). At the next abstraction level, the seminar analyzed these issues in *logical frameworks* (e.g., *Hybrid*, *Dedukti*), where proofs from different proof assistants may be verified, exchanged, translated, and hence re-verified. Work on the representation and composition of first-order models in libraries of problems for first-order theorem provers (e.g., TPTP) is also gaining momentum, and the seminar offered an excellent discussion forum, since several developers of theorem provers were attending.

The drive to improve the search capabilities of deduction procedures in order to meet Challenge (1) leads also to the composition of *reasoning* and *learning*, while Challenge (2) leads to the composition of *reasoning systems* and *knowledge systems*. Learning is a native capability of automated reasoners, as in *lemma learning*. SAT/SMT solvers and theorem provers *learn within a derivation* by learning lemmas to reduce the search space by avoiding repeated work. Reasoners also *learn across derivations* by applying *machine learning* to learn from a very high number of derivations which *strategies* or *tactics* to select for an input problem with certain *features*. The composition of reasoning and learning was discussed at the seminar in *SAT solving*, and in *resolution-based* first-order theorem proving, where the prover is interfaced with an ontology-based knowledge system (e.g., Adimen SUMO).

The sentiment that emerged at the seminar is that approaches based on *composition* will contribute to meet Challenge (3), by endowing deduction systems with *compositionality*, towards going beyond portfolios. The participants discussed the *crisis of growth* that the field is facing, given the rise of so many rule systems, strategies, and techniques. Since it is a crisis of *growth*, the field will emerge from it even stronger. For this to happen, however, it is key to address the issues that make it difficult to transfer new ideas into stable and useable deduction systems. The existing dichotomy, between short-lived prototypes and powerful, but big, monolithic, unwieldy systems, was discussed as an *automated reasoning software crisis*. The need for *modularity* was recognized, and a distinction between *industry powertools* and *pedagogical platforms* was outlined. The latter will have to give up on a unique programming language and programming style, as well as on award-winning efficiency, but will facilitate the entrance of new students, currently discouraged by the impossibility of competing with established tools. Thanks to such platforms, the building of new systems will be less expensive in terms of human time and labor. The risk of new ideas being forgotten without having been properly implemented and tested will be reduced.

The atmosphere throughout the seminar was excellent. For example, a participant told one of the organizers that this seminar motivated them and rekindled their enthusiasm for automated deduction research. An outing – an excursion to Bernkastel-Kues followed by a social dinner in a nearby village – also contributed to establishing a relaxed, friendly atmosphere, conducive to new or strengthtened collaborations.

The bottom-up style of the Dagstuhl experience was preserved, thanks to a flexible program that allowed the participants to volunteer topics and talks throughout the gathering. This seminar maintained a feature that was introduced in the 2021 edition, namely the possibility of giving a tutorial using two time slots rather than one. Altogether, *five tutorials* were given on topics ranging from *proofs in SMT*, *reasoning with quantifiers in SMT*, *composition of reasoning and neuro-symbolic methods*, and *model-based reasoning*.

The following section contains the abstracts for most of the talks and tutorials listed in alphabetical order.

2 Table of Contents

Executive Summary
Maria Paola Bonacina, Pascal Fontaine, Cláudia Nalon, and Claudia Schon 131

Overview of Talks

Combining Proofs for Description Logic and Concrete Domain Reasoning
Franz Baader 136

SMT Proof Production and Integration with the Lean Theorem Prover
Haniel Barbosa 136

The QSMA algorithm
Maria Paola Bonacina 137

An Isabelle/HOL Formalization of the SCL(FOL) Calculus
Martin Desharnais 138

Compositionality from Temporal Logics to Verification for Autonomous Robot Systems
Clare Dixon 138

(Re)Verification of Proofs with Coq or Dedukti
Catherine Dubois 139

Formal Verification at CLEARSY : Needs and Prospects
David Déharbe 139

Reasoning with Structured Contexts of Assumptions
Amy Felty 140

On the need for a modular approach for automated reasoners
Pascal Fontaine 141

Interpolation Properties for Array Theories: Positive and Negative Results
Silvio Ghilardi 141

Formal Verification at Certora
Antti Hyvärinen 142

Improving SMT Solving via Incorporating More Techniques
Fuqi Jia 142

Higher-order constraint term rewriting
Cynthia Kop 142

Reconstruction of cvc5 Proofs in Isabelle/HOL
Hanna Lachnitt 143

Solving Reasoning Problems with Neuro-Symbolic Methods
Feifei Ma and Fuqi Jia 143

A Compositional Proof System for Cylindrical Algebraic Decomposition
Jasper Nalbach 144

A Unified Proof System for Discrete Combinatorial Problems
Jakob Nordström 144

Aspects of Knowledge for Next Generation Systems
Florian Rabe 144

Proofs in cvc5: New Directions with AletheLF
Andrew Joseph Reynolds 145

Using Word Similarities to Guide Resolution
Claudia Schon 145

Proofs for Quantified Boolean Formulas
Martina Seidl 146

More than unit equality
Nick Smallbone 146

On hierarchical reasoning and symbol elimination and applications to parametric verification
Viorica Sofronie-Stokkermans 147

On Finding Short Proofs
Alexander Steen 147

TPTP World Standards and Tools for Tarskian and Kripke Interpretations
Geoff Sutcliffe, Pascal Fontaine, Jack McKeown, and Alexander Steen 148

Mechanizing the Splitting Framework
Sophie Tourret 148

On the (In-)Completeness of Destructive Equality Resolution in the Superposition Calculus
Uwe Waldmann 149

The SCL Calculus and its Implementation
Christoph Weidenbach 149

Participants 150

3 Overview of Talks

3.1 Combining Proofs for Description Logic and Concrete Domain Reasoning

Franz Baader (TU Dresden, DE)

Joint work of Christian Alrabbaa, Franz Baader, Stefan Borgwardt, Patrick Koopmann, Alisa Kovtunova
Main reference Christian Alrabbaa, Franz Baader, Stefan Borgwardt, Patrick Koopmann, Alisa Kovtunova: "Combining Proofs for Description Logic and Concrete Domain Reasoning", in Proc. of the Rules and Reasoning – 7th International Joint Conference, RuleML+RR 2023, Oslo, Norway, September 18-20, 2023, Proceedings, Lecture Notes in Computer Science, Vol. 14244, pp. 54–69, Springer, 2023.
URL https://doi.org//10.1007/978-3-031-45072-3_4

Logic-based approaches to AI have the advantage that their behavior can in principle be explained with the help of proofs of the computed consequences in an appropriate calculus. To benefit from this in practice, considerable work beyond the implementation of a reasoning system is needed to be able to compute proofs that are appropriate for explanation purposes. For ontologies based on Description Logic (DL), we have put this advantage into practice by showing how proofs for consequences derived by DL reasoners can be computed and displayed in a user-friendly way. However, these methods are insufficient in applications where also numerical reasoning is relevant. The present paper considers proofs for DLs extended with concrete domains (CDs) based on the rational numbers, which leave reasoning tractable if integrated into the lightweight DL $\mathcal{EL}_\bot$. Since no implemented DL reasoner supports these CDs, we first develop reasoning procedures for them, and show how they can be combined with reasoning approaches for pure DLs, both for $\mathcal{EL}_\bot$ and the more expressive DL $\mathcal{ALC}$. These procedures are designed such that it is easy to extract proofs from them. We show how the extracted CD proofs can be combined with proofs on the DL side into integrated proofs that explain both the DL and the CD reasoning. We have implemented our reasoning and proof extraction approaches for DLs with concrete domains and have evaluated them on several self-created benchmarks.

3.2 SMT Proof Production and Integration with the Lean Theorem Prover

Haniel Barbosa (Federal University of Minas Gerais-Belo Horizonte, BR)

Joint work of Haniel Barbosa, Tomaz Gomes Mascarenhas, Bruno Andreotti, Andrew Reynolds, Gereon Kremer, Hanna Lachnitt, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Arjun Viswanathan, Scott Viteri, Yoni Zohar, Cesare Tinelli, Clark Barrett
Main reference Haniel Barbosa, Andrew Reynolds, Gereon Kremer, Hanna Lachnitt, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Arjun Viswanathan, Scott Viteri, Yoni Zohar, Cesare Tinelli, Clark W. Barrett: "Flexible Proof Production in an Industrial-Strength SMT Solver", in Proc. of the Automated Reasoning – 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings, Lecture Notes in Computer Science, Vol. 13385, pp. 15–35, Springer, 2022.
URL https://doi.org//10.1007/978-3-031-10769-6_3

SMT solvers can be hard to trust, since it generally means assuming their large and complex codebases do not contain bugs leading to wrong results. Machine-checkable certificates, via proofs of the logical reasoning the solver has performed, address this issue by decoupling confidence in the results from the solver's implementation. In this talk we will describe extensive proof infrastructure of the state-of-the-art SMT solver cvc5, which has enabled the production of proofs in a number of complex domains. We will also show ongoing work towards integrating these proofs into the proof assistant Lean, thus enabling its composition with SMT solvers in a trusted way.

3.3 The QSMA algorithm

Maria Paola Bonacina (University of Verona, IT)

Joint work of Maria Paola Bonacina, Stéphane Graham-Lengrand, Christophe Vauthier
Main reference Maria Paola Bonacina, Stéphane Graham-Lengrand, Christophe Vauthier: "QSMA: A New Algorithm for Quantified Satisfiability Modulo Theory and Assignment", in Proc. of the Automated Deduction – CADE 29 – 29th International Conference on Automated Deduction, Rome, Italy, July 1-4, 2023, Proceedings, Lecture Notes in Computer Science, Vol. 14132, pp. 78–95, Springer, 2023.
URL https://doi.org//10.1007/978-3-031-38499-8_5

Automated theorem provers (ATP) for first-order or higher-order logic and solvers for satisfiability modulo theories (SMT) exhibit impressive power and sophistication. ATP systems reason well about formulas with free symbols and universally quantified variables, removing existential quantifiers by Skolemization. SMT solvers reason well about formulas with free or existentially quantified variables and symbols defined by a theory. However, formulas from key applications involve both arbitrary quantification and defined symbols. The successful composition of quantifier and theory reasoning is a major objective for the next generation of deduction systems. QSMA is a new algorithm for quantifiers in SMT. QSMA stands for Quantified Satisfiability Modulo theory and Assignment. Currently, QSMA works for one theory with unique interpretation of symbols (e.g., arithmetic), so that models differ only in the assignment to free variables. QSMA accepts arbitrary formulas: the quantifiers may alternate and occur in arbitrary positions, as not even prenex normal form is required. After turning universal quantifiers into existential ones by double negation, QSMA performs a recursive descent over the tree structure of the formula, peeling off quantifiers and instantiating variables. Thus, each call works modulo assignment. By building under- and over- approximations of the formula, QSMA zooms in on a model or finds that none exists. The YicesQS solver implements QSMA on top of the Yices 2 solver and exhibits excellent performance in arithmetic. Composing QSMA within the CDSAT framework for conflict-driven satisfiability in a union of theories is the next challenge.

(QSMA is joint work with Stéphane Graham-Lengrand and Christophe Vauthier. CDSAT is joint work with Stéphane Graham-Lengrand and Natarajan Shankar. Stéphane Graham-Lengrand is the author of YicesQS. Bruno Dutertre and Dejan Jovanović are the authors of Yices 2.)

3.4 An Isabelle/HOL Formalization of the SCL(FOL) Calculus

Martin Desharnais (Max-Planck-Institut für Informatik Saarbrücken, DE)

Joint work of Martin Bromberger, Martin Desharnais, Christoph Weidenbach
Main reference Martin Bromberger, Martin Desharnais, Christoph Weidenbach: "An Isabelle/HOL Formalization of the SCL(FOL) Calculus", in Proc. of the Automated Deduction – CADE 29 – 29th International Conference on Automated Deduction, Rome, Italy, July 1-4, 2023, Proceedings, Lecture Notes in Computer Science, Vol. 14132, pp. 116–133, Springer, 2023.
URL https://doi.org//10.1007/978-3-031-38499-8_7

We present an Isabelle/HOL formalization of SCL(FOL): Simple Clause Learning for first-order logic without equality. The main results are formal proofs of soundness, non-redundancy of learned clauses, termination, and refutational completeness. Compared to the unformalized version, the formalized calculus is simpler, a number of results could be generalized, and the non-redundancy property strengthened. We found one bug in a previously published version of the SCL Backtrack rule. Compared to related formalizatons, we introduce a new technique for showing termination based on non-redundant clause learning.

3.5 Compositionality from Temporal Logics to Verification for Autonomous Robot Systems

Clare Dixon (University of Manchester, GB)

This talk was split into two parts: firstly relating to a resolution based calculus and its implementation for propositional linear-time temporal logic and secondly relating to experiences with verification for autonomous robots. With respect to the former I discussed a resolution calculus for proposition linear-time temporal logic and showed how some of the resolution rules could be implemented by calls to a first order logic prover (composition). Secondly I discussed more recent work towards verification for robots (compositionality). Two approaches to verification were mentioned: heterogeneous verification and corroborative verification. With heterogeneous verification we need the robot system being considered to be split into modular subcomponents. On each subcomponent we apply the most suitable verification (including both formal or non-formal verification) for that subsystem for example model checking, theorem proving, software testing, simulation based testing, real robot experiments etc. For each component the assumptions on inputs made on the system eventually must be shown to guarantee required outputs. Ongoing work from colleagues involves how to compose such results to get an overall confidence in the system. Corroborative verification involves applying different verification types to a (sub) system and utilising the outputs to improve the verification models and properties for the other verification types increasing the confidence in the systems.

3.6 (Re)Verification of Proofs with Coq or Dedukti

Catherine Dubois (ENSIIE – Evry, FR)

Joint work of Catherine Dubois, Chantal Keller
Main reference Valentin Blot, Denis Cousineau, Enzo Crance, Louise Dubois de Prisque, Chantal Keller, Assia Mahboubi, Pierre Vial: "Compositional Pre-processing for Automated Reasoning in Dependent Type Theory", in Proc. of the 12th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2023, Boston, MA, USA, January 16-17, 2023, pp. 63–77, ACM, 2023.
URL https://doi.org//10.1145/3573105.3575676
Main reference Burak Ekici, Alain Mebsout, Cesare Tinelli, Chantal Keller, Guy Katz, Andrew Reynolds, Clark W. Barrett: "SMTCoq: A Plug-In for Integrating SMT Solvers into Coq", in Proc. of the Computer Aided Verification – 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II, Lecture Notes in Computer Science, Vol. 10427, pp. 126–133, Springer, 2017.
URL https://doi.org//10.1007/978-3-319-63390-9_7
Main reference Frédéric Blanqui, Gilles Dowek, Émilie Grienenberger, Gabriel Hondet, François Thiré: "A modular construction of type theories", Log. Methods Comput. Sci., Vol. 19(1), 2023.
URL https://doi.org//10.46298/LMCS-19(1:12)2023
Main reference Guillaume Burel, Guillaume Bury, Raphaël Cauderlier, David Delahaye, Pierre Halmagrand, Olivier Hermant: "First-Order Automated Reasoning with Theories: When Deduction Modulo Theory Meets Practice", J. Autom. Reason., Vol. 64(6), pp. 1001–1050, 2020.
URL https://doi.org//10.1007/S10817-019-09533-Z
Main reference Mohamed Yacine El Haddad, Guillaume Burel, Frédéric Blanqui: "EKSTRAKTO A tool to reconstruct Dedukti proofs from TSTP files (extended abstract)", in Proc. of the Proceedings Sixth Workshop on Proof eXchange for Theorem Proving, PxTP 2019, Natal, Brazil, August 26, 2019, EPTCS, Vol. 301, pp. 27–35, 2019.
URL https://doi.org//10.4204/EPTCS.301.5

Verifying or cross-verifying proofs improves the confidence we have in proofs. The talk focusses on the use of Coq or Dedukti as proof checkers. We first give a quick overview of SMTCoq and the recent tactic sniper that allows for more automation when a Coq first order goal is discharged using a SAT/SMT solver. Then we briefly introduce the Dedukti logical framework. The last part of the talk quickly presents the proof tools Zenon Modulo, iProverModulo, Archsat, and Ekstrakto. The three first ones directly produce Dedukti proofs that can be checked by the Dedukti checker. The latter reconstructs a Dedukti proof from a proof trace by reproving each step using a Dedukti producing tool and combining the proofs of the steps to get a proof of the original formula. Finally we point out 2 projects: BWare and ICSPA. The first one aimed at developing a mechanized framework for automated verification of AtelierB proof obligations where Zenon Modulo and iProvermodulo were used. ICSPA is a project in progress where the objectives are to improve confidence in the proofs realized in the context of B/Event-B and TLA+ by formally and independently verifying these proofs and also enable sharing and reusing proofs and models between B/Event-B and TLA+ using lambda-PI calculus modulo theory and Dedukti.

3.7 Formal Verification at CLEARSY : Needs and Prospects

David Déharbe (CLEARSY – Aix-en-Provence, FR)

The talk first briefly presents CLEARSY, a French SME created in the early 2000s that literally has formal methods in its DNA, as it was created to promote the B method and to distribute and maintain Atelier B, the tooling of the B method. The B method, is a rigorous, logic-based framework to design correct-by-construction software components. Invented by

J.R. Abrial, its first industrial application has been software that safeguards a fully automatic metro line in Paris, France. The talk gives some technical details on the B method, so that it should be evident that a robust, reliable, efficient and versatile automatic proof support is essential to make this method even more attractive, by reducing the burden to interact with a proof assistant to discharge proof obligations (POs), and therefore to make it more competitive. Formal verification support in Atelier B has historically relied on custom automatic provers (pp and pr) and proof assistant (pri). The talk presents how third-party provers may now be used in Atelier B, through extension points called proof mechanisms. A proof mechanism is here a combination of tool chains made of an external prover, a translator that encodes the logic of B to that of the prover, and an interpreter for the prover's output. Several such tool chains may be applied to the same PO to increase proof coverage, or trust in the result, or both. Finally, the talk presents novel ideas to improve such proof mechanisms by taking advantage of the capability of ATP systems and SMT solvers to produce proofs and so-called unsat cores. Indeed an unsat core identifies a subset of the (usually very large number of) hypotheses that is sufficient to prove the goal is valid. We can then use an unsat core to build a reduced PO that may then be more easily processed by another tool chain. We thus expect to achieve a much higher coverage for each tool chain and eventually improved confidence and efficiency in using third-party provers in Atelier B.

3.8 Reasoning with Structured Contexts of Assumptions

Amy Felty (University of Ottawa, CA)

Joint work of Amy Felty, Mohamed Yousri Mahmoud, Alberto Momigliano, Brigitte Pientka

We present past and current work on adding support for reasoning on open terms with structured contexts of assumptions in the Hybrid logical framework (LF). Hybrid is implemented in Coq and is designed to support the use of higher-order abstract syntax (HOAS), also called lambda-tree syntax, for representing and reasoning about formal systems such as logics and programming languages. In previous work, we considered a large class of intuitionistic LFs supporting HOAS, and introduced a common infrastructure and general language for structuring such reasoning on open terms with structued contexts, along with some benchmarks. Our recent work has also included large case studies in a linear logic version of Hybrid.

In this talk, we discuss combining and extending our past work in these directions. In particular, we present a variety of examples specific to Hybrid and our case studies, both intuitionistic and linear, and discuss our planned work on extending the general infrastructure and language designed for intuitionistic LFs to the setting of linear LFs. We also discuss automating the generation of lemmas and proofs in both the intuitionistic and linear settings.

References

1. Amy Felty, Alberto Momigliano, and Brigitte Pientka. Benchmarks for reasoning with syntax trees containing binders and contexts of assumptions. *Mathematical Structures in Computer Science*, 28:1507–1540, 2018.
2. Amy P. Felty and Alberto Momigliano. Hybrid: A definitional two-level approach to reasoning with higher-order abstract syntax. *Journal of Automated Reasoning*, 48(1):43–105, 2012.

3 Mohamed Yousri Mahmoud and Amy P. Felty. Formalization of metatheory of the quipper quantum programming language in a linear logic. *Journal of Automated Reasoning*, 63:967–1002, 2019.

3.9 On the need for a modular approach for automated reasoners

Pascal Fontaine (University of Liège, BE)

Joint work of Pascal Fontaine, Haniel Barbosa, Martin Bromberger, Sophie Tourret

In this short presentation, essentially meant to stimulate the discussion among participants, I exposed a very subjective view on the evolution of automated reasoning software, from many small one-person projects in the 90s to a few huge tools now. This poses a problem for the future of the field. I advocate a modular approach to software in our field, to enable reuse, for better distribution of the work, for students to more easily understand the tools by parts, and for better evaluation of parts of automated reasoning software. I briefly reported on my first experiment for a modular approach in SMT with modulariT.

3.10 Interpolation Properties for Array Theories: Positive and Negative Results

Silvio Ghilardi (University of Milan, IT)

Main reference Silvio Ghilardi, Alessandro Gianola, Deepak Kapur, Chiara Naso: "Interpolation Results for Arrays with Length and MaxDiff", ACM Trans. Comput. Log., Vol. 24(4), pp. 28:1–28:33, 2023.
URL https://doi.org//10.1145/3587161

In this talk, we first review basic correspondences between syntactic interpolation properties of a first order theory (quantifier-free interpolation property, general quantifier-free interpolation property, uniform quantifier-free interpolation property) and semantic features related to the class of its models (amalgamation, strong amalgamation, model completability). Then we shall analyze these notions for variants of McCarthy extensional theory of arrays. Whereas the basic theory does not have quantifier-free interpolation property, such property can be restored by adding it an extra symbol 'diff' skolemizing the extensionality axiom. General quantifier-free interpolation property also holds for this theory but not uniform quantifier-free interpolation property, as shown by an explicit counterexample. Since the semantic content of diff operation is rather underspecified, we strenghten the theory by asking diff(a,b) to return the maximum index where two arrays a,b differ (diff returns 0 if they are equal). We also add to a unary 'length' operation. We so end up in a theory still having quantifier-free interpolation, as witnessed by a hyerarchic polynomial reduction to general interpolation for linear arithmetics over indexes. General quantifier free interpolation property may fail, but can be re-gained by introducing constant arrays.

The second part of this talk comes from joint work with A. Gianola, D. Kapur, C. Naso [ACM-TOCL, October 2023]. The first part of the talk reviews old joint work with R. Bruttomesso and S. Ranise, adding to such old work some recent achievements.

3.11 Formal Verification at Certora

Antti Hyvärinen (Certora – Pregassona, CH)

Traditional finance is largely based on the assumption that human actors behave in a trustworthy manner. When this trust was misplaced, this has resulted in big losses for financial systems. Decentralized finance (DeFi) provides a solution by making financial protocols transparent and automated. As a result DeFi does not have the guardrails provided by humans, and catastrophic failures result from incorrect implementations. Certora's bounded model checking based tool helps finding faults in the protocols in an exhaustive way. In this talk I describe how a critical bug was found and fixed in a protocol design and how the tool helped in this process.

3.12 Improving SMT Solving via Incorporating More Techniques

Fuqi Jia (Chinese Academy of Sciences, CN)

Joint work of Fuqi Jia, Feifei Ma, Minghao Liu

In this talk, we would like to introduce some new approaches to solving the SMT problem, including: 1. A bit-blasting based algorithm for SMT(NIA) formulas; 2. A gradient-based algorithm for SMT(NRA) formulas; 3. SMT solving under probability distribution. These works explored the advancement of four components of SMT solving: Search Space Allocation, Variable Order Selection, Model or Partial Model Generation, and Value Decision.

3.13 Higher-order constraint term rewriting

Cynthia Kop (Radboud University Nijmegen, NL)

Joint work of Cynthia Kop, Liye Guo

Logically Constrained Term Rewriting Systems offer a way to couple traditional reasoning on term rewriting systems with SMT reasoning (and tools). This allows them, in turn, to be used for program analysis in a more natural way than pure rewriting (and in different ways than pure SMT). But to model functional languages naturally, we should ideally combine *higher-order* term rewriting systems with SMT. In this presentation, I will discuss the choices to be made for that goal.

3.14 Reconstruction of cvc5 Proofs in Isabelle/HOL

Hanna Lachnitt (Stanford University, US)

Joint work of Hanna Lachnitt, Mathias Fleury, Andrew Reynolds, Haniel Barbosa, Andres Noetzli, Leni Aniva, Clark Barrett, Cesare Tinelli

The proof assistant Isabelle/HOL can call external solvers to automate proof search, which is crucial for using it more effectively. In particular, statements containing bit-vectors are notoriously tedious to prove manually. cvc5 is an efficient satisfiability modulo theories (SMT) solver that is currently only indirectly used by Isabelle. The process of finding a proof inside of Isabelle with the information provided by cvc5 is slow and often fails. In this work we extend the integration between Isabelle and cvc5 so that a proof certificate from cvc5 is shared with Isabelle that can be reconstructed internally into native Isabelle/HOL proofs. We present our ongoing effort to reconstruct these proofs, including problems containing bit-vectors whose reconstruction in Isabelle is currently not supported by any other SMT solver. Modern SMT solvers implement hundreds of term rewriting rules. cvc5 is able to output fine-grained proofs using a separate database of rewrite rules written in the RARE language. We also present IsaRARE, a plugin for Isabelle, that translates such rules to lemmas in Isabelle that can then be used in the reconstruction process out of the box. Additionally, IsaRARE can be used as a verifier for rewrite rules. We evaluate our approach by verifying an extensive set of rewrite rules used by the cvc5 SMT solver.

3.15 Solving Reasoning Problems with Neuro-Symbolic Methods

Feifei Ma (Chinese Academy of Sciences – Beijing, CN) and Fuqi Jia (Chinese Academy of Sciences, CN)

Joint work of Feifei Ma, Fuqi Jia, Minghao Liu

Symbolism and connectionism are two fundamental paradigms for artificial intelligence. In the past decade, connectionism has revived in the name of deep learning, achieving great success in many areas. Recently, neuro-symbolic methods, aiming to bridge the gap between connectionism and symbolism, receive much attention. In this talk, we will introduce some of our initial efforts in this area, which can be classified into two categories: 1. The end-to-end approach where a neural network takes as input the reasoning task and directly outputs the result; 2. The composition of neural network and symbolic method, where a neural network provides assistance to the reasoning algorithm. The targeted reasoning problems include pseudo-Boolean constraint solving, MaxSAT and cylindrical algebraic decomposition.

3.16 A Compositional Proof System for Cylindrical Algebraic Decomposition

Jasper Nalbach (RWTH Aachen, DE)

Joint work of asper Nalbach, Erika Ábrahám, Philippe Specht, Christopher W. Brown, James H. Davenport, Matthew England
Main reference Jasper Nalbach, Erika Ábrahám, Philippe Specht, Christopher W. Brown, James H. Davenport, Matthew England: "Levelwise construction of a single cylindrical algebraic cell", CoRR, Vol. abs/2212.09309, 2022.
URL https://doi.org//10.48550/ARXIV.2212.09309

Cylindrical algebraic decomposition (CAD) is the only complete method implemented in Satisfiability-modulo-theories solvers for solving non-linear arithmetic. Due to its doubly exponential complexity, modern algorithms compute only parts of its projection operation, making solving some practical instances of NRA tractable. There is a variety of cases where savings in the projection are possible, and often there are multiple alternatives for the projection. To manage the maintainability of an algorithm when incorporating special cases, we developed a proof system for modern CAD-based SMT algorithms. This proof system is extensible, separates heuristic decisions (which projection to take) from the correctness of the projection and can be employed in different algorithms. Further, the proof system could be a step towards formal proofs for real algebra.

3.17 A Unified Proof System for Discrete Combinatorial Problems

Jakob Nordström (University of Copenhagen, DK & Lund University, SE)

URL https://gitlab.com/MIAOresearch/software/VeriPB

We give a brief overview of VeriPB, a proof system based on pseudo-Boolean reasoning with 0-1 integer linear inequalities that seems well suited to provide a unified proof logging method for discrete combinatorial problems. We have implemented VeriPB proof logging, together with efficient proof checking, for state-of-the-art solvers in Boolean satisfiable (SAT) solving, SAT-based optimization, graph solving, constraint programming, and a growing list of other combinatorial solving paradigms. We believe that ideas from VeriPB could be useful also in the context of mixed integer linear programming and satisfiability modulo theories (SMT) solving.

This is based on joint work with Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, Magnus O. Myreen, Andy Oertel, and Yong Kiam Tan.

3.18 Aspects of Knowledge for Next Generation Systems

Florian Rabe (Universität Erlangen-Nürnberg, DE)

The Tetrapod model organizes mathematical knowledge into 4+1 aspects, visualized as the corners and the center of a tetrahedral shape. The corners represent fundamentally different ways of assigning semantics, each with an ecosystem of highly specialized tools and large libraries:

- Deduction: proofs, especially if formalized and mechanically verified in proof assistants
- Computation: algorithms, especially if executably implemented in programming languages and computer algebra systems
- Tabulation: systematic lists of examples, especially if encoded as concrete objects stored in databases
- Documentation: human-readable narrative explanations, especially if systematically structured and annotated to enable machine processing

A key novelty of the model is to identify as the central aspect the intersection of the above, called Ontology: names, types, definitions, notations, and properties of mathematical objects, i.e., the information that is critical for knowledge exchange between the dedicated software systems for the other aspects.

This talk gives a high-level overview of the model in discussion-starter style and can be seen as a position statement that next generation systems must invent fundamentally new designs to fully utilize the combination of all aspects.

3.19 Proofs in cvc5: New Directions with AletheLF

Andrew Joseph Reynolds (University of Iowa – Iowa City, US)

Satisfiability Modulo Theories (SMT) solvers are a critical component of many formal methods applications, including for software verification and security analysis. Their soundness is of the utmost importance. While SMT solvers are highly complex systems, some modern SMT solvers now are capable of generating externally checkable proofs. This talk gives the current state of proofs in the SMT solver cvc5. We introduce AletheLF, the new standard format for proofs generated by cvc5. AletheLF is a logical framework based on the SMT-LIB version 3.0 language. It combines the benefits of several previous proof efforts, including a clean syntax, extensibility and integration with other proof formats like DRAT via the use of oracles. We present an initial evaluation of AletheLF, showing the viability of performant proof generation and checking for SMT.

3.20 Using Word Similarities to Guide Resolution

Claudia Schon (Hochschule Trier, DE)

Unlike automated reasoning, human reasoning does not adhere to logical rules exclusively. This is also reflected in the observation of Kahneman that the human mind seems to be based on two integrated systems: a System 1 that works quickly and unconsciously, and a System 2 that works slowly and calculates logically. System 1 embodies intuitions and fast reactions to sensory signals, while System 2 represents deliberate thinking and abstract problem solving. It can be seen as a strength humans have that we have these two very different systems which we are able to combine. And in fact these two systems complement each other very nicely. Hence, the combination of statistical procedures and logical reasoning holds promise for automated reasoning. The meaning of words, like they are captured in Word Embeddings constitutes an important source of information for automated reasoning systems. In knowledge bases where predicate and function symbols align closely with words, these Word Embeddings can be employed. In previous studies, we have demonstrated the successful integration of word similarities into the selection process, where relevant knowledge for a specific query needs to be extracted from a large knowledge base. Additionally, we incorporated Word Embeddings into the selection of the given clause in the given clause algorithm within resolution provers. Initial experimental results indicate that integrating word similarities leads to provers deriving fewer resolvents and maintaining a more focused approach to the query context.

3.21 Proofs for Quantified Boolean Formulas

Martina Seidl (Johannes Kepler Universität Linz, AT)

Quantified Boolean Formulas (QBFs) extend propositional logic by quantifiers over the Boolean variables. As a consequence of having quantifiers, the decision problem of QBF is PSPACE-complete. There is a symmetry between models of true QBFs and counter-models of false QBFs. Both can be represented as binary trees or as sets of Boolean functions, encoding the solutions of application problems that have been translated to QBFs. In practice, those solutions are often extracted from proofs as produced by the QBF solvers.

The landscape of QBF solving paradigms rather heterogeneous, resulting in solvers are based on various proof systems of different strength. In this talk, we review three different proof systems on which recent solvers are built. In particular, we consider Q-resolution for true and false formulas as found in QCDCL, forall-Exp Res as implemented in expansion-based systems as well as QRAT that was developed for recent pre- and inprocessing techniques.

3.22 More than unit equality

Nick Smallbone (Chalmers University of Technology – Göteborg, SE)

Joint work of Nicholas Smallbone, Koen Claessen
Main reference Nicholas Smallbone: "Twee: An Equational Theorem Prover", in Proc. of the Automated Deduction – CADE 28 – 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings, Lecture Notes in Computer Science, Vol. 12699, pp. 602–613, Springer, 2021.
URL https://doi.org//10.1007/978-3-030-79876-5_35

Equational theorem provers based on Knuth-Bendix completion can solve difficult reasoning problems in, for example, algebra. But the expressive power is limited by the lack of logical connectives. I show that a completion-based prover can reason about practical problems involving connectives with the help of a SAT solver and efficient encodings. I also argue that completion is a useful setting for studying problems in saturation provers, such as how to reason in a goal-directed manner, an important but under-studied problem.

3.23 On hierarchical reasoning and symbol elimination and applications to parametric verification

Viorica Sofronie-Stokkermans (Universität Koblenz, DE)

Joint work of Viorica Sofronie-Stokkermans, Dennis Peuter and Philipp Marohn
Main reference Dennis Peuter, Viorica Sofronie-Stokkermans: "Symbol Elimination and Applications to Parametric Entailment Problems", in Proc. of the Frontiers of Combining Systems – 13th International Symposium, FroCoS 2021, Birmingham, UK, September 8-10, 2021, Proceedings, Lecture Notes in Computer Science, Vol. 12941, pp. 43–62, Springer, 2021.
URL https://doi.org//10.1007/978-3-030-86205-3_3
Main reference Dennis Peuter, Viorica Sofronie-Stokkermans: "On Invariant Synthesis for Parametric Systems", in Proc. of the Automated Deduction – CADE 27 – 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings, Lecture Notes in Computer Science, Vol. 11716, pp. 385–405, Springer, 2019.
URL https://doi.org//10.1007/978-3-030-29436-6_23

We present past and current work on hierarchical symbol elimination.
We first present a goal-oriented symbol elimination method which, given (i) a base theory $\mathcal{T}_0$ allowing quantifier elimination, (ii) an extension $\mathcal{T}_1$ of $\mathcal{T}_0$ with additional function symbols whose properties are axiomatised by a set $\mathcal{K}$ of clauses, (iii) a subset of the additional functions which are considered to be parameters, and (iv) a set G of ground clauses, such that $\mathcal{T}_1 \wedge G$ is satisfiable, computes a universal formula Γ containing symbols in the base theory $\mathcal{T}_0$ and parameters such that $\mathcal{T}_1 \wedge \Gamma \wedge G$ is unsatisfiable. The computation of Γ is done in a hierarchical way, and relies on methods for quantifier elimination in $\mathcal{T}_0$. We identify situations under which the formula Γ computed with our method is the weakest universal formula with the property above, and explain how we used this method for the verification of parametric systems:

1. for generating (weakest) constraints on parameters under which certain properties are guaranteed to be inductive invariants,
2. for iteratively strengthening properties to obtain inductive invariants.

We then briefly present a method for general symbol elimination which uses a constraint resolution calculus obtained from specializing the hierarchical superposition calculus, and explain how we used it – together with goal-oriented symbol elimination – in problems from wireless research theory.

3.24 On Finding Short Proofs

Alexander Steen (Universität Greifswald, DE)

Joint work of Christoph Benzmüller, David Fuenmayor, Alexander Steen, Geoff Sutcliffe
Main reference Christoph Benzmüller, David Fuenmayor, Alexander Steen, Geoff Sutcliffe: "Who Finds the Short Proof?", Logic Journal of the IGPL, p. jzac082, 2023.
URL https://doi.org//10.1093/jigpal/jzac082

The talk reports on an exploration of Boolos' Curious Inference, using higher-order automated theorem provers (ATPs). Surprisingly, only suitable shorthand notations had to be provided by hand for ATPs to find a short proof. The higher-order lemmas required for constructing a short proof are automatically discovered by the ATPs. Given the observations and suggestions in this paper, full proof automation of Boolos' and related examples now seems to be within reach of higher-order ATPs. Preliminary work on automating the synthesis of such shorthand notations is briefly presented.

The talk is based on joint work with Chris Benzmüller, David Fuenmayor and Geoff Sutcliffe [1].

References

1 Christoph Benzmüller, David Fuenmayor, Alexander Steen, Geoff Sutcliffe. *Who Finds the Short Proof? An Exploration of Variants of Boolos' Curious Inference using Higher-order Automated Theorem Provers*. Logic Journal of the IGPL, 2023. DOI: http://doi.org/10.1093/jigpal/jzac082

3.25 TPTP World Standards and Tools for Tarskian and Kripke Interpretations

Geoff Sutcliffe (University of Miami, US), Pascal Fontaine (University of Liège, BE), Jack McKeown (University of Miami, US), and Alexander Steen (Universität Greifswald, DE)

URL https://www.tptp.org/Seminars/TPTPInterpretations/

This talk describes the (new) TPTP World format for representing Tarskian and Kripke interpretations of formulae in classical (FOF, TFF, TXF, THF) and non-classical (NXF, NHF) logics. A technique and implemented tool for verifying models, and a tool for visualizing Tarskian interpretations, are presented. This work provides TPTP World standards that allow interpretations to be shared between components of complex compositional reasoning systems.

3.26 Mechanizing the Splitting Framework

Sophie Tourret (INRIA Nancy – Grand Est, FR)

Joint work of Ghilain Bergeron, Sophie Tourret
Main reference Gabriel Ebner, Jasmin Blanchette, Sophie Tourret: "Unifying Splitting", J. Autom. Reason., Vol. 67(2), p. 16, 2023.
URL https://doi.org//10.1007/S10817-023-09660-8

In this talk, I present the current state of the Isabelle/HOL mechanization efforts by Ghilain Bergeron and myself of the splitting framework by Gabriel Ebner, Jasmin Blanchette and myself. These results include the splitting calculus from section 3 of the framework as well as a partial instance of splitting without backtracking over resolution in FOL. There is still one assumption of this instantiation that is not discharged: the compactness of FOL. Surprisingly, we were unable to find this folklore result in Isabelle/HOL already. I also present the mechanization of this result in Isabelle/HOL via Los's theorem and explain why it is not (yet) usable to discharge the desired assumption of the splitting instance. Finally, I discuss other leads to reach this desired result.

3.27 On the (In-)Completeness of Destructive Equality Resolution in the Superposition Calculus

Uwe Waldmann (MPI für Informatik – Saarbrücken, DE)

Bachmair's and Ganzinger's abstract redundancy concept for the Superposition Calculus justifies almost all operations that are used in superposition provers to delete or simplify clauses, and thus to keep the clause set manageable. Typical examples are tautology deletion, subsumption deletion, and demodulation, and with a more refined definition of redundancy joinability and connectedness can be covered as well. The notable exception is destructive equality resolution, that is, the replacement of a clause $x \not\approx t \vee C$ with $x \notin \mathit{vars}(t)$ by $C\{x \mapsto t\}$. This operation is implemented in state-of-the-art provers, and it is useful in practice, but little is known about how it affects refutational completeness. We demonstrate on the one hand that the naive addition of destructive equality resolution to the standard abstract redundancy concept renders the calculus refutationally incomplete. On the other hand, we present several restricted variants of the superposition calculus that are refutationally complete even with destructive equality resolution.

3.28 The SCL Calculus and its Implementation

Christoph Weidenbach (MPI für Informatik – Saarbrücken, DE)

Main reference Martin Bromberger, Simon Schwarz, Christoph Weidenbach: "SCL(FOL) Revisited", CoRR, Vol. abs/2302.05954, 2023.
URL https://doi.org//10.48550/ARXIV.2302.05954

The talk includes an introduction to the SCL calculus, in particular its version for first-order logic. In addition, I discuss implementation aspects, in particular lifting the CDCL 2-Watched Literal Scheme to first-order logic.

Participants

- Franz Baader
TU Dresden, DE
- Haniel Barbosa
Federal University of Minas Gerais-Belo Horizonte, BR
- Maria Paola Bonacina
University of Verona, IT
- David Déharbe
CLEARSY – Aix-en-Provence, FR
- Martin Desharnais
Max-Planck-Institut für Informatik Saarbrücken, DE
- Clare Dixon
University of Manchester, GB
- Catherine Dubois
ENSIIE – Evry, FR
- Amy Felty
University of Ottawa, CA
- Pascal Fontaine
University of Liège, BE
- Silvio Ghilardi
University of Milan, IT
- Antti Hyvärinen
Certora – Pregassona, CH
- Fuqi Jia
Chinese Academy of Sciences – Beijing, CN
- Chantal Keller
ENS – Gif-sur-Yvette, FR
- Cynthia Kop
Radboud University Nijmegen, NL
- Konstantin Korovin
University of Manchester, GB
- Hanna Lachnitt
Stanford University, US
- Feifei Ma
Chinese Academy of Sciences – Beijing, CN
- Jasper Nalbach
RWTH Aachen, DE
- Claudia Nalon
University of Brasília, BR
- Jakob Nordström
University of Copenhagen, DK & Lund University, SE
- Florian Rabe
Universität Erlangen-Nürnberg, DE
- Andrew Joseph Reynolds
University of Iowa – Iowa City, US
- Claudia Schon
Hochschule Trier, DE
- Stephan Schulz
Duale Hochschule Baden-Württemberg – Stuttgart, DE
- Martina Seidl
Johannes Kepler Universität Linz, AT
- Nick Smallbone
Chalmers University of Technology – Göteborg, SE
- Viorica Sofronie-Stokkermans
Universität Koblenz, DE
- Alexander Steen
Universität Greifswald, DE
- Geoff Sutcliffe
University of Miami, US
- Sophie Tourret
INRIA Nancy – Grand Est, FR
- Uwe Waldmann
MPI für Informatik – Saarbrücken, DE
- Christoph Weidenbach
MPI für Informatik – Saarbrücken, DE
- Akihisa Yamada
AIST – Tokyo, JP