IN PARTNERSHIP WITH:
CNRS

Max Planck Institut für Informatik de Saarbrücken

Université de Lorraine

Activity Report 2017

Project-Team VERIDIS

Modeling and Verification of Distributed Algorithms and Systems

IN COLLABORATION WITH: Laboratoire lorrain de recherche en informatique et ses applications (LORIA)

RESEARCH CENTER
Nancy - Grand Est

THEME
Proofs and Verification

Table of contents

Project-Team VERIDIS

Creation of the Team: 2010 January 01, updated into Project-Team: 2012 July 01

Keywords:

Computer Science and Digital Science:

A2.1.7. - Distributed programming
A2.1.11. - Proof languages
A2.4. - Verification, reliability, certification
A2.4.1. - Analysis
A2.4.2. - Model-checking
A2.4.3. - Proofs
A7.2. - Logic in Computer Science
A8.4. - Computer Algebra

Other Research Topics and Application Domains:

B6.1. - Software industry
B6.3.2. - Network protocols
B6.6. - Embedded systems

1. Personnel

Research Scientists

Jasmin Christian Blanchette [Inria, Starting Research Position, until Feb 2017; external collaborator from Mar 2017]
Stephan Merz [Team leader, Inria, Senior Researcher, HDR]
Thomas Sturm [CNRS, Senior Researcher, HDR]
Sophie Tourret [Max-Planck Institut für Informatik, Researcher, from Oct 2017]
Uwe Waldmann [Max-Planck Institut für Informatik, Senior Researcher]
Christoph Weidenbach [Team leader, Max-Planck Institut für Informatik, Senior Researcher, HDR]

Faculty Members

Marie Duflot-Kremer [Univ. de Lorraine, Associate Professor]
Pascal Fontaine [Univ. de Lorraine, Associate Professor]
Dominique Méry [Univ. de Lorraine, Professor, HDR]
Martin Strecker [Univ. Paul Sabatier Toulouse, Associate Professor, Inria secondment until Aug 2017]

PhD Students

Haniel Barbosa [Univ. de Lorraine, until Oct 2017]
Martin Bromberger [Univ. des Saarlandes]
Margaux Duroeulx [Univ. de Lorraine]
Daniel El Ouraoui [Inria, intern from Mar 2017 until Aug 2017 and PhD student from Oct 2017]
Mathias Fleury [Univ. des Saarlandes]
Souad Kherroubi [Univ. de Lorraine]
Nicolas Schnepf [Inria, joint with Team Madynes]
Hans-Jörg Schurr [Inria, from Nov 2017]
Andreas Teucke [Univ. des Saarlandes]
Marco Voigt [Univ. des Saarlandes]
Daniel Wand [Univ. des Saarlandes, until Jul 2017]

Technical staff

Simon Cruanes [Inria, until Sep 2017]
Martin Riener [Inria, Microsoft-Inria Joint Centre]

Intern

Poonam Kumari [Inria, from Mar 2017 until Jul 2017]

Administrative Assistants

Sophie Drouot [Inria]
Christelle Levêque [Univ. de Lorraine]
Jennifer Müller [Max-Planck Institut für Informatik]

Visiting Scientists

Andrew Reynolds [Univ. of Iowa, from Jul 2017 until Sep 2017]
Tung Vu Xuan [JAIST Kanazawa, until May 2017]

2. Overall Objectives

2.1. Overall Objectives

The VeriDis project team includes members of the MOSEL group at LORIA, the computer science laboratory in Nancy, and members of the research group *Automation of Logic* at Max-Planck-Institut für Informatik in Saarbrücken. It is headed by Stephan Merz and Christoph Weidenbach. VeriDis was created in 2010 as a local research group of Inria Nancy – Grand Est and has been an Inria project team since July 2012.

The objectives of VeriDis are to contribute to advances in verification techniques, including automated and interactive theorem proving, and to make them available for the formal development and analysis of concurrent and distributed algorithms and systems, within the framework of mathematically precise and practically applicable development methods. We intend to assist designers of algorithms and systems in carrying out formally proved developments, where proofs of relevant properties, as well as bugs, can be found with a high degree of automation.

Verification techniques based on theorem proving are already having substantial impact. In particular, they have been successfully applied to the verification and analysis of sequential programs, often in combination with static analysis and software model checking. Ideally, systems and their properties would be specified in high-level, expressive languages, errors in specifications would be discovered automatically, and finally, full verification could also be performed completely automatically. Due to the inherent complexity of the problem, this cannot be achieved in general. We have, however, observed significant advances in theorem proving in recent years. We are particularly interested in the integration of different deduction techniques and tools, such as automated theorem proving for relevant theories, such as different fragments of arithmetic. These advances suggest that a substantially higher degree of automation can be achieved in system verification than what is available in today's verification tools.

VeriDis aims at exploiting and further developing automation in system verification, and at applying its techniques within the context of concurrent and distributed algorithms, which are by now ubiquitous and whose verification is a big challenge. Concurrency problems are central for the development and verification of programs for multi- and many-core architectures, and distributed computation underlies the paradigms of grid and cloud computing. The potential of distributed systems for increased resilience to component failures makes them attractive in many contexts, but also makes formal verification important and challenging. We aim at moving current research in this area to a new level of productivity and quality. To give a concrete example: today the designer of a new distributed protocol may validate it using testing or model checking. Model checking will help finding bugs, but can only guarantee properties of a high-level model of the protocol, usually restricted to finite instances. Testing distributed systems and protocols is notoriously difficult because corner cases are hard to establish and reproduce. Also, many testing techniques require an executable, whose production is expensive and time-consuming, and since an implementation is needed, errors are found only

when they are expensive to fix. The techniques that we develop aim at automatically proving significant properties of the protocol already during the design phase. Our methods mainly target designs and algorithms at high levels of abstraction; we aim at components of operating systems, distributed services, and down to the (mobile) network systems industry.

3. Research Program

3.1. Automated and Interactive Theorem Proving

The VeriDis team gathers experts in techniques and tools for automatic deduction and interactive theorem proving, and specialists in methods and formalisms designed for the development of trustworthy concurrent and distributed systems and algorithms. Our common objective is twofold: first, we wish to advance the state of the art in automated and interactive theorem proving, and their combinations. Second, we work on making the resulting technology available for the computer-aided verification of distributed systems and protocols. In particular, our techniques and tools are intended to support sound methods for the development of trustworthy distributed systems that scale to algorithms relevant for practical applications.

VeriDis members from Saarbrücken are developing SPASS [10], one of the leading automated theorem provers for first-order logic based on the superposition calculus [52]. The group also studies general frameworks for the combination of theories such as the locality principle [64] and automated reasoning mechanisms these induce.

In a complementary approach to automated deduction, VeriDis members from Nancy work on techniques for integrating reasoners for specific theories. They develop veriT [1], an SMT [1] solver that combines decision procedures for different fragments of first-order logic and that integrates an automatic theorem prover for full first-order logic. The veriT solver is designed to produce detailed proofs; this makes it particularly suitable as a component of a robust cooperation of deduction tools.

Finally, VeriDis members design effective quantifier elimination methods and decision procedures for algebraic theories, supported by their efficient implementation in the Redlog system [4].

An important objective of this line of work is the integration of theories in automated deduction. Typical theories of interest, including fragments of arithmetic, are not expressible in first-order logic. We therefore explore efficient, modular techniques for integrating semantic and syntactic reasoning methods, develop novel combination results and techniques for quantifier instantiation. These problems are addressed from both sides, e.g. by embedding decision procedures into the superposition framework or by allowing an SMT solver to accept axiomatizations for plug-in theories. We also develop specific decision procedures for theories such as non-linear real arithmetic that are important when reasoning about certain classes of (e.g., real-time) systems but that also have interesting applications beyond verification.

We rely on interactive theorem provers for reasoning about specifications at a high level of abstraction when fully automatic verification is not (yet) feasible. An interactive proof platform should help verification engineers lay out the proof structure at a sufficiently high level of abstraction; powerful automatic plug-ins should then discharge the resulting proof steps. Members of VeriDis have ample experience in the specification and subsequent machine-assisted, interactive verification of algorithms. In particular, we participate in a project at the joint Microsoft Research-Inria Centre in Saclay on the development of methods and tools for the formal proof of TLA$^+$ [59] specifications. Our prover relies on a declarative proof language, and calls upon several automatic backends [3]. Trust in the correctness of the overall proof can be ensured when the backends provide justifications that can be checked by the trusted kernel of a proof assistant. During the development of a proof, most obligations that are passed to the prover actually fail – for example, because necessary information is not present in the context or because the invariant is too weak, and we are interested in explaining failed proof attempts to the user, in particular through the construction of counter-models.

[1] Satisfiability Modulo Theories [54]

3.2. Formal Methods for Developing and Analyzing Algorithms and Systems

Theorem provers are not used in isolation, but they support the application of sound methodologies for modeling and verifying systems. In this respect, members of VeriDis have gained expertise and recognition in making contributions to formal methods for concurrent and distributed algorithms and systems [2], [9], and in applying them to concrete use cases. In particular, the concept of *refinement* [49], [53], [60] in state-based modeling formalisms is central to our approach because it allows us to present a rational (re)construction of system development. An important goal in designing such methods is to establish precise proof obligations many of which can be discharged by automatic tools. This requires taking into account specific characteristics of certain classes of systems and tailoring the model to concrete computational models. Our research in this area is supported by carrying out case studies for academic and industrial developments. This activity benefits from and influences the development of our proof tools.

In this line of work, we investigate specific development and verification patterns for particular classes of algorithms, in order to reduce the work associated with their verification. We are also interested in applications of formal methods and their associated tools to the development of systems that underlie specific certification requirements in the sense of, e.g., Common Criteria. Finally, we are interested in the adaptation of model checking techniques for verifying actual distributed programs, rather than high-level models.

Today, the formal verification of a new algorithm is typically the subject of a PhD thesis, if it is addressed at all. This situation is not sustainable given the move towards more and more parallelism in mainstream systems: algorithm developers and system designers must be able to productively use verification tools for validating their algorithms and implementations. On a high level, the goal of VeriDis is to make formal verification standard practice for the development of distributed algorithms and systems, just as symbolic model checking has become commonplace in the development of embedded systems and as security analysis for cryptographic protocols is becoming standard practice today. Although the fundamental problems in distributed programming are well-known, they pose new challenges in the context of modern system paradigms, including ad-hoc and overlay networks or peer-to-peer systems, and they must be integrated for concrete applications.

4. Application Domains

4.1. Application Domains

Distributed algorithms and protocols are found at all levels of computing infrastructure, from many-core processors and systems-on-chip to wide-area networks. We are particularly interested in the verification of algorithms that are developed for supporting novel computing paradigms, including ad-hoc networks that underly mobile and low-power computing or overlay networks, peer-to-peer networking that provide services for telecommunication or cloud computing services. Computing infrastructure must be highly available and is ideally invisible to the end user, therefore correctness is crucial. One should note that standard problems of distributed computing such as consensus, group membership or leader election have to be reformulated for the dynamic context of these modern systems. We are not ourselves experts in the design of distributed algorithms, but we work together with domain experts on designing formal models of these protocols, and on verifying their properties. These cooperations help us focus on concrete algorithms and ensure that our work is relevant to the distributed algorithm community.

Formal verification techniques can contribute to certifying the correctness of systems. In particular, they help assert under which assumptions an algorithm or system functions as required. For example, the highest levels of the Common Criteria for Information Technology Security Evaluation encourage the use of formal methods. While initially the requirements of certified development have mostly been restricted to safety-critical systems, the cost of unavailable services due to malfunctioning system components and software provides wider incentives for verification. For example, we have been working on modeling and verifying medical devices that require closed-loop models of both the system and its environment.

5. Highlights of the Year

5.1. Highlights of the Year

Jasmin Blanchette, Mathias Fleury, and Christoph Weidenbach were invited to submit a short version of their IJCAR 2016 paper "A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality" (which had received the Best Paper Award) to the Sister Conference Best Paper Track of IJCAI 2017 [25]. The paper was also invited to a special issue of *Logical Methods in Computer Science*.

The paper "A Formal Proof of the Expressiveness of Deep Learning" [22] by Jasmin Blanchette et al., presented at ITP 2017, has been invited to a special issue of the *Journal of Automated Reasoning*.

The paper "Decidability of the Monadic Shallow Linear First-Order Fragment with Straight Dismatching Constraints" [39] by Andreas Teucke and Christoph Weidenbach presented at CADE 26 has been invited to a special issue of the *Journal of Automated Reasoning*.

Two systems developed in the context of the SMArT project were submitted to the SMT competition SMT-COMP 2017. Redlog won the non-linear real arithmetic (NRA) category, and veriT+Redlog performed nicely on the quantifier-free non-linear real arithmetic (QF_NRA) category.

6. New Software and Platforms

6.1. Redlog

Reduce Logic System
KEYWORDS: Computer algebra system (CAS) - First-order logic - Constraint solving
SCIENTIFIC DESCRIPTION: Redlog is an integral part of the interactive computer algebra system Reduce. It supplements Reduce's comprehensive collection of powerful methods from symbolic computation by supplying more than 100 functions on first-order formulas.

Redlog generally works with interpreted first-order logic in contrast to free first-order logic. Each first-order formula in Redlog must exclusively contain atoms from one particular Redlog-supported theory, which corresponds to a choice of admissible functions and relations with fixed semantics. Redlog-supported theories include Nonlinear Real Arithmetic (Real Closed Fields), Presburger Arithmetic, Parametric QSAT, and many more.
NEWS OF THE YEAR: In 2017, there was a strong focus on applications of Redlog. With the final phase of the ANR-DFG Project SMArT, Redlog was integrated with the SMT solver veriT. That combination, as well as a stand-alone version of Redlog, participated in the SMT competition SMTCOMP 2017. All configurations performed very well, the stand-alone version won the category NRA (nonlinear real arithmetic).

On the scientific side, we made significant progress with the symbolic bifurcation analysis for biological networks.

Redlog technology for biological network analysis from last year, viz. subtropical solving, has raised considerable attention in the SMT community, where it has been adopted and triggered new research.

- Participant: Thomas Sturm
- Contact: Thomas Sturm
- URL: http://www.redlog.eu/

6.2. SPASS

KEYWORD: First-order logic

SCIENTIFIC DESCRIPTION: The classic SPASS is an automated theorem prover based on superposition that handles first-order logic with equality and several extensions for particular classes of theories. With version SPASS 3.9 we have stopped the development of the classic prover and have started the bottom-up development of SPASS 4.0 that will actually be a workbench of automated reasoning tools. Furthermore, we use SPASS 3.9 as a test bed for the development of new calculi.

Meanwhile we have released the second version of SPASS-IQ, our solver for linear integer arithmetic that we are currently extending to real and mixed real-integer arithmetic. We didn't release SPASS-SATT yet, instead we further investigated the use of redundency elimination in SAT solving and underlying implementation techniques. Our aim is a new approach to SAT solving that needs fewer conflicts (on average) *and* is faster than the current state-of-the art solvers. Furthermore, we have developed a new calculus and first prototypical implementation of a SAT solver with mixed OR/XOR clauses.

SPASS 3.9 has been used as the basis for SPASS-AR, an new approximation refinement theorem proving approach.
FUNCTIONAL DESCRIPTION: SPASS is an automated theorem prover based on superposition that handles first-order logic with equality and several extensions for particular classes of theories.

- Contact: Christoph Weidenbach
- URL: http://www.spass-prover.org/

6.3. TLAPS

TLA+ proof system
KEYWORD: Proof assistant
FUNCTIONAL DESCRIPTION: TLAPS is a platform for developing and mechanically verifying proofs about TLA+ specifications. The TLA+ proof language is hierarchical and explicit, allowing a user to decompose the overall proof into proof steps that can be checked independently. TLAPS consists of a proof manager that interprets the proof language and generates a collection of proof obligations that are sent to backend verifiers. The current backends include the tableau-based prover Zenon for first-order logic, Isabelle/TLA+, an encoding of TLA+ set theory as an object logic in the logical framework Isabelle, an SMT backend designed for use with any SMT-lib compatible solver, and an interface to a decision procedure for propositional temporal logic.
NEWS OF THE YEAR: In 2017, we have continued to work on a complete reimplementation of the proof manager. Its objectives are a cleaner interaction with the TLA$^+$ front-ends, in particular SANY, the standard parser and semantic analyzer. The reimplementation is also necessary for extending the scope of the fragment of TLA$^+$ that is handled by TLAPS, in particular full temporal logic and module instantiation.

- Participants: Damien Doligez, Stephan Merz and Martin Riener
- Contact: Stephan Merz
- URL: https://tla.msr-inria.inria.fr/tlaps/content/Home.html

6.4. veriT

KEYWORDS: Automated deduction - Formula solving - Verification
SCIENTIFIC DESCRIPTION: veriT comprises a SAT solver, a decision procedure for uninterpreted symbols based on congruence closure, a simplex-based decision procedure for linear arithmetic, and instantiation-based quantifier handling.
FUNCTIONAL DESCRIPTION: VeriT is an open, trustable and efficient SMT (Satisfiability Modulo Theories) solver, featuring efficient decision procedure for uninterpreted symbols and linear arithmetic, and quantifier reasoning.
NEWS OF THE YEAR: Efforts in 2017 have been focused on non-linear arithmetic reasoning and quantifier handling. The reasoning capabilities of veriT have been significantly improved along those two axes.

The veriT solver participated in the SMT competition SMT-COMP 2017 with good results.

We target applications where validation of formulas is crucial, such as the validation of TLA$^+$ and B specifications, and work together with the developers of the respective verification platforms to make veriT even more useful in practice. The solver is available as a plugin for the Rodin platform, it is integrated within the Atelier B.

- Participants: Haniel Barbosa, Daniel El Ouraoui, Pascal Fontaine and Hans-Jörg Schurr
- Partner: Université de Lorraine
- Contact: Pascal Fontaine
- URL: http://www.veriT-solver.org

6.5. Nunchaku

The Nunchaku Higher-Order Model Finder
KEYWORDS: Proof - Higher-order logic
SCIENTIFIC DESCRIPTION: Nunchaku is a model finder for higher-order logic, with dedicated support for various definitional principles. It is designed to work as a backend for various proof assistants (notably Isabelle/HOL and Coq) and to use state-of-the-art model finders and other solvers as backends.
FUNCTIONAL DESCRIPTION: Nunchaku is a model finder (counterexample generator) for higher-order logic.
NEWS OF THE YEAR: A noteworthy development this year is the creation of a backend called SMBC, based on new ideas by Cruanes about how to combine SAT solving and narrowing.

- Participants: Jasmin Christian Blanchette and Simon Cruanes
- Contact: Jasmin Christian Blanchette
- URL: https://github.com/nunchaku-inria

7. New Results

7.1. Automated and Interactive Theorem Proving

Participants: Haniel Barbosa, Jasmin Christian Blanchette, Martin Bromberger, Simon Cruanes, Daniel El Ouraoui, Mathias Fleury, Pascal Fontaine, Stephan Merz, Martin Riener, Hans-Jörg Schurr, Martin Strecker, Thomas Sturm, Andreas Teucke, Sophie Tourret, Marco Voigt, Tung Vu Xuan, Uwe Waldmann, Daniel Wand, Christoph Weidenbach.

7.1.1. IsaFoL: Isabelle Formalization of Logic

Joint work with Andreas Halkjær From (DTU Copenhagen), Alexander Birch Jensen (DTU Copenhagen), Maximilian Kirchmeier (TU München), Peter Lammich (TU München), John Bruntse Larsen (DTU Copenhagen), Julius Michaelis (TU München), Tobias Nipkow (TU München), Nicolas Peltier (IMAG Grenoble) Anders Schlichtkrull (DTU Copenhagen), Dmitriy Traytel (ETH Zürich), and Jørgen Villadsen (DTU Copenhagen).

Researchers in automated reasoning spend a significant portion of their work time specifying logical calculi and proving metatheorems about them. These proofs are typically carried out with pen and paper, which is error-prone and can be tedious. As proof assistants are becoming easier to use, it makes sense to employ them.

In this spirit, we started an effort, called IsaFoL (Isabelle Formalization of Logic), that aims at developing libraries and methodology for formalizing modern research in the field, using the Isabelle/HOL proof assistant.[2] Our initial emphasis is on established results about propositional and first-order logic. In particular, we are formalizing large parts of Weidenbach's forthcoming textbook, tentatively called *Automated Reasoning—The Art of Generic Problem Solving*.

[2] https://bitbucket.org/isafol/isafol/wiki/Home

The objective of formalization work is not to eliminate paper proofs, but to complement them with rich formal companions. Formalizations help catch mistakes, whether superficial or deep, in specifications and theorems; they make it easy to experiment with changes or variants of concepts; and they help clarify concepts left vague on paper.

The repository contains 14 completed entries and four entries that are still in development. Notably, Mathias Fleury formalized a SAT solver framework with learn, forget, restart, and incrementality. This year he extended it with key optimizations such as the two-watched-literal procedure. The corresponding paper, written together with Jasmin Blanchette and Peter Lammich, was accepted at a highly competitive conference (CPP 2018).

7.1.2. Extension of Term Orders to λ-Free Higher-Order Logic

Superposition is one of the most successful proof calculi for first-order logic today, but in contrast to resolution, tableaux, and connections, it has not yet been generalized to higher-order logic (also called simple type theory). Yet, most proof assistants and many specification languages are based on some variant of higher-order logic.

This motivates us to design a *graceful* generalization of superposition: a proof calculus that behaves like standard superposition on first-order problems and that smoothly scales up to arbitrary higher-order problems. A challenge is that superposition relies on a simplification order, which is fixed in advance of the proof attempt, to prune the search space.

We started our investigations by focusing on a fragment devoid of λ-abstractions, but with partial application and application of variables, two crucial higher-order features. We generalized the two main orders that are used in superposition-based provers today—the lexicographic path order (LPO) [27] and the Knuth-Bendix order (KBO) [21]. The new orders gracefully generalize their first-order counterparts and enjoy nearly all properties needed for superpositions. An exception is compatibility with contexts, which is missing for LPO and some KBO variants. Preliminary work suggests that we can define a version of the superposition calculus that works well in theory and practice (i.e., is refutationally complete and does not lead to a search-space explosion) despite the missing property.

7.1.3. A Fine-Grained Approach of Understanding First-Order Logic Complexity

By the introduction of the separated fragment [65] we have initiated a new framework for a fine-grained understanding of the complexity of fragments of first-order logic, with and without the addition of theories. We have related the classes of the polynomial hierarchy to subclasses of the separated fragment [40] and developed new decidability results [36], [41] based on the techniques of our framework for the combination of the Bernays-Schoenfinkel subfragment with linear arithmetic.

7.1.4. Theorem Proving Based on Approximation-Refinement into the Monadic Shallow Linear Fragment with Straight Dismatching Constraints

We have introduced an approximation-refinement approach for first-order theorem proving based on counterexample-guided abstraction refinement [39]. A given first-order clause set is transformed into an over-approximation contained in the fragment of monadic, shallow, linear clauses with straight dismatching constraints. We have shown the fragment to be decidable, strictly extending known results. If the abstraction obtained that way is satisfiable, so is the original clause set. However, if it is unsatisfiable, then the approximation provides a terminology for lifting the found refutation, step by step, into a proof for the original clause set. If lifting fails, the cause is analyzed to refine the original clause set such that the found refutation is ruled out for the future, and the procedure repeats. We have shown that this approach is superior to all known calculi on certain classes of first-order clauses. In particular, it is able to detect satisfiability of clause sets that have only infinite models.

7.1.5. Combination of Satisfiability Procedures

Joint work with Christophe Ringeissen from the PESTO project-team of Inria Nancy – Grand Est, and Paula Chocron at IIIA-CSIC, Bellaterra, Catalonia, Spain.

A satisfiability problem is often expressed in a combination of theories, and a natural approach consists in solving the problem by combining the satisfiability procedures available for the component theories. This is the purpose of the combination method introduced by Nelson and Oppen. However, in its initial presentation, the Nelson-Oppen combination method requires the theories to be signature-disjoint and stably infinite (to ensure the existence of an infinite model). The design of a generic combination method for non-disjoint unions of theories is difficult, but it is worth exploring simple non-disjoint combinations that appear frequently in verification. An example is the case of shared sets, where sets are represented by unary predicates. Another example is the case of bridging functions between data structures and a target theory (e.g., a fragment of arithmetic).

In 2015, we defined [55] a sound and complete combination procedure à la Nelson-Oppen for the theory of absolutely free data structures (including lists and trees) connected to another theory via bridging functions. This combination procedure has also been refined for standard interpretations. The resulting theory has a nice politeness property, enabling combinations with arbitrary decidable theories of elements. We also investigated other theories [56] amenable to similar combinations: this class includes the theory of equality, the theory of absolutely free data structures, and all the theories in between.

In 2017, we have been improving the framework and unified both results. A new paper is in preparation.

7.1.6. Quantifier Handling in SMT

Joint work with Andrew J. Reynolds, Univ. of Iowa, USA.

SMT solvers generally rely on various instantiation techniques for handling quantifiers. We built a unifying framework encompassing quantified formulas with equality and uninterpreted functions, such that the major instantiation techniques in SMT solving can be cast in that framework. It is based on the problem of E-ground (dis)unification, a variation of the classic Rigid E-unification problem. We introduced a sound and complete calculus to solve this problem in practice: Congruence Closure with Free Variables (CCFV). Experimental evaluations of implementations of CCFV demonstrate notable improvements in the state-of-the-art solver CVC4 and make the solver veriT competitive with state-of-the-art solvers for several benchmark libraries, in particular those originating in verification problems. This was the subject of a publication [20]. In later, unpublished work, we are revisiting enumerative instantiation for SMT. This effort takes place in the context of the Matryoshka project.

7.1.7. Non-Linear Arithmetic in SMT

In the context of the SMArT ANR-DFG (Satisfiability Modulo Arithmetic Theories), KANASA and SC^2 projects (cf. sections 9.1 and 9.3), we study the theory, design techniques, and implement software to push forward the non-linear arithmetic (NLA) reasoning capabilities in SMT. This year, we designed a framework to combine interval constraint propagation with other decision procedures for NLA, with promising results, notably in the international competition of SMT solvers SMT-COMP 2017. We also studied integration of these procedures into combinations of theories. The ideas are validated within the veriT solver, together with code from the raSAT solver (from JAIST). An article is in preparation.

We also adapted the subtropical method to use in an SMT context, with valuable results. This was the subject of a publication in 2017 [33].

7.1.8. Proofs for SMT

We have developed a framework for processing formulas in automatic theorem provers, with generation of detailed proofs. The main components are a generic contextual recursion algorithm and an extensible set of inference rules. Clausification, skolemization, theory-specific simplifications, and expansion of 'let' expressions are instances of this framework. With suitable data structures, proof generation adds only a linear-time overhead, and proofs can be checked in linear time. We implemented the approach in the SMT solver veriT. This allowed us to dramatically simplify the code base while increasing the number of problems for which detailed proofs can be produced, which is important for independent checking and reconstruction in proof assistants. This was the subject of a publication in [19]. This effort takes place in the context of the Matryoshka project.

7.1.9. Coding Modal and Description Logics in SAT solvers

The application scenario behind this research is the verification of graph transformations, which themselves are relevant for a wide range of practical problems such as pointer structures in imperative programs, graph databases or access control mechanisms.

Graph structures can typically be perceived as models of modal logics, and modal logics and variants (such as description logics that are the basis for the web ontology language OWL) are in principle suitable specification formalisms for graph transformations. It turns out, however, that pure modal logics are often not sufficiently expressive for the intended verification purpose and that extensions are needed for which traditional proof methods such as tableau calculi become complex: the termination of the calculi are often very difficult to prove, and huge efforts are required to obtain an efficient implementation.

For these reasons, we have explored methods of encoding the above-mentioned logics in SAT and SMT solvers such as CVC4 and veriT. The idea is to traverse the formula to be verified in order to span up a pre-model that possibly contains more elements (worlds in a Kripke structure) than the real model, and then to run a solver to find out which of these elements can effectively be realized. A prototype has been implemented, with encouraging results. It remains to connect this prototype to the graph verification engine and to publish this work.

7.1.10. Work on the TLA+ Proof System

We continued our work on encoding set-theoretic formulas in multi-sorted first-order logic, and in particular for SMT solvers. Specifically, we unified and streamlined a technique combining an injection of unsorted expressions into sorted languages, simplification by rewriting, and abstraction that underlies the SMT backend of the TLA$^+$ proof system TLAPS. A presentation of our technique was accepted in the journal *Science of Computer Programming*, to appear in 2018.

The proof of the join protocol in a pure-join variant of the Pastry protocol [63] implementing a distributed hash table over a peer-to-peer network is the largest case study carried out so far within TLAPS. Consisting of roughly 30k lines of proof, it was developed as part of Noran Azmy's PhD thesis, defended at the end of 2016 [51]. A presentation of the design of the protocol and its proof was accepted in the journal *Science of Computer Programming*, to appear in 2018.

7.1.11. Automated Analysis of Systems of ODE for Multistationarity

Joint work with R. Bradford and J. Davenport (Bath, UK), M. England (Coventry, UK), H. Errami, C. Hoyt, and A. Weber (Bonn, Germany), V. Gerdt (Dubna, Russia), D. Grigoriev (Lille, France), O. Radulescu (Montpellier, France)

We considered the problem of determining multiple steady states for positive real values in models of biological networks. Investigating the potential for these in models of the mitogen-activated protein kinases (MAPK) network has consumed considerable effort using special insights into the structure of corresponding models. We have applied combinations of symbolic computation methods for mixed equality/inequality systems, specifically automated deduction methods like virtual substitution, lazy real triangularization and cylindrical algebraic decomposition. We have determined multistationarity of an 11-dimensional MAPK network when numeric values are known for all but potentially one parameter. More precisely, our considered model has 11 equations in 11 variables and 19 parameters, 3 of which are of interest for symbolic treatment, and furthermore positivity conditions on all variables and parameters [28].

Subsequent work [31] demonstrates that our techniques benefit tremendously from a new graph theoretical symbolic preprocessing method. We apply our combined techniques to visualize of parameter regions for multistationarity. Comparing computation times and quality of results it turns out that our automated deduction-based approach clearly outperforms established numerical continuation methods.

While automated deduction technology is a bit under the hood here, this interdisciplinary research line addresses important questions related to contemporary research in systems biology. With researchers from that area very actively involved, the results are recognized also within their communities.

7.2. Formal Methods for Developing and Analyzing Algorithms and Systems

Participants: Marie Duflot-Kremer, Margaux Duroeulx, Souad Kherroubi, Poonam Kumari, Dominique Méry, Stephan Merz, Nicolas Schnepf, Christoph Weidenbach.

7.2.1. Making Explicit Domain Knowledge in Formal System Development

Joint work with partners of the IMPEX project.

As explained in the description of the IMPEX project in section 9.1, we advocate that formal modeling languages should explicitly represent the knowledge resulting from an analysis of the application domain, and that ontologies are good candidates for handling explicit domain knowledge. Our objective in doing so is to offer rigorous mechanisms for handling domain knowledge in design models.

We developed the notion of dependency for state-based models. Context-awareness is an important feature in system design. We argue that in proof systems and conceptual modelling this notion should be highlighted precisely. Since we focus on conceptual modelling, understandability and clarity are of high importance. We introduce a new definition [37] for proof context in state-based formalisms with an application to the Event-B modeling language. Furthermore, we introduce a dependency relation between two Event-B models. The contextualization of Event-B models is based on knowledge provided from domains that we classified into constraints, hypotheses and dependencies. The dependency mechanism between two models makes it possible to structure the development of systems models, by organizing phases identified in the analyzed process. These ideas are inspired by work based on the modelling of situations in situation theory that emphasize capabilities of type theory with regard to situation modelling to represent knowledge. Our approach is illustrated on small case studies, and was validated on a development of design patterns for voting protocols.

7.2.2. Incremental Development of Systems and Algorithms

Joint work with Manamiary Bruno Andriamiarina, Neeraj Kumar Singh (IRIT, Toulouse), Rosemary Monahan (NUI Maynooth, Ireland), Zheng Cheng (LINA, Nantes), and Mohammed Mosbah (LaBRI, Bordeaux).

The development of distributed algorithms and, more generally, of distributed systems, is a complex, delicate, and challenging process. The approach based on refinement applies a design methodology that starts from the most abstract model and leads, in an incremental way, to a distributed solution. The use of a proof assistant gives a formal guarantee on the conformance of each refinement with the model preceding it.

Our main result during 2017 is the development of a proved-based pattern for integrating the local computation models and the Visidia platform [32].

7.2.3. Modeling Network Flows in View of Building Security Chains

Joint work with Rémi Badonnel and Abdelkader Lahmadi of the Madynes team of Inria Nancy – Grand Est.

We are working on the application of formal modeling and verification techniques in the area of network communications, and in particular for constructing security functions in a setting of software-defined networks (SDN). Concretely, Nicolas Schnepf defined an extension of the Pyretic language [58] taking into account both the control and the data planes of SDN controllers and implemented a translation of that extension to the input languages of the nuXmv model checker and of SMT solvers. This work was published at NetSoft 2017 [38].

Extending this approach, we have worked on inferring probabilistic finite-state automata models that represent network flows generated by Android applications. The objective is to exploit this representation for generating security chains that detect significant deviations from the behavior represented by the automata and can initiate protective actions. Comparing our models with automata produced by the state-of-the-art tools Invarimint and Synoptic, we obtain representations that are as succinct as those inferred by Invarimint, and significantly smaller than Synoptic, but that include information about transition probability, which Invarimint does not. This work was accepted for publication at NOMS 2018.

7.2.4. Satisfiability Techniques for Reliability Assessment

Joint work with Nicolae Brînzei at Centre de Recherche en Automatique de Nancy.

The reliability of complex systems is typically assessed using probabilistic methods, based on the probabilities of failures of individual components, relying on graphical representations such as fault trees or reliability block diagrams. Mathematically, the dependency of the overall system on the working status of its components is described by its Boolean-valued *structure function*, and binary decision diagrams (BDDs) have been used to construct a succinct representation of that function. We explore the use of modern satisfiability techniques as an alternative to BDD-based algorithms. In [30], we develop three different algorithms for computing minimal tie sets (i.e., component configurations that ensure that the system is functioning). Our algorithms are based on either conjunctive or disjunctive normal form representations of the structure function or on the Hasse diagram representing the configurations. These algorithms have been prototypically implemented in Python, and we are evaluating them on existing benchmarks in order to understand which algorithm works best for typical fault dependencies.

7.2.5. Statistical evaluation of the robustness of production schedules

Joint work with Alexis Aubry, Sara Himmiche, Pascale Marangé, and Jean-François Pétin at Centre de Recherche en Automatique de Nancy.

Finding a good schedule for a production system, especially when it is flexible and when several machines can perform the same operation on products, is a challenging and interesting problem. For a long time, operations research has provided state-of-the-art methods for optimizing scheduling problems. However, approaches based on Discrete Event Systems present interesting alternatives, especially when dealing with uncertainties on the demand or the production time. In this particular case, the flexibility of the automata-based modeling approach is really useful. Using probabilistic timed automata, we demonstrated [35] that statistical model checking can be used successfully for evaluating the robustness of a given schedule w.r.t. probabilistic variations of the processing time. We were thus able to compare different schedules based on their level of service (i.e., the probability that the system will complete the production process within a deadline slightly higher that the schedule time) and their sensitivity (the minimal deadline for which the level of service is greater than a given threshold) [42].

An interdisciplinary workshop on this topic was organized jointly with our colleagues of Centre de Recherche en Automatique and funded by Fédération Charles Hermite.

7.2.6. Using Cubicle for Verifying TLA+ Specifications

Cubicle [3] is a model checker for the verification of parameterized transition systems whose state is described by arrays of variables indexed by an abstract sort representing processes. During her internship, Poonam Kumari designed a translation algorithm from a restricted class of TLA$^+$ specifications into the input language of Cubicle. A prototypical implementation demonstrates the feasibility of the approach, although more work will be necessary to widen the scope of the translation. This work will be continued within the PARDI project, described in section 9.1.

8. Bilateral Contracts and Grants with Industry

8.1. Modeling a Distributed File System

Participant: Stephan Merz.

In a bilateral contract with Huawei R&D, we continued our work on modeling and verifying protocols underlying the Ceph distributed file system [66] in TLA$^+$. We also provided email support to Huawei engineers who use TLA$^+$ for modeling the systems they develop.

8.2. Modeling a Distributed Development Process

Participant: Christoph Weidenbach.

[3] http://cubicle.lri.fr

On the basis of a bilateral contract with L4B (Logic 4 Business), we studied models for a distributed development process of a leading German car manufacturer.

9. Partnerships and Cooperations

9.1. National Initiatives

9.1.1. ANR-DFG Project SMArT

Participants: Haniel Barbosa, Pascal Fontaine, Stephan Merz, Thomas Sturm.

The SMArT (Satisfiability Modulo Arithmetic Theories) project was funded by ANR-DFG Programmes blancs 2013, a bilateral (French-German) program of Agence Nationale de la Recherche and Deutsche Forschungsgemeinschaft DFG. It started in April 2014 and finished in September 2017. The project gathered members of VeriDis in Nancy and Saarbrücken, and the Systerel company.

The objective of the SMArT project was to provide advanced techniques for arithmetic reasoning beyond linear arithmetic for formal system verification, and particularly for SMT. The results feed back into the implementations of Redlog and veriT, which also serve as experimentation platforms for theories, techniques and methods designed within this project.

More information on the project can be found on http://smart.gforge.inria.fr/.

9.1.2. ANR Project IMPEX

Participants: Souad Kherroubi, Dominique Méry.

The ANR Project IMPEX, within the INS program, started in December 2013 for 4 years. It was coordinated by Dominique Méry, the other partners are IRIT/ENSEIHT, Systerel, Supelec, and Telecom Sud Paris. The work reported here also included a cooperation with Pierre Castéran from LaBRI Bordeaux.

Modeling languages provide techniques and tool support for the design, synthesis, and analysis of the models resulting from a given modeling activity, as part of a system development process. These languages quite successfully focus on the analysis of the designed system, exploiting the semantic power of the underlying modeling language. The semantics of this modeling languages are well understood by its users (in particular the system designers), i.e. the semantics is implicit in the model. In general, modeling languages are not equipped with resources, concepts or entities handling explicitly domain engineering features and characteristics (domain knowledge) underlying the modeled systems. Indeed, the designer has to explicitly handle the knowledge resulting from an analysis of this application domain [61], i.e. explicit semantics. At present, making explicit the domain knowledge inside system design models does not obey any methodological rules validated by practice. The users of modeling languages introduce these domain knowledge features through types, constraints, profiles, etc. Our claim is that ontologies are good candidates for handling explicit domain knowledge. They define domain theories and provide resources for uniquely identifying domain knowledge concepts. Therefore, allowing models to make references to ontologies is a modular solution for models to explicitly handle domain knowledge. Overcoming the absence of explicit semantics expression in the modeling languages used to specify systems models will increase the robustness of the designed system models. Indeed, the axioms and theorems resulting from the ontologies can be used to strengthen the properties of the designed models. The objective [50] is to offer rigorous mechanisms for handling domain knowledge in design models.

9.1.3. ANR Project Formedicis

Participant: Dominique Méry.

The ANR Project Formedicis, within the INS program, started in January 2017 for 4 years. It is coordinated by Bruno d'Augsbourg, the partners are ONERA, IRIT/ENSEIHT, ENAC, and LORIA.

During the last 30 years, the aerospace domain has successfully devised rigorous methods and tools for the development of safe functionally-correct software. During this process, interactive software has received a relatively lower amount of attention. However, Human-System Interactions (HSI) are important for critical systems and especially in aeronautics: new generations of aircraft cockpits make use of sophisticated electronic devices that may be driven by more and more complex software applications. The criticality of these applications require a high degree of assurance for their intended behavior. The report by the French *Bureau d'Enquêtes et d'Analyses* about the crash of the Rio-Paris flight AF 447 in 2009 pointed out a design issue in the behavior of the Flight Director interface as one of the original causes of the crash.

We believe that part of these issues are due to the lack of a well-defined domain specific "hub" language to represent interactive software design in a way that allows system designers to iterate on their designs before injecting them in a development process, and system developers to verify their software against the chosen design. Formedicis aims at designing such a formal hub language L, in which designers can express their requirements concerning the interactive behavior that must be embedded inside the interactive applications. The project will also develop a framework for validating, verifying, and implementing critical interactive applications designed and denoted in L.

More information on the project is available at http://www.agence-nationale-recherche.fr/Project-ANR-16-CE25-0007.

9.1.4. ANR Project PARDI

Participants: Marie Duflot-Kremer, Stephan Merz.

PARDI (Verification of parameterized distributed systems) is funded by ANR. The project started in January 2017 for a duration of 48 months. The project partners other than VeriDis are Toulouse INP (coordinator), Université Paris Sud, and Université Paris Marie Curie.

Distributed systems and algorithms are parameterized by the number of participating processes, the communication model, the fault model, and more generally the properties of interaction among the processes. The project aims at providing methodological and tool support for verifying parameterized systems, using combinations of model checking and theorem proving. VeriDis contributes its expertise on TLA$^+$ and its verification tools, and the integration with the Cubicle model checker is a specific goal of the project.

More information on the project is available at http://pardi.enseeiht.fr/.

9.1.5. Inria IPL HAC SPECIS

Participants: Marie Duflot-Kremer, Stephan Merz.

The goal of the HAC SPECIS (High-performance Application and Computers: Studying PErformance and Correctness In Simulation) project is to answer methodological needs of HPC application and runtime developers and to allow studying real HPC systems with respect to both correctness and performance. To this end, this Inria Project Lab assembles experts from the HPC, formal verification, and performance evaluation communities.

HAC SPECIS started in 2016. VeriDis contributes through its expertise in formal verification techniques. In particular, our goal is to extend the functionalities of exhaustive and statistical model checking within the SimGrid platform.

9.1.6. Inria Technological Development Action CUIC

Participants: Jasmin Christian Blanchette, Simon Cruanes.

Most "theorems" initially given to a proof assistant are incorrect, whether because of a typo, a missing assumption, or a fundamental flaw. Novices and experts alike can enter invalid formulas and find themselves wasting hours, or even days, on an impossible proof. This project, funded by Inria and running from 2015 to 2017, supported the development of a counterexample generator for higher-order logic. This new tool, called Nunchaku, is intended for integration with various proof assistants. The project was coordinated by Jasmin Blanchette and also involved Inria Saclay – Île de France (Toccata group) and Inria Rennes – Bretagne

Atlantique (Celtique group), among others. Simon Cruanes worked on Nunchaku from October 2015 to September 2017, whereas Blanchette has developed an Isabelle frontend. Four releases have taken place so far, and the tool is an integral part of the Isabelle2017 official release. Work has started on Coq and TLAPS frontends, and we will soon work on a Lean frontend as well. The tool is described in [62] and was presented at a workshop last year [57]. A noteworthy development this year is the creation of a backend called SMBC, based on new ideas by Cruanes about how to combine SAT solving and narrowing [29].

9.2. European Initiatives

9.2.1. FP7 & H2020 Projects

9.2.1.1. ERC Matryoshka

Program: European Union's Horizon 2020 research and innovation program

Project acronym: Matryoshka

Duration: April 2017 – March 2022

Coordinator: Jasmin Blanchette (VU Amsterdam)

Proof assistants are increasingly used to verify hardware and software and to formalize mathematics. However, despite the success stories, they remain very laborious to use. The situation has improved with the integration of first-order automatic theorem provers—superposition provers and SMT (satisfiability modulo theories) solvers—through middleware such as Sledgehammer for Isabelle/HOL and HOLyHammer for HOL Light and HOL4; but this research has now reached the point of diminishing returns. Only so much can be done when viewing automatic provers as black boxes.

To make interactive verification more cost-effective, we propose to deliver very high levels of automation to users of proof assistants by fusing and extending two lines of research: automatic and interactive theorem proving. This is our grand challenge. Our starting point is that first-order (FO) automatic provers are the best tools available for performing most of the logical work. Our approach will be to enrich superposition and SMT with higher-order (HO) reasoning in a careful manner, in order to preserve their desirable properties. We will design proof rules and strategies, guided by representative benchmarks from interactive verification.

With higher-order superposition and higher-order SMT in place, we will develop highly automatic provers building on modern superposition provers and SMT solvers, following a novel stratified architecture. To reach end users, these new provers will be integrated in proof assistants and will be available as backends to more specialized verification tools. The users of proof assistants and similar tools stand to experience substantial productivity gains: From 2010 to 2016, the success rate of automatic provers on interactive proof obligations from a representative benchmark suite called Judgment Day has risen from 47% to 77%; with this project, we aim at 90%–95% proof automation.

The Matryoshka ERC grant of Jasmin Blanchette includes Pascal Fontaine and Uwe Waldmann as senior researchers.

9.2.1.2. FET-Open CSA SC^2

Program: European Union's Horizon 2020 research and innovation program

Project acronym: SC^2

Project title: Symbolic Computation and Satisfiability Checking

Duration: July 2016 – August 2018

Coordinator: James Davenport (U. of Bath, UK)

Other partners: see http://www.sc-square.org/CSA/welcome.html

The use of advanced methods for solving practical and industrially relevant problems by computers has a long history. Whereas Symbolic Computation is concerned with the algorithmic determination of exact solutions to complex mathematical problems, more recent developments in the area of Satisfiability Checking tackle similar problems but with different algorithmic and technological solutions.

Though both communities have made remarkable progress in the last decades, they still need to be strengthened to tackle practical problems of rapidly increasing size and complexity. Their separate tools (computer algebra systems and SMT solvers) are urgently needed to examine prevailing problems with a direct effect to our society. For example, Satisfiability Checking is an essential backend for assuring the security and the safety of computer systems. In various scientific areas, Symbolic Computation enables dealing with large mathematical problems out of reach of pencil and paper developments.

Currently the two communities are largely disjoint and unaware of the achievements of each other, despite strong reasons for them to discuss and collaborate, as they share many central interests. However, researchers from these two communities rarely interact, and also their tools lack common, mutual interfaces for unifiying their strengths. Bridges between the communities in the form of common platforms and roadmaps are necessary to initiate an exchange, and to support and to direct their interaction. These are the main objectives of this CSA. We initiate a wide range of activities to bring the two communities together, identify common challenges, offer global events and bilateral visits, propose standards, and so on.

We believe that these activities will foster cross-fertilisation of both fields and bring mutual improvements. Combining the knowledge, experience and the technologies in these communities will enable the development of radically improved software tools.

This project is locally coordinated by Pascal Fontaine.

9.3. International Initiatives

9.3.1. Inria International Partners

Title: Kanazawa-Nancy for Satistifiability and Arithmetics (KANASA)

International Partner: Japan Advanced Institute for Science and Technology (Dept. Intelligent Robotics, Mizuhito Ogawa)

Starting year: 2016

During the last decade, there has been tremendous progress on symbolic verification techniques, spurred in particular by the development of SMT (satisfiability modulo theories) techniques and tools. Our first direction of research will be to investigate the theoretical background and the practical techniques to integrate Interval Constraint Propagation within a generic SMT framework, including other decision procedures and quantifier handling techniques. On the purely arithmetic side, we also want to study how to unite the reasoning power of all arithmetic techniques developed in the team, including simplex-based SMT-like reasoners, Virtual Substitution, and Cylindrical Algebraic Decomposition. In particular, this includes developing theory combination frameworks for linear and non-linear arithmetic. There is a strong incentive for these kind of combinations since even non-linear SMT problems contain a large proportion of linear constraints. The partnership is supported by a Memorandum of Understanding between JAIST and LORIA.

One PhD student from JAIST spent one year in the VeriDiS team, until May 2017. The partnership evolves towards applying SMT to find malware in obfuscated code.

9.4. International Research Visitors

9.4.1. Visits of International Scientists

Tung Vu Xuan

Date: 1 May 2016 – 30 April 2017

Institution: JAIST

Host: Pascal Fontaine

Tung Vu Xuan is a PhD student at JAIST, Japan. He was visiting VeriDis in the context of the KANASA project. He works mainly on Interval Constraint Propagation (ICP), a heuristic but powerful method for satisfiability checking of non-linear arithmetic (NLA) constraints. During his stay, we investigated techniques to combine ICP with decision procedures for NLA within an SMT context, and adapted the subtropical method from computer algebra to the context of SMT. This work is relevant for the SMArT and SC^2 projects.

Andrew J. Reynolds

Date: 16 July 2017 – 17 September 2017

Institution: The University of Iowa

Host: Pascal Fontaine

Andrew J. Reynolds is a Research Scientist at the University of Iowa and one of main developers of the award-winning Satisfiability Modulo Theories (SMT) solver CVC4. His current research interests include implementing techniques in SMT solvers for unbounded strings and regular expressions, first-order quantified formulas and synthesis conjectures. He was an Inria invited researcher for two months in Nancy. We continued working on quantifier handling for SMT, along the lines of [20], and studied enumerative instantiation. This work contributes to the Matryoshka, SMArT and SC^2 projects.

9.4.2. Internships

Poonam Kumari

Date: 1 March – 31 July

Institution: Université de Lorraine (Erasmus Mundus DESEM)

Host: Stephan Merz

Poonam Kumari worked on a translation from a restricted subset of TLA^+ specifications into the input language of the Cubicle model checker for array-based parameterized systems.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Organization of Scientific Events

10.1.1.1. Member of the Organizing Committees

Jasmin Blanchette co-organized the *(Co)programming in Isabelle/HOL* tutorials at ICFP 2017 in Oxford, UK, and at CADE-26 in Gothenburg, Sweden.

Jasmin Blanchette co-organized the Dagstuhl Seminar on *Deduction beyond First-Order Logic* held at Schloss Dagstuhl in Germany.

Jasmin Blanchette co-organized the *(Co)programming in Isabelle/HOL* tutorials at ICFP 2017 in Oxford, UK, and at CADE-26 in Gothenburg, Sweden.

Dominique Méry was a member of the organizing committees of the workshops F-IDE [43] and IMPEX'2017.

The International Summer School on Verification Techniques, Systems, and Applications (VTSA) has been organized since 2008 in the Greater Region (Nancy, Saarbrücken, Luxembourg, Liège, and Koblenz), and Stephan Merz and Christoph Weidenbach are co-organizers of VTSA. In 2017, VTSA took place in July in Saarbrücken, Germany.

The SC^2 Summer School 2017 took place in Saarbrücken, Germany. It has been co-organized by Thomas Sturm. The school introduced graduate students and researchers from academia and industry into research and methodology in both Satisfiability Checking (SAT/SMT) and Symbolic Computation with one focus on their interconnections. It combined a thorough introduction into the theory of both fields with lectures on state-of-the-art software systems and their implementation. This was supplemented with presentations by lecturers from industry discussing the practical relevance of the topics of the school.

Together with the CADE trustees, Christoph Weidenbach started the first CADE workshop on *Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements* (ARCADE 2017)

10.1.2. Program Committees

10.1.2.1. Chair of Conference Program Committees

Stephan Merz co-chaired the program committee of the Fourth International Workshop on Formal Reasoning in Distributed Algorithms (FRiDA), organized in October 2017 as a satellite of DISC in Vienna, Austria.

10.1.2.2. Member of the Conference Program Committees

Jasmin Blanchette served on the program committees of the Conference on *Computer-Aided Verification* (CAV 2017), the *Conference on Automated Deduction* (CADE-26), the *International Conference on Tests and Proofs* (TAP 2017), and the Conference on *Artificial Intelligence and Theorem Proving* (AITP 2017). He also served on the following workshop committees: *Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements* (ARCADE 2017), *International Workshop on the Implementation of Logics* (IWIL 2017), *Proof Exchange for Theorem Proving* (PxTP 2017), and *Satisfiability Modulo Theories* (SMT 2017).

Pascal Fontaine served on the program committees of the *International Symposium on Frontiers of Combining Systems* (FroCoS 2017), the *Conference on Automated Deduction* (CADE-26) and the *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (TABLEAUX 2017). He also served on the following workshop committees: *Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements* (ARCADE 2017), *Satisfiability Modulo Theories* (SMT 2017), *Satisfiability Checking and Symbolic Computation* (SC^2 2017), *Proof Exchange for Theorem Proving* (PxTP 2017)

Stephan Merz served on the program committees of the international conferences *Formal Techniques for Distributed Objects, Components, and Systems* (FORTE 2017), *Foundations of Software Technology and Theoretical Computer Science* (FSTTCS 2017), and *Formal Engineering Methods* (ICFEM 2017), the national conference *Modélisation des Systèmes Réactifs* (MSR 2017), and of the workshops FMICS-AVoCS and GRSRD.

Thomas Sturm served on the program committees of the *Second International Workshop on Satisfiability Checking and Symbolic Computation* (SC^2 2017) and the *19th International Workshop on Computer Algebra in Scientific Computing* (CASC 2017).

Uwe Waldmann served on the program committee of the workshop *International Workshop on the Implementation of Logics* (IWIL 2017) colocated with LPAR.

Christoph Weidenbach served on the program committees of the *Conference on Automated Deduction* (CADE-26) and the *International Symposium on Frontiers of Combining Systems* (FroCoS 2017). He also served on the workshop committee *Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements* (ARCADE 2017).

10.1.3. Journals

Jasmin Blanchette and Stephan Merz are the editors of a special issue of *Journal of Automated Reasoning* following the international conference *Interactive Theorem Proving* 2016.

Dominique Méry is the review book editor of the journal *Formal Aspects for Computing*.

Thomas Sturm is a member of the editorial boards of the *Journal of Symbolic Computation* (Elsevier) and *Mathematics in Computer Science* (Springer).

Christoph Weidenbach is a member of the editorial board of the *Journal of Automated Reasoning* (Springer).

10.1.4. Invited Talks

Jasmin Blanchette was invited to give a joint keynote talk at the FroCoS 2017, ITP 2017, and TABLEAUX 2017 conferences held in Brasília, Brazil. He presented Isabelle/HOL's support for (co)datatypes and (co)recursion [18]. He also gave invited seminar talks at the Big Proof Workshop organized by the Isaac Newton Institute in Cambridge, UK, at the TeReSe (Term Rewriting Systems) meeting in Eindhoven, the Netherlands, and at the Shonan Meeting on Automated Deduction at the Shonan Village Center in Japan.

Stephan Merz gave an invited presentation on "Formal Methods for the Cloud" at the Cloud Resiliency Workshop 2017 in Shenzhen, China.

Thomas Sturm was invited to give a keynote talk at the *3ème BIOSS Journées annuelles du groupe de travail* in Montpellier, France.

Uwe Waldmann gave an invited talk on "Saturation Theorem Proving – Basic Ideas, History, and Recent Developments" at the Seminar on Proof Assistants and Related Tools at DTU Lyngby, Denmark in October 2017.

Christoph Weidenbach gave invited talks on "Design Principles of Automated Reasoning Systems" at VSTTE 2017 and "The Role of Horn Clauses in Automatic Reasoning" at HCVS 2017.

10.1.5. Leadership within the Scientific Community

Jasmin Blanchette was elected as a regular member of the steering committee for the ITP (*Interactive Theorem Proving*) conference series, after serving for two years as an ex officio member. He is also a regular member of the CADE (*Conference on Automated Deduction*) Inc. Board of Trustees.

Pascal Fontaine is an SMT-LIB manager, together with Clark Barrett (Stanford University) and Cesare Tinelli (University of Iowa). He is a member of the FroCoS steering committee. He was an elected CADE trustee since October 2014 until October 2017 and served as a member of the Association for Automated Reasoning (AAR) board until October 2017.

Stephan Merz is a member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*. He is a member of the steering committee of the workshop on Automated Verification of Critical Systems (AVoCS).

Thomas Sturm has been a member of the steering committee of the conference series *International Conference on Mathematical Aspects of Computer and Information Sciences (MACIS).* His term ended in November 2017. In July 2017 he was elected as a member at large of the steering committee of the conference series *International Symposium on Symbolic and Algebraic Computation (ISSAC).*

Christoph Weidenbach is the president of CADE and a member of the steering committee of IJCAR.

10.1.6. Scientific Expertise

Pascal Fontaine was a panel member for the CASC-26 competition of first-order theorem prover. He served as an expert for the French Agence Nationale de la Recherche (ANR).

Stephan Merz served as an expert for the French Agence Nationale de la Recherche (ANR) and for the European Research Council (ERC).

Christoph Weidenbach served as an expert for the Austrian Science Fund and the University of Stellenbosch.

10.1.7. Research Administration

Dominique Méry was the head of the Doctoral School IAEM Lorraine of University of Lorraine until September 2017.

Stephan Merz is the delegate for scientific affairs at the Inria Nancy – Grand Est research center and a member of Inria's Evaluation Committee. In 2017, he was a member of the hiring committees of junior researchers at Inria Saclay – Île de France as well as of senior researchers at Inria. He is a member of the committee for the SIF thesis award (*Prix Gilles Kahn*). He is a member of the *bureau* of the computer science committee of the doctoral school IAEM Lorraine. Until October 2017, he was a member of the Scientific Directorate of the International Computer Science Meeting Center in Schloss Dagstuhl.

Christoph Weidenbach is a member of the selection committee of the Saarbrücken Graduate School in Computer Science.

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

Master: Jasmin Blanchette, Logical Verification, 36 HETD, M1/M2, Vrije Universiteit Amsterdam, the Netherlands.

Licence: Marie Duflot-Kremer, Algorithmique et Programmation 1, 70 HETD L1 Mathématiques, Informatiques Sciences pour l'Ingénieur, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Bases de données 2, 20 HETD, L2 Informatique, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Projet personnel et communication, 60 HETD, L2 Informatique, Université de Lorraine, France.

Master : Marie Duflot-Kremer, Vérification de systèmes, 30 HETD, M1 Informatique, Université de Lorraine, France.

Master: Marie Duflot-Kremer and Stephan Merz, Vérification algorithmique, 40 HETD, M2 Informatique, Université de Lorraine, France.

Master : Marie Duflot-Kremer and Stephan Merz, Conception et architectures distribuées 24 HETD M1 informatique, Université de Lorraine, France.

Licence : Pascal Fontaine, Structure des ordinateurs, 47 HETD, L2 MIASHS, parcours MIAGE, Université de Lorraine, France.

Master : Pascal Fontaine, Réseaux, 50 HETD, M1 MIAGE, Université de Lorraine, France.

Master : Pascal Fontaine, Génie Logiciel, 30 HETD, M1 MIAGE, IGA Rabbat et Université de Lorraine, Maroc.

Master: Dominique Méry, Models and algorithms, 60 HETD, M1, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Formal model engineering, 24 HETD, M2, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Modeling Systems, 30 HETD, M2, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Modeling Systems, 36 HETD, M2 Informatique and Master Erasmus Mundus DESEM, Université de Lorraine, France.

Master: Dominique Méry, Event-B modeling, 8 HETD, NUI Maynooth, Ireland.

Master: Stephan Merz, Modeling and Verifying Distributed Algorithms in TLA$^+$, 8 HETD, NUI Maynooth, Ireland.

Master: Christoph Weidenbach, Automated Reasoning I & II, 150 HETD, Universität des Saarlandes, Germany.

10.2.2. Supervision

PhD: Haniel Barbosa, New techniques for instantiation and proof production in SMT solving, Université de Lorraine and UFRN (Natal, Brazil) [11]. Supervised by David Déharbe, Pascal Fontaine, and Stephan Merz, since 12/2013. Defended on September 5, 2017.

PhD: Andreas Teucke, *An Approximation and Refinement Approach to First-Order Automated Reasoning*, Saarland University. Supervised by Christoph Weidenbach, thesis submitted in October 2017.

PhD: Daniel Wand, First-Order Extensions to Support Higher-Order Reasoning, Saarland University [12]. Supervised by Christoph Weidenbach and Jasmin Blanchette, since 02/2011. Defended on August 4, 2017.

PhD in progress: Martin Bromberger, Arithmetic Reasoning, Saarland University. Supervised by Christoph Weidenbach, since July 2014.

PhD in progress: Margaux Duroeulx, SAT Techniques for Reliability Assessment, Université de Lorraine. Supervised by Nicolae Brînzei, Marie Duflot-Kremer, and Stephan Merz, since October 2016.

PhD in progress: Daniel El Ouraoui, Higher-Order SMT, Université de Lorraine. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, since November 2017.

PhD in progress: Mathias Fleury, Formalization of Logical Calculi, Saarland University. Supervised by Christoph Weidenbach and Jasmin Blanchette, since September 2015.

PhD in progress: Souad Kherroubi, A framework to formally handle domain knowledge in system design, Université de Lorraine. Supervised by Dominique Méry, since November 2014.

PhD in progress: Nicolas Schnepf, Orchestration and Verification of Security Functions for Smart Environments, Université de Lorraine. Supervised by Rémi Badonnel, Abdelkader Lahmadi, and Stephan Merz, since October 2016.

PhD in progress: Hans-Jörg Schurr, Higher-Order SMT, Université de Lorraine. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, since November 2017.

PhD in progress: Marco Voigt, Decidable Hierarchic Combinations, Saarland University. Supervised by Christoph Weidenbach, since November 2013.

10.2.3. Thesis committees

Stephan Merz served as a reviewer for the PhD theses of Florent Chevrou (Univ. de Toulouse), Sebastian Krings (Univ. Düsseldorf), Ognjen Marič (ETH Zurich), and Yannick Zakowski (ENS Rennes). He was an examiner for the PhD thesis of Zeinab Bakhtiarinoodeh (Univ. de Lorraine) and the habilitation of Alain Giorgetti (Univ. de Bourgogne et Franche Comté).

10.3. Science outreach

Marie Duflot-Kremer took part in various science outreach activities, with a public ranging from primary school kids to teachers and potential university students. A selection of these activities is given below.

General activities.

She is responsible for the scientific part of the fifth and last module in the Class'Code project (supervising a programming project from scratch), aiming at training teachers and educators for carrying out computer science activities with childrens aged 8 to 14 years.

Three days at “Fête de la Science” in Nancy (Faculté des Sciences et Technologies) and at Cité des Sciences, Paris, including a visit of the Minister of Higher Education and Research, Frédérique Vidal.

She is a member of the steering committee of an itinerant exhibition, intended for explaining computer science to the public, and that had its opening in early 2017.

She co-organized the SCRATCH17BDX international conference on Scratch and creative programming for kids in Bordeaux.

Activities for teachers/trainers.

She is a member of three working groups (unplugged activities, programming in secondary school and in high school) including university and secondary school teachers, dedicated to the training of math teachers. Two days of training on unplugged computer science activities were given to secondary and high school teachers.

A training session for kindergarden teachers to include in their “school project” unplugged activities related to programming for kids from 3 to 6 years old.

Several activities for the “ISN day”, aimed at high school teachers teaching computer science courses.

A publication (post proceedings to appear in 2018) was accepted at the COPIRELEM colloquium, aimed at math trainers for primary school teachers.

Activities for students/pupils.

Several activities for school kids from 6 to 10 years old at Ecole Marcel Leroy, Nancy.

She was involved in the *Math en Jeans* project where secondary school kids discover what doing research means.

Various outreach activities (related to data bases, model checking, algorithms etc.) during two days aimed at presenting the university to high school students.

11. Bibliography

Major publications by the team in recent years

[1] T. BOUTON, D. C. B. DE OLIVEIRA, D. DÉHARBE, P. FONTAINE. *veriT: an open, trustable and efficient SMT-solver*, in "Proc. Conference on Automated Deduction (CADE)", Montreal, Canada, R. SCHMIDT (editor), Lecture Notes in Computer Science, Springer, 2009, vol. 5663, pp. 151-156

[2] D. CANSELL, D. MÉRY. *The Event-B Modelling Method: Concepts and Case Studies*, in "Logics of Specification Languages", Berlin-Heidelberg, D. BJØRNER, M. C. HENSON (editors), Monographs in Theoretical Computer Science, Springer, 2008, pp. 47-152

[3] D. COUSINEAU, D. DOLIGEZ, L. LAMPORT, S. MERZ, D. RICKETTS, H. VANZETTO. *TLA+ Proofs*, in "18th International Symposium On Formal Methods - FM 2012", Paris, France, D. GIANNAKOPOULOU, D. MÉRY (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7436, pp. 147-154

[4] A. DOLZMANN, T. STURM. *Redlog: Computer algebra meets computer logic*, in "ACM SIGSAM Bull.", 1997, vol. 31, n^{o} 2, pp. 2-9

[5] D. DÉHARBE, P. FONTAINE, S. MERZ, B. WOLTZENLOGEL PALEO. *Exploiting Symmetry in SMT Problems*, in "23rd Intl. Conf. Automated Deduction (CADE 2011)", Wroclaw, Poland, N. BJØRNER, V. SOFRONIE-STOKKERMANS (editors), LNCS, Springer, 2011, vol. 6803, pp. 222-236

[6] A. FIETZKE, C. WEIDENBACH. *Superposition as a Decision Procedure for Timed Automata*, in "Mathematics in Computer Science", 2012, vol. 6, n^{o} 4, pp. 409-425

[7] E. KRUGLOV, C. WEIDENBACH. *Superposition Decides the First-Order Logic Fragment Over Ground Theories*, in "Mathematics in Computer Science", 2012, vol. 6, n^{o} 4, pp. 427-456

[8] F. KRÖGER, S. MERZ. *Temporal Logic and State Systems*, Texts in Theoretical Computer Science, Springer, 2008, 436 p. , http://hal.inria.fr/inria-00274806/en/

[9] S. MERZ. *The Specification Language TLA^{+}*, in "Logics of Specification Languages", Berlin-Heidelberg, D. BJØRNER, M. C. HENSON (editors), Monographs in Theoretical Computer Science, Springer, 2008, pp. 401-451

[10] C. WEIDENBACH, D. DIMOVA, A. FIETZKE, M. SUDA, P. WISCHNEWSKI. *SPASS Version 3.5*, in "22nd International Conference on Automated Deduction (CADE-22)", Montreal, Canada, R. SCHMIDT (editor), LNAI, Springer, 2009, vol. 5663, pp. 140-145

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] H. BARBOSA. *New techniques for instantiation and proof production in SMT solving*, Université de Lorraine, September 2017, https://tel.archives-ouvertes.fr/tel-01591108

[12] D. WAND. *Superposition: Types and Induction*, Saarland University, August 2017, https://hal.inria.fr/tel-01592497

Articles in International Peer-Reviewed Journals

[13] J. C. BLANCHETTE, A. POPESCU, D. TRAYTEL. *Soundness and Completeness Proofs by Coinductive Methods*, in "Journal of Automated Reasoning", January 2017, vol. 58, n^{o} 1, pp. 149 - 179 [*DOI :* 10.1007/s10817-016-9391-3], https://hal.inria.fr/hal-01643157

[14] M. BROMBERGER, C. WEIDENBACH. *New Techniques for Linear Arithmetic: Cubes and Equalities*, in "Formal Methods in System Design", 2017, vol. 51, n^{o} 3, pp. 433-461 [*DOI :* 10.1007/s10703-017-0278-7], https://hal.inria.fr/hal-01656397

[15] D. MÉRY. *Playing with State-Based Models for Designing Better Algorithms*, in "Future Generation Computer Systems", March 2017, vol. 68, pp. 445-455, https://hal.inria.fr/hal-01316026

[16] D. MÉRY, M. POPPLETON. *Towards An Integrated Formal Method for Verification of Liveness Properties in Distributed Systems: with application to Population Protocols*, in "Software and Systems Modeling (SoSyM)", October 2017, vol. 16, n^{o} 4, pp. 1083–1115, https://hal.inria.fr/hal-01245819

[17] T. STURM. *A Survey of Some Methods for Real Quantifier Elimination, Decision, and Satisfiability and Their Applications*, in "Mathematics in Computer Science", December 2017, vol. 11, n° 3-4, pp. 483 - 502 [*DOI :* 10.1007/s11786-017-0319-z], https://hal.inria.fr/hal-01648690

Invited Conferences

[18] J. BIENDARRA, J. C. BLANCHETTE, A. BOUZY, M. DESHARNAIS, M. FLEURY, J. HÖLZL, O. KUNČAR, A. LOCHBIHLER, F. MEIER, L. PANNY, A. POPESCU, C. C. STERNAGEL, R. THIEMANN, D. TRAYTEL. *Foundational (Co)datatypes and (Co)recursion for Higher-Order Logic*, in "Frontiers of Combining Systems, 11th International Symposium", Brasilia, Brazil, C. DIXON, M. FINGER (editors), Lecture Notes in Computer Science, Springer, September 2017, vol. 10483, pp. 3-21 [*DOI :* 10.1007/978-3-319-66167-4_1], https://hal.inria.fr/hal-01592196

International Conferences with Proceedings

[19] H. BARBOSA, J. C. BLANCHETTE, P. FONTAINE. *Scalable Fine-Grained Proofs for Formula Processing*, in "Proc. Conference on Automated Deduction (CADE)", Gotenburg, Sweden, L. DE MOURA (editor), Lecture Notes in Computer Science, Springer, 2017, vol. 10395, pp. 398 - 412 [*DOI :* 10.1007/978-3-642-02959-2_10], https://hal.inria.fr/hal-01590922

[20] H. BARBOSA, P. FONTAINE, A. REYNOLDS. *Congruence Closure with Free Variables*, in "Tools and Algorithms for Construction and Analysis of Systems (TACAS)", Uppsala, Sweden, 2017, vol. 205, pp. 220 - 230 [*DOI :* 10.1007/10721959_17], https://hal.inria.fr/hal-01590918

[21] H. BECKER, J. C. BLANCHETTE, U. WALDMANN, D. WAND. *A Transfinite Knuth-Bendix Order for Lambda-Free Higher-Order Terms*, in "CADE-26 - 26th International Conference on Automated Deduction", Gothenburg, Sweden, L. DE MOURA (editor), Lecture Notes in Computer Science, Springer, August 2017, vol. 10395, pp. 432-453 [*DOI :* 10.1007/978-3-319-63046-5_27], https://hal.inria.fr/hal-01592186

[22] A. BENTKAMP, J. C. BLANCHETTE, D. KLAKOW. *A Formal Proof of the Expressiveness of Deep Learning*, in "ITP 2017: 8th International Conference on Interactive Theorem Proving", Brasilia, Brazil, September 2017 [*DOI :* 10.1007/3-540-48256-3_12], https://hal.inria.fr/hal-01599172

[23] J. C. BLANCHETTE, A. BOUZY, A. LOCHBIHLER, A. POPESCU, D. TRAYTEL. *Friends with Benefits: Implementing Corecursion in Foundational Proof Assistants*, in "Programming Languages and Systems - 26th European Symposium on Programming, ESOP 2017", Uppsala, Sweden, April 2017, https://hal.inria.fr/hal-01599167

[24] J. C. BLANCHETTE, M. FLEURY, D. TRAYTEL. *Nested Multisets, Hereditary Multisets, and Syntactic Ordinals in Isabelle/HOL*, in "FSCD 2017: 2nd International Conference on Formal Structures for Computation and Deduction", Oxford, United Kingdom, September 2017, vol. 11, pp. 1 - 11 [*DOI :* 10.4230/LIPIcs.FSCD.2017.11], https://hal.inria.fr/hal-01599176

[25] J. C. BLANCHETTE, M. FLEURY, C. WEIDENBACH. *A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality*, in "26th International Joint Conference on Artificial Intelligence", Melbourne, Australia, C. SIERRA (editor), August 2017, pp. 4786-4790 [*DOI :* 10.24963/ijcai.2017/667], https://hal.inria.fr/hal-01592164

[26] J. C. BLANCHETTE, F. MEIER, A. POPESCU, D. TRAYTEL. *Foundational nonuniform (Co)datatypes for higher-order logic*, in "LICS 2017: 32nd Annual ACM/IEEE Symposium on Logic in Computer Science",

Reykjavik, Iceland, June 2017, pp. 1 - 12 [*DOI :* 10.1109/LICS.2017.8005071], https://hal.inria.fr/hal-01599174

[27] J. C. BLANCHETTE, U. WALDMANN, D. WAND. *A Lambda-Free Higher-Order Recursive Path Order*, in "Foundations of Software Science and Computation Structures, 20th International Conference (FOSSACS 2017)", Uppsala, Sweden, J. ESPARZA, A. S. MURAWSKI (editors), Lecture Notes in Computer Science, Springer, April 2017, vol. 10203, pp. 461-479 [*DOI :* 10.1007/978-3-662-54458-7_27], https://hal.inria.fr/hal-01592189

[28] R. BRADFORD, J. H. DAVENPORT, M. ENGLAND, H. ERRAMI, V. GERDT, D. GRIGORIEV, C. HOYT, M. KOŠTA, O. RADULESCU, T. STURM, A. WEBER. *A Case Study on the Parametric Occurrence of Multiple Steady States*, in "ISSAC 2017 - International Symposium on Symbolic and Algebraic Computation", Kaiserslautern, Germany, ACM, July 2017, pp. 45-52 [*DOI :* 10.1145/3087604.3087622], https://hal.inria.fr/hal-01648694

[29] S. CRUANES. *Satisfiability Modulo Bounded Checking*, in "International Conference on Automated Deduction (CADE)", Gothenburg, Sweden, Leonardo de Moura, August 2017, vol. 26, pp. 114-129 [*DOI :* 10.1007/978-3-319-63046-5_8], https://hal.inria.fr/hal-01572531

[30] M. DUROEULX, N. BRINZEI, M. DUFLOT, S. MERZ. *Satisfiability techniques for computing minimal tie sets in reliability assessment*, in "10th International Conference on Mathematical Methods in Reliability, MMR 2017", Grenoble, France, July 2017, pp. 1-8, https://hal.inria.fr/hal-01630851

[31] M. ENGLAND, H. ERRAMI, D. GRIGORIEV, O. RADULESCU, T. STURM, A. WEBER. *Symbolic Versus Numerical Computation and Visualization of Parameter Regions for Multistationarity of Biological Networks*, in "CASC 2017 - 19th International Workshop on Computer Algebra in Scientific Computing", Beijing, China, V. P. GERDT, W. KOEPF, W. M. SEILER, E. V. VOROZHTSOV (editors), LNCS - Lecture Notes in Computer Science, Springer, September 2017, vol. 10490 [*DOI :* 10.1007/978-3-319-66320-3], https://hal.inria.fr/hal-01648691

[32] F. FAKHFAKH, M. TOUNSI, M. MOSBAH, A. HADJ KACEM, D. MÉRY. *A Formal Approach for Maintaining Forest Topologies in Dynamic Networks*, in "ICIS 2017 - 16th IEEE/ACIS International Conference on Computer and Information Science", Wuhan, China, Studies in Computational Intelligence, May 2017, vol. 719, pp. 123-137 [*DOI :* 10.1007/978-3-319-60170-0_9], https://hal.archives-ouvertes.fr/hal-01495807

[33] P. FONTAINE, M. OGAWA, T. STURM, X. VU. *Subtropical Satisfiability*, in "FroCoS 2017 - 11th International Symposium on Frontiers of Combining Systems", Brasilia, Brazil, C. DIXON, M. FINGER (editors), Lecture Notes in Artificial Intelligence, Springer, September 2017, vol. 10483 [*DOI :* 10.1007/978-3-319-66167-4], https://hal.inria.fr/hal-01590899

[34] P. J. GIBSON, S. KHERROUBI, D. MÉRY. *Applying a Dependency Mechanism for Voting Protocol Models Using Event-B*, in "37th International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE 2017)", Neuchâtel, Switzerland, A. BOUAJJANI, A. SILVA (editors), Formal Techniques for Distributed Objects, Components, and Systems, Springer International Publishing, June 2017, vol. LNCS-10321, pp. 124-138 [*DOI :* 10.1007/978-3-319-60225-7_9], https://hal.inria.fr/hal-01658423

[35] S. HIMMICHE, A. AUBRY, P. MARANGÉ, J.-F. PÉTIN, M. DUFLOT. *Using statistical-model-checking-based simulation for evaluating the robustness of a production schedule*, in "7th Workshop on Service Orientation

in Holonic and Multi-Agent Manufacturing, SOHOMA'17", Nantes, France, October 2017, https://hal.inria.fr/hal-01652140

[36] M. HORBACH, M. VOIGT, C. WEIDENBACH. *On the Combination of the Bernays–Schönfinkel–Ramsey Fragment with Simple Linear Integer Arithmetic*, in "CADE 26 - 26th International Conference on Automated Deduction", Gothenburg, Sweden, L. DE MOURA (editor), Lecture Notes in Computer Science, Springer, August 2017, vol. 10395, pp. 77-94 [*DOI :* 10.1007/978-3-319-63046-5_6], https://hal.inria.fr/hal-01592160

[37] S. KHERROUBI, D. MÉRY. *Contextualization and Dependency in State-Based Modelling - Application to Event-B*, in "MEDI 2017 - International Conference on Model and Data Engineering", Barcelona, Spain, Lecture Notes in Computer Science, Springer, October 2017, vol. 10563, pp. 137–152 [*DOI :* 10.1007/978-3-319-66854-3_11], https://hal.inria.fr/hal-01631017

[38] N. SCHNEPF, R. BADONNEL, A. LAHMADI, S. MERZ. *Automated Verification of Security Chains in Software-Defined Networks with Synaptic*, in "NetSoft 2017 - IEEE Conference on Network Softwarization", Bologna, Italy, IEEE Computer Society, July 2017, 9 p. [*DOI :* 10.1109/NETSOFT.2017.8004195], https://hal.inria.fr/hal-01630806

[39] A. TEUCKE, C. WEIDENBACH. *Decidability of the Monadic Shallow Linear First-Order Fragment with Straight Dismatching Constraints*, in "CADE 2017 - 26th International Conference on Automated Deduction", Gothenburg, Sweden, L. DE MOURA (editor), CADE 2017: Automated Deduction – CADE 26, Springer, August 2017, vol. 10395, pp. 202-219 [*DOI :* 10.1007/978-3-319-63046-5_13], https://hal.inria.fr/hal-01657026

[40] M. VOIGT. *A fine-grained hierarchy of hard problems in the separated fragment*, in "LICS 2017 - 32nd Annual ACM/IEEE Symposium on Logic in Computer Science", Reykjavik, Iceland, J. OUAKNINE (editor), IEEE Computer Society, June 2017, pp. 1 - 12 [*DOI :* 10.1109/LICS.2017.8005094], https://hal.inria.fr/hal-01592172

[41] M. VOIGT. *The Bernays–Schönfinkel–Ramsey Fragment with Bounded Difference Constraints over the Reals Is Decidable*, in "FroCoS 2017 - 11th International Symposium on Frontiers of Combining Systems", Brasilia, Brazil, C. DIXON, M. FINGER (editors), Lecture Notes in Computer Science, Springer, September 2017, vol. 10483, pp. 244-261 [*DOI :* 10.1007/978-3-319-66167-4_14], https://hal.inria.fr/hal-01592169

Conferences without Proceedings

[42] S. HIMMICHE, P. MARANGÉ, A. AUBRY, M. DUFLOT, J.-F. PÉTIN. *Evaluation de la robustesse d'un ordonnancement par Automates Temporisés Stochastiques*, in "11ème Colloque sur la Modélisation des Systèmes Réactifs, MSR 2017", Marseille, France, November 2017, https://hal.inria.fr/hal-01652138

Books or Proceedings Editing

[43] C. DUBOIS, P. MASCI, D. MÉRY (editors). *Proceedings of the Third Workshop on Formal Integrated Development Environment, F-IDE@FM 2016, Limassol, Cyprus, November 8, 2016*, EPTCS, Cyprus, January 2017, vol. 240 [*DOI :* 10.4204/EPTCS.240], https://hal.inria.fr/hal-01652413

Research Reports

[44] H. BARBOSA, J. C. BLANCHETTE, P. FONTAINE. *Scalable Fine-Grained Proofs for Formula Processing*, Universite de Lorraine, CNRS, Inria, LORIA, Nancy, France ; Universidade Federal do Rio Grande do

Norte, Natal, Brazil ; Vrije Universiteit Amsterdam, Amsterdam, The Netherlands ; Max-Planck-Institut für Informatik, Saarbrücken, Germany, May 2017, 25 p. , https://hal.inria.fr/hal-01526841

[45] H. BARBOSA, P. FONTAINE, A. REYNOLDS. *Congruence Closure with Free Variables*, Inria, Loria, Universite de Lorraine, UFRN, University of Iowa, January 2017, https://hal.inria.fr/hal-01442691

[46] L. LAMPORT, S. MERZ. *Auxiliary Variables in TLA+*, Inria Nancy - Grand Est (Villers-lès-Nancy, France) ; Microsoft Research, May 2017, https://arxiv.org/abs/1703.05121 , https://hal.inria.fr/hal-01488617

Other Publications

[47] M. DUROEULX, N. BRINZEI, M. DUFLOT, S. MERZ. *Satisfiability techniques for computing minimal tie sets in reliability assessment*, April 2017, working paper or preprint, https://hal.inria.fr/hal-01518920

[48] M. HORBACH, M. VOIGT, C. WEIDENBACH. *The Universal Fragment of Presburger Arithmetic with Unary Uninterpreted Predicates is Undecidable*, September 2017, working paper or preprint, https://hal.inria.fr/hal-01592177

References in notes

[49] J.-R. ABRIAL. *Modeling in Event-B: System and Software Engineering*, Cambridge University Press, 2010

[50] Y. AIT AMEUR, D. MÉRY. *Making explicit domain knowledge in formal system development*, in "Science of Computer Programming", March 2016, vol. 121, pp. 100-127 [*DOI :* 10.1016/J.SCICO.2015.12.004], https://hal.inria.fr/hal-01245832

[51] N. AZMY. *A Machine-Checked Proof of Correctness of Pastry*, Saarland University and University of Lorraine, Saarbrücken, Germany, and Nancy, France, 2016

[52] L. BACHMAIR, H. GANZINGER. *Rewrite-Based Equational Theorem Proving with Selection and Simplification*, in "Journal of Logic and Computation", 1994, vol. 4, n^{o} 3, pp. 217–247

[53] R. BACK, J. VON WRIGHT. *Refinement calculus—A systematic introduction*, Springer Verlag, 1998

[54] C. BARRETT, R. SEBASTIANI, S. A. SESHIA, C. TINELLI. *Satisfiability Modulo Theories*, in "Handbook of Satisfiability", A. BIERE, MARIJN J. H. HEULE, H. VAN MAAREN, T. WALSH (editors), Frontiers in Artificial Intelligence and Applications, IOS Press, February 2009, vol. 185, chap. 26, pp. 825-885

[55] P. CHOCRON, P. FONTAINE, C. RINGEISSEN. *A Polite Non-Disjoint Combination Method: Theories with Bridging Functions Revisited*, in "25th International Conference on Automated Deduction, CADE-25", Berlin, Germany, A. P. FELTY, A. MIDDELDORP (editors), Lecture Notes in Computer Science, Springer, August 2015, vol. 9195, pp. 419-433 [*DOI :* 10.1007/978-3-319-21401-6_29], https://hal.inria.fr/hal-01157898

[56] P. CHOCRON, P. FONTAINE, C. RINGEISSEN. *A Rewriting Approach to the Combination of Data Structures with Bridging Theories*, in "Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015", Wroclaw, Poland, C. LUTZ, S. RANISE (editors), Lecture Notes in Computer Science, Springer, September 2015, vol. 9322, pp. 275–290 [*DOI :* 10.1007/978-3-319-24246-0_17], https://hal.inria.fr/hal-01206187

[57] S. CRUANES, J. C. BLANCHETTE. *Extending Nunchaku to Dependent Type Theory*, in "Hammers for Type Theories (HaTT 2016)", Coimbra, Portugal, EPTCS, July 2016, vol. 210, pp. 3-12 [*DOI :* 10.4204/EPTCS.210.3], https://hal.inria.fr/hal-01401696

[58] N. FOSTER, A. GUHA, M. REITBLATT, A. STORY, M. J. FREEDMAN, N. PRAVEEN KATTA, C. MONSANTO, J. REICH, J. REXFORD, C. SCHLESINGER, D. WALKER, R. HARRISON. *Languages for software-defined networks*, in "IEEE Communications Magazine", 2013, vol. 51, n^{o} 2, pp. 128-134

[59] L. LAMPORT. *Specifying Systems*, Addison-Wesley, Boston, Mass., 2002

[60] C. MORGAN. *Programming from Specifications*, Prentice Hall, 1998, 2nd edition

[61] D. MÉRY, S. RUSHIKESH, A. TARASYUK. *Integrating Domain-Based Features into Event-B: a Nose Gear Velocity Case Study*, in "Model and Data Engineering - 5th International Conference, MEDI 2015", Rhodos, Greece, L. BELLATRECHE, Y. MANOLOPOULOS (editors), LNCS, Springer, 2015, vol. 9344, pp. 89-102, https://hal.inria.fr/hal-01245991

[62] A. REYNOLDS, J. C. BLANCHETTE, S. CRUANES, C. TINELLI. *Model Finding for Recursive Functions in SMT*, in "8th International Joint Conference on Automated Reasoning (IJCAR 2016)", Coimbra, Portugal, June 2016 [*DOI :* 10.1007/978-3-319-40229-1_10], https://hal.inria.fr/hal-01336082

[63] A. ROWSTRON, P. DRUSCHEL. *Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems*, in "IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)", Heidelberg, Germany, R. GUERRAOUI (editor), Lecture Notes in Computer Science, Springer, 2001, vol. 2218, pp. 329-350

[64] V. SOFRONIE-STOKKERMANS. *Hierarchical and modular reasoning in complex theories: The case of local theory extensions*, in "Frontiers of Combining Systems. 6th International Symposium FroCos 2007, Proceedings", Liverpool, UK, B. KONEV, F. WOLTER (editors), Lecture Notes in Computer Science, Springer, 2007, vol. 4720, pp. 47-71, Invited paper

[65] T. STURM, M. VOIGT, C. WEIDENBACH. *Deciding First-Order Satisfiability when Universal and Existential Variables are Separated*, in "LICS 2016", New York, United States, July 2016, pp. 86 - 95 [*DOI :* 10.1145/2933575.2934532], https://hal.inria.fr/hal-01389744

[66] S. A. WEIL, S. A. BRANDT, E. L. MILLER, D. D. E. LONG, C. MALTZAHN. *Ceph: A Scalable, High-Performance Distributed File System*, in "7th Symp. Operating Systems Design and Implementation (OSDI '06)", Seattle, WA, Usenix Association, 2006, pp. 307-320