IN PARTNERSHIP WITH:
CNRS

Max-Planck-Institut für Informatik Saarbrücken

Université de Lorraine

Activity Report 2018

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

VeriDis is located both at the Inria research center in Nancy and at the Max-Planck Institute for Informatics in Saarbrücken, Germany.

Keywords:

Computer Science and Digital Science:

A2.1.7. - Distributed programming
A2.1.11. - Proof languages
A2.4. - Formal method for verification, reliability, certification
A2.4.1. - Analysis
A2.4.2. - Model-checking
A2.4.3. - Proofs
A2.5. - Software engineering
A7.2. - Logic in Computer Science
A8.4. - Computer Algebra

Other Research Topics and Application Domains:

B6.1. - Software industry
B6.1.1. - Software engineering
B6.3.2. - Network protocols
B6.6. - Embedded systems

1. Team, Visitors, External Collaborators

Research Scientists

Stephan Merz [Team leader, Inria, Senior Researcher, HDR]
Igor Konnov [Inria, Researcher, from March 2018]
Thomas Sturm [CNRS, Senior Researcher, HDR]
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, Inria secondment]
Dominique Méry [Univ. de Lorraine, Professor]
Sorin Stratulat [Univ. de Lorraine, Associate Professor, Inria secondment from September 2018]

Post-Doctoral Fellows

Yann Duplouy [Inria, from December 2018]
Sophie Tourret [Max-Planck Institut für Informatik]

PhD Students

Martin Bromberger [Univ. des Saarlandes]
Margaux Duroeulx [Univ. de Lorraine]
Daniel El Ouraoui [Inria]
Alberto Fiori [Univ. des Saarlandes, from September 2018]
Mathias Fleury [Univ. des Saarlandes]
Alexis Grall [Univ. de Lorraine, from October 2018]

Souad Kherroubi [Univ. de Lorraine]
Pierre Lermusiaux [Univ. de Lorraine]
Nicolas Schnepf [Inria, shared with Resist team]
Hans-Jörg Schurr [Inria]
Andreas Teucke [Univ. des Saarlandes, until May 2018]
Marco Voigt [Univ. des Saarlandes]

Technical staff

Stéphane Glondu [Inria, seconded from SED Inria Nancy for one person-month]

Interns

Alexis Grall [Univ. de Lorraine, from March until August 2018]
Axel Palaude [Inria, from May until July 2018]

Administrative Assistants

Anne Chretien [Univ. de Lorraine, from April until September 2018]
Sophie Drouot [Inria]
Sylvie Hilbert [CNRS, from July 2018]
Jennifer Müller [Max-Planck Institut für Informatik]

Visiting Scientist

Cezary Kaliszyk [Univ. de Lorraine, from May until June 2018]

External Collaborator

Jasmin Christian Blanchette [Vrije Universiteit Amsterdam]

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 as early as 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 the SPASS [10] workbench. It currently consists of one of the leading automated theorem provers for first-order logic based on the superposition calculus [51] and a theory solver for linear arithmetic.

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. 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 difficult or impossible to express 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.

[1] Satisfiability Modulo Theories [54]

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$^+$ [67] 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.

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], [52], [68] 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 networks 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 were mostly 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

Marie Duflot-Kremer received the Serge Hocquenghem prize awarded biannually by *Association pour l'Innovation Didactique* for her contributions to the popularization of computer science and in particular her work on developing and promoting unplugged computer science activities.

Thomas Sturm was a plenary invited speaker at ISSAC 2018, the leading conference in Symbolic Computation.

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: Parts of the Redlog code are 25 years old now. Version 1 of the underlying computer algebra system Reduce has been published even 50 years ago. In 2018 we therefore decided to go for major revisions and improvements of Redlog's software architecture.

Redlog is, as well as the underlying Reduce, implemented in a language called RLISP, which technically parses an Algol-style procedural notation into a quite minimalistic Lisp 1 dialect called Standard Lisp. RLISP and Reduce and, subsequently, Redlog are bootstrapped on the basis of an existing Standard Lisp. Today, there are two independent implementations of Standard Lisp left, which are supported only on the basis of private commitment of essentially one individual per Lisp. With the large code base of Redlog plus the necessary algebraic algorithms from Reduce a migration to a different language or computer algebra system is not feasible. We are therefore experimenting with the realization of a Standard Lisp on the basis of ANSI Common Lisp, which could allow an RLISP-Reduce-Redlog bootstrap. Given that Common Lisp is a Lisp 2, this is feasible but not at all straightforward. Also, it naturally comes with a loss of efficiency, which requires careful programming. We are grateful that Inria supports this project with an engineer's position for limited time (ADT-135 Fast Track).

We are furthermore working on an improved design of Redlog's black-box and service schedulers [Dolzmann and Sturm, ACM SIGSAM Bull. 31, 1997] and a revision of global Boolean switches in favor of named optional arguments. In that course we store related information more explicitly, which will allow automatic consistency checks of the schedulers and automated interface generation with named arguments. In addition, this supports an interactive help system inside Reduce, which is also under construction.

- 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.

SPASS 3.9 has been used as the basis for SPASS-AR, a 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.
NEWS OF THE YEAR: We 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.

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

6.3. SPASS-SATT

KEYWORDS: Automated deduction - Decision
SCIENTIFIC DESCRIPTION: SPASS -SATT is an SMT solver for the theories of linear integer arithmetic, linear rational arithmetic and mixed linear arithmetic. It features new tests for the satisfiability of unbounded systems, as well as new algorithms for the detection of integer solutions.

We further investigated the use of redundancy 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.
FUNCTIONAL DESCRIPTION: SPASS-SATT is an SMT solver for linear integer arithmetic, mixed linear arithmetic and rational linear arithmetic.
NEWS OF THE YEAR: The first version of SPASS-SATT was released in June 2018. It participated in SMTCOMP-2018 in the quantifier free integer and rational linear arithmetic categories. In both categories it solved more problems in a shorter period of time than any other SMT solver. With respect to the weighted bucket ranking it scored first in the linear integer category and second in the linear rational category.

- Participants: Martin Bromberger, Mathias Fleury and Christoph Weidenbach
- Contact: Martin Bromberger
- URL: https://www.mpi-inf.mpg.de/departments/automation-of-logic/software/spass-workbench/spass-satt/

6.4. SPIKE

KEYWORDS: Proof - Automated deduction - Automated theorem proving - Term Rewriting Systems - Formal methods

SCIENTIFIC DESCRIPTION: SPIKE, an automatic induction-based theorem prover built to reason on conditional theories with equality, is one of the few formal tools able to perform automatically mutual and lazy induction. Designed in the 1990s, it has been successfully used in many non-trivial applications and served as a prototype for different proof experiments and extensions.
FUNCTIONAL DESCRIPTION: Automated induction-based theorem prover

RELEASE FUNCTIONAL DESCRIPTION: Proof certification with Coq, cyclic induction, decision procedures

- Participant: Sorin Stratulat
- Contact: Sorin Stratulat

6.5. 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 2018 have been focused on non-linear arithmetic reasoning, quantifier handling and proof production.

The veriT solver participated in the SMT competition SMT-COMP 2018 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.

veriT is also a prototype platform for ideas developed within the Matryoshka project, aiming at greater availability of automated reasoning for proof assistants.

- 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

6.6. 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 a preliminary integration of Nunchaku in the Lean proof assistant. This work was performed by Pablo Le Hénaff during an internship at Vrije Universiteit Amsterdam. See his internship report at http://matryoshka.gforge.inria.fr/pubs/lehenaff_report.pdf for details.

- Participants: Jasmin Christian Blanchette and Simon Cruanes
- Partner: Vrije Universiteit Amsterdam
- Contact: Jasmin Christian Blanchette
- Publications: Extending Nunchaku to Dependent Type Theory - Model Finding for Recursive Functions in SMT
- URL: https://github.com/nunchaku-inria

6.7. 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: Ioannis Filippidis joined the development team in November 2018 and started designing and implementing support for reasoning about TLA+'s ENABLED construct.

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

6.8. Apalache

Abstraction-based Parameterized TLA+ Checker

KEYWORD: Model Checker

FUNCTIONAL DESCRIPTION: The first version implements a symbolic bounded model checker for TLA$^+$ that runs under the same assumptions as the explicit-state model checker TLC. It checks whether a TLA$^+$ specification satisfies an invariant candidate by checking satisfiability of an SMT formula that encodes: (1) an execution of bounded length, and (2) preservation of the invariant candidate in every state of the execution. Our tool is still in the experimental phase, due to a number of challenges posed by the semantics of TLA$^+$ to SMT solvers.

- Partner: Technische Universität Wien
- Contact: Igor Konnov
- Publications: BmcMT: Bounded Model Checking of TLA + Specifications with SMT - Extracting Symbolic Transitions from $TLA+$ Specifications
- URL: https://forsyte.at/research/apalache/

6.9. ByMC

Byzantine Model Checker

KEYWORDS: Model Checker - Distributed computing - Verification

SCIENTIFIC DESCRIPTION: In recent work, we have introduced a series of techniques for automatic verification of threshold-guarded distributed algorithms that have the following features: (1) up to t of n processes may exhibit crash or Byzantine failures, (2) the correct processes count messages and progress when they receive sufficiently many messages, e.g., at least $t+1$, (3) the number n of processes in the system is a parameter, as well as t, (4) and the parameters are restricted by a resilience condition, e.g., $n > 3t$.

FUNCTIONAL DESCRIPTION: ByMC implements several techniques for the parameterized verification of threshold-guarded distributed algorithms such as reliable broadcast, one-step Byzantine consensus, non-blocking atomic commit, condition-based consensus, and randomized consensus. The tool accepts two kinds of inputs: (i) threshold automata (the framework of our verification techniques) and (ii) Parametric Promela (which is similar to the way in which the distributed algorithms are presented in the distributed computing literature). Internally, the tool analyzes representative executions by querying an SMT solver. Apart from verification, ByMC also implements a technique for the automatic synthesis of threshold guards.

The tool can run on a single computer as well as in an MPI cluster, e.g., Grid5000 or Vienna Scientific Cluster.

NEWS OF THE YEAR: We have introduced a parallel extension of the tool, which allows one to run verification experiments in an MPI cluster. The parallel version of the tool demonstrated a good speed-up on large benchmarks run in Vienna Scientific Cluster and Grid 5000.

- Partner: Technische Universität Wien
- Contact: Igor Konnov
- Publications: ByMC: Byzantine Model Checker - Reachability in Parameterized Systems: All Flavors of Threshold Automata - Model Checking of Fault-Tolerant Distributed Algorithms: from Classics towards Contemporary - Verification of Randomized Distributed Algorithms under Round-Rigid Adversaries
- URL: https://forsyte.at/software/bymc/

7. New Results

7.1. Automated and Interactive Theorem Proving

Participants: Jasmin Christian Blanchette, Martin Bromberger, Daniel El Ouraoui, Mathias Fleury, Pascal Fontaine, Stephan Merz, Hans-Jörg Schurr, Sorin Stratulat, Thomas Sturm, Andreas Teucke, Sophie Tourret, Marco Voigt, Uwe Waldmann, Christoph Weidenbach.

7.1.1. Extension of the Superposition Calculus with λ-free Higher-Order Terms and (Co)datatypes

Joint work with Alexander Bentkamp (VU Amsterdam), Simon Cruanes (Aesthetic Integration), Nicolas Peltier (IMAG Grenoble), and Simon Robillard (Chalmers Gothenburg).

Superposition is a highly successful calculus for reasoning about first-order logic with equality. As a stepping stone towards extending the calculus to full higher-order logic, Bentkamp et al. [19] designed a graceful generalization of the calculus to a fragment devoid of λ-abstractions, but with partial application and application of variables, two crucial higher-order features. This builds on the work on term orders, namely the recursive path order [57] and the Knuth-Bendix order [55]. We implemented the calculi in Simon Cruanes's Zipperposition prover and evaluated them on TPTP benchmarks. The performance is substantially better than with the traditional, encoding-based approach. The new superposition-like calculus serves as a stepping stone towards complete, efficient automatic theorem provers for full higher-order logic.

Another extension of superposition, by Blanchette et al. [21], concerns the native support for inductive and coinductive datatypes. The ability to reason about datatypes has many applications in program verification, formalization of the metatheory of programming languages, and even formalization of mathematics.

Both lines of work aim at bridging the gap between automatic and interactive theorem provers, by increasing the expressiveness and efficiency of best-of-breed automatic first-order provers based on the superposition calculus.

7.1.2. IsaFoL: Isabelle Formalization of Logic

Joint work with Alexander Bentkamp (VU Amsterdam), Andreas Halkjær From (DTU Copenhagen), Alexander Birch Jensen (DTU Copenhagen), Peter Lammich (TU München), John Bruntse Larsen (DTU Copenhagen), Julius Michaelis (TU München), Tobias Nipkow (TU München), Nicolas Peltier (IMAG Grenoble), Simon Robillard (Chalmers Gothenburg), Anders Schlichtkrull (DTU Copenhagen), Dmitriy Traytel (ETH Zürich), Jørgen Villadsen (DTU Copenhagen), and Petar Vukmirović (VU Amsterdam).

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.

The main result this year has been a formalization of a large part of Bachmair and Ganzinger's chapter on resolution theorem proving in the *Handbook of Automated Reasoning*, by Anders Schlichtkrull et al. The work was conducted by Schlichtkrull largely during a visit at the MPI in Saarbrücken and was published at IJCAR 2018 [34]. The following quote of one of the reviews nicely sums up the objective of the project:

> The authors convinced me that their development is a great tool for exploring/developing calculus extensions. It will enable us to "*extend/hack without fear*."

A follow-up paper [33], also by Schlichtkrull et al., has been accepted at CPP 2019. In this work, a chain of refinement leads to a verified executable prover.

The IsaFoL repository has welcome several further additions in 2018, and there is largely finished work, which we expect will lead to at least two publications in 2019:

- After the journal publication [13] following up on an IJCAR 2016 paper and a publication at CPP 2018 [23], Fleury has improved his verified SAT solver IsaSAT further by implementing four optimizations: restarts, forgetting, blocking literals, and machine integers. IsaSAT is now by far the most efficient verified SAT solver, and it is catching up with MiniSat, a reference (but unverified) SAT solver implementation.
- Sophie Tourret and Simon Robillard have formalized a new framework, designed primarily by Uwe Waldmann, that captures abstractly the lifting from completeness of a calculus for propositional logic to a first-order prover. This will yield a simpler proof of Bachmair and Ganzinger's completeness theorem and will be reusable for reasoning about other provers (e.g., superposition provers), whether with pen and paper or in Isabelle.

Jasmin Blanchette briefly describes this ongoing research in an invited paper [20], which he will present at CPP 2019.

7.1.3. Subtropical Reasoning for Real Inequalities

Joint work with Hoon Hong (North Carolina State University, Raleigh, NC).

We consider systems of strict multivariate polynomial inequalities over the reals. All polynomial coefficients are parameters ranging over the reals, where for each coefficient we prescribe its sign. We are interested in the existence of positive real solutions of our system for all choices of coefficients subject to our sign conditions. We give a decision procedure for the existence of such solutions. In the positive case our procedure yields a parametric positive solution as a rational function in the coefficients. Our framework allows heuristic subtropical approaches to be reformulated for non-parametric systems of polynomial inequalities. Such systems have been recently used in qualitative biological network analysis and, independently, in satisfiability modulo theory solving. We apply our results to characterize the incompleteness of those methods.

The approach allows SMT solving for non-linear real arithmetic to be heuristically reduced to linear real arithmetic, to which, e.g., methods from 7.1.4 are applicable. In the special case of single inequalities one can even reduce to linear programming. [25]. This has been successfully applied to heuristic search for Hopf bifurcation fixed points in chemical and biological network analysis.

7.1.4. Reasoning in Linear Arithmetic

We have continued our work on reasoning in linear integer (LIA), linear real (LRA) and linear mixed arithmetic (LIRA). Whereas the standard branch-and-bound techniques [63] for LIA typically work well for bounded systems of inequations, they often diverge on unbounded systems. We already proposed cube techniques for this case. They comprise efficiently computable sufficient tests for the existence of a solution [58]. However, these tests are only necessary for the existence of a solution in the case of a system that is

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

unbounded in all directions. For the case of partially unbounded systems, our combination of the Mixed-Echelon-Hermite transformation and the Double-Bounded Reduction for systems of linear mixed arithmetic preserve satisfiability, can be computed in polynomial time, and turn any LIRA system into a bounded system [22]. Existing approaches for LIRA, e.g., branch-and-bound and cuts from proofs, only explore a finite search space after the application of our two transformations. The transformations orient themselves on the structure of an input system instead of computing *a priori* (over-)approximations out of the available constants. We also developed a polynomial method for converting certificates of (un)satisfiability from the transformed to the original system.

Meanwhile our techniques have been integrated into the SMT solver veriT, but also in other SMT solvers such as Z3 [72] or MathSAT [62]. They have been substantial for our success at SMTComp2018.

7.1.5. Combination of Satisfiability Procedures

Joint work with Christophe Ringeissen (Inria Nancy – Grand Est, Pesto) and Paula Chocron (IIIA-CSIC, Bellaterra, 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. 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 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 [60]. 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 [61] 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 2018, we have been improving the framework and unified both results. A paper is under review.

7.1.6. Quantifier Handling in SMT

Joint work with Andrew J. Reynolds (Univ. of Iowa, USA) and Cezary Kaliszyk (Univ. of Innsbruck).

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 in 2017 [53]. In a publication at TACAS 2018 [31], we revisit enumerative instantiation for SMT.

We are currently investigating machine learning techniques as a tool for filtering instantiations. Other ongoing work aims at lifting the above techniques to higher-order reasoning.

7.1.7. Real Quantifier Elimination, Decision, and Satisfiability and Their Applications

Effective quantifier elimination procedures for first-order theories provide a powerful tool for generically solving a wide range of problems based on logical specifications. In contrast to general first-order provers, quantifier elimination procedures are based on a fixed set of admissible logical symbols with an implicitly fixed semantics. This admits the use of sub-algorithms from symbolic computation. Specifically quantifier elimination for the reals has been successfully applied in geometry, verification, and the life sciences.

A survey paper with an invited talk at ISSAC 2018 provides a coherent view on the scientific developments of the virtual substitution method for real quantifier elimination during the past three decades [17]. Another recent survey paper had illustrated relevant applications of that method [71].

7.1.8. Non-Linear Arithmetic in SMT

Joint work with M. Ogawa and X. T. Vu (Japan Advanced Institute of Science and Technology), V. K. To (University of Engineering and Technology, VNU, Hanoi, Vietnam).

In the context of the SC^2 project (cf. sections 8.1 and 8.3), we study the theory, design techniques, and implement software to push forward the non-linear arithmetic (NLA) reasoning capabilities in SMT. Previously, 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. We also studied integration of these procedures into combinations of theories. These ideas were validated through an implementation within the veriT solver, together with code from the raSAT solver (from JAIST), and they were presented at the SC^2 workshop 2018 [24].

7.1.9. Proofs for SMT

We have previously 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 conference publication in 2017. In 2018, we polished the approach, fully implementing proof reconstruction of veriT proofs in Isabelle. A paper has been accepted in the Journal of Automated Reasoning.

7.1.10. A More Efficient Technique for Validating Cyclic Pre-Proofs

Cyclic pre-proofs can be represented as sets of finite tree derivations with back-links. In a setting of first-order logic with inductive definitions, the nodes of the tree derivations are labelled by sequents and the back-links connect particular terminal nodes, referred to as buds, to other nodes labelled by the same sequent. However, only some back-links can constitute sound pre-proofs. Previously, it was shown that special ordering and derivability conditions, defined along the minimal cycles of the digraph representing a particular normal form of the cyclic pre-proof, are sufficient for validating the back-links. In that approach, a single constraint could be checked several times when processing different minimal cycles, hence one may require additional recording mechanisms to avoid redundant computation in order to achieve polynomial time complexity.

In [39], we presented a new approach that does not need to process minimal cycles. It is based on a normal form in which the validation conditions are defined by taking into account only the root-bud paths from the non-singleton strongly connected components of its digraph.

7.1.11. Mechanical Synthesis of Algorithms by Logical and Combinatorial Techniques

Joint work with Isabela Dramnesc (West University, Timisoara, Romania) and Tudor Jebelean (RISC, Johannes Kepler University, Linz, Austria).

In [14], we developed logical and combinatorial methods for automating the generation of sorting algorithms for binary trees, starting from input-output specifications and producing conditional rewrite rules. The main approach consists in proving (constructively) the existence of an appropriate output from every input. The proof may fail if some necessary sub-algorithms are lacking. Then, their specifications are suggested and their synthesis is performed by the same principles.

The main goal is to avoid the possibly prohibitive cost of pure resolution proofs by using a natural-style proving in which domain-specific strategies and inference steps lead to a significant increase of efficiency. We introduce novel techniques and combine them with classical techniques for natural-deduction style proving, as well as methods based on the properties of domain-specific relations and functions. In particular, we use combinatorial techniques in order to generate possible witnesses, which in certain cases lead to the discovery of new induction principles. From the proof, the algorithm is extracted by transforming inductive proof steps into recursions, and case-based proof steps into conditionals.

The approach was demonstrated using the Theorema system for developing the theory, implementing the prover, and performing the proofs of the necessary properties and synthesis conjectures. It was also validated in the Coq system, allowing us to compare the facilities of the two systems in view of our application.

7.1.12. Formal Proofs of Tarjan's Algorithm

Joint work with Ran Chen (Chinese Academy of Sciences), Cyril Cohen and Laurent Théry (Inria Sophia Antipolis Méditerranée, Marelle), and Jean-Jacques Lévy (Inria Paris, Pi.r2).

We compare formal proofs of Tarjan's algorithm for computing strongly connected components in a graph in three different proof assistants: Coq, Isabelle/HOL, and Why3. Our proofs are based on a representation of the algorithm as a functional program (rather than its more conventional imperative representation), which was verified in Why3 by Chen and Lévy [59]. The proofs in all three assistants are thus closely comparable and in particular employ the same invariants. This lets us focus on different formalizations due to idiosyncracies of the proof assistants, such as w.r.t. handling mutually recursive function definitions whose termination is not obvious according to syntactic criteria, and compare the degree of automation in the three assistants. A report is available on arXiv [45].

7.2. Formal Methods for Developing and Analyzing Algorithms and Systems

Participants: Marie Duflot-Kremer, Yann Duplouy, Margaux Duroeulx, Souad Kherroubi, Igor Konnov, Dominique Méry, Stephan Merz, Axel Palaude, Nicolas Schnepf, Christoph Weidenbach.

7.2.1. Parameterized Verification of Threshold-Guarded Fault-Tolerant Distributed Algorithms

Joint work with Nathalie Bertrand (Inria Rennes, SUMO project team) and Jure Kukovec, Marijana Lazić, Ilina Stoilkovska, Josef Widder, Florian Zuleger (TU Wien).

Many fault-tolerant distributed algorithms use threshold guards: processes broadcast messages and count the number of messages that they receive from their peers. Based on the total number n of processes and an upper bound on the number t of faulty processes, a correct process tolerates faults by receiving "sufficiently many" messages. For instance, when a correct process has received $t + 1$ messages from distinct processes, at least one of these messages must originate from a non-faulty process. The main challenge is to verify such algorithms for all combinations of parameters n and t that satisfy a resilience condition, e.g., $n > 3t$.

In earlier work, we introduced threshold automata for representing processes in such algorithms and showed that systems of threshold automata have bounded diameters that do not depend on the parameters such as n and t, provided that a single-step acceleration is allowed [66]. In the contribution [27] to CONCUR'18, we reported on various extensions of this result to less restrictive forms of automata: the guards can be non-linear, shared variables can be incremented and decremented, non-trivial loops are allowed, and more general forms of acceleration are used. In the contribution [26] to ISOLA'18, we presented a parallel extension of our tool Byzantine Model Checker (ByMC), which allows one to distribute the verification queries across the computation nodes in an MPI cluster.

Our previous results apply to asynchronous algorithms. It is well-known that distributed consensus cannot be solved in purely asynchronous systems [64]. However, when an algorithm is provided with a random coin, consensus becomes solvable [56]. In [44], we introduced an approach to parameterized verification of randomized threshold-guarded distributed algorithms, which proceed in an unbounded number of rounds and toss a coin to break symmetries. This approach integrates two levels of reasoning: (1) proving safety and liveness of a single round system with ByMC by replacing randomization with non-determinism, (2) showing almost-sure termination of an algorithm by using the verification results for the non-deterministic system. To show soundness, we proved several theorems that reduce reasoning about multiple rounds to reasoning about a single round. We verified five prominent algorithms, including Ben-Or's randomized consensus [56] and randomized one-step consensus (RS-BOSCO [70]). The verification of the latter algorithm required us to run experiments in Grid5000. A paper describing these results is under review at TACAS 2019.

Another way of making consensus solvable is to impose synchrony on the executions of a distributed system. In [48] we introduced synchronous threshold automata, which execute in lock-step and count the number of processes in given local states. In general, we showed that even reachability of a parameterized set of global states in such a distributed system is undecidable. However, we proved that systems of automata with monotonic guards have bounded diameters, which allows us to use SMT-based bounded model checking as a complete parameterized verification technique. We introduced a procedure for computing the diameter of a counter system of synchronous threshold automata, applied it to the counter systems of 8 distributed algorithms from the literature, and found that their diameters are tiny (from 1 to 4). This makes our approach practically feasible, despite undecidability in general. A paper about this work is under review at TACAS 2019.

7.2.2. Symbolic Model Checking of TLA+ Specifications

Joint work with Jure Kukovec, Thanh Hai Tran, Josef Widder (TU Wien).

TLA$^+$ is a general language introduced by Leslie Lamport for specifying temporal behavior of computer systems [67]. The tool set for TLA$^+$ includes an explicit-state model checker TLC. As explicit state model checkers do not scale to large verification problems, we started the project APALACHE [3] on developing a symbolic model checker for TLA$^+$ in 2016.

In the contribution [28] to ABZ'18, we addressed the first principal challenge towards developing the symbolic model checker. We introduced a technique for identifying assignments in TLA$^+$ specifications and decomposing a monolithic TLA$^+$ specification into a set of symbolic transitions. At the TLA$^+$ community meeting 2018, we presented a prototype solution [46] to a second challenge. We have developed an SMT encoding of TLA$^+$ expressions for model checking purposes. We presented the first version of a symbolic model checker for TLA$^+$ specifications that works under the same assumptions as TLC: the input parameters are fixed and finite structures, and the reachable states are finite structures. The experimental results are encouraging, and we are thus preparing a conference submission. Finally, in a contribution to the DSN Workshop on Byzantine Consensus and Resilient Blockchains [47], we considered challenges for automatic verification techniques for Blockchain protocols.

7.2.3. Making Explicit Domain Knowledge in Formal System Development

Joint work with partners of the IMPEX project.

The IMPEX project (cf. section 8.1) advocates 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. We strive at offering rigorous mechanisms for handling domain knowledge in design models. The main results of the project are summarized in [18] and show the importance of three operations over models, namely annotation, dependency, and refactoring [38].

7.2.4. 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 about the conformance of each refinement with the model preceding it. Our main

[3] WWTF project APALACHE (ICT15-103): https://forsyte.at/research/apalache/

result during 2018 is the development of patterns for different kinds of paradigms including the iterative pattern, the recursive pattern, and the distributed pattern [30].

7.2.5. Synthesis of Security Chains for Software Defined Networks

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

The PhD work of Nicolas Schnepf focuses on applying formal methods techniques in the area of network communications, and in particular for the construction, analysis, and optimization of security functions in the setting of software-defined networks (SDN). In previous work, we defined an extension of the Pyretic language [65] for representing 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 year, our work focused on synthesizing security chains for Android applications based on their observed communications. The first step consists in inferring probabilistic finite-state automata models that represent network flows generated by Android applications. Comparing our models with automata produced by the state-of-the-art tools Invarimint and Synoptic, we obtain representations that are significantly smaller than those generated by Synoptic and as succinct as those inferred by Invarimint, but that include information about transition probability, unlike Invarimint. This work was presented at NOMS 2018 [35], [37]. In a second step, we encode security policies defined by network administrators in a rule-based program that is then used to generate a high-level representation of a security chain for the application, which is then translated to Pyretic. For example, an application that contacts different ports at the same IP address in rapid succession could be qualified as performing a port scanning attack, and these connections could then be blocked. This work was presented at AVoCS 2018 [36]. The third step consists in factorizing the chains generated for different applications in order to reduce the size of the overall chain that must be deployed in a network. A paper describing appropriate algorithms for that purpose will be presented at IM 2019.

7.2.6. 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 traditionally 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 2018, our work focused on the encoding of dynamic fault trees whose structure function needs to take into account the order in which components fail.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR International Project SYMBIONT

Project acronym: SYMBIONT.

Project title: Symbolic Methods for Biological Networks.

Duration: July 2018 – June 2021.

Coordinators: Thomas Sturm and Andreas Weber (Univ. of Bonn, Germany).

Other partners: Univ. of Lille 1, Univ. of Montpellier, Inria Saclay Île de France (Lifeware), RWTH Aachen (Department of Mathematics and Joint Research Center for Computational Biomedecine), Univ. of Kassel.

Participants: Thomas Sturm.

Abstract: SYMBIONT is an international interdisciplinary project, funded by ANR in France and by DFG in Germany under the PRCI program. It includes researchers from mathematics, computer science, systems biology, and systems medicine. Computational models in systems biology are built from molecular interaction networks and rate laws, involving parameters, resulting in large systems of differential equations. The statistical estimation of model parameters is computationally expensive and many parameters are not identifiable from experimental data. The project aims at developing novel symbolic methods, aiming at the formal deduction of principal qualitative properties of models, for complementing the currently prevailing numerical approaches. Concrete techniques include tropical geometry, real algebraic geometry, theories of singular perturbations, invariant manifolds, and symmetries of differential systems. The methods are implemented in software and validated against models from computational biology databases.

More information: https://www.symbiont-project.org/.

8.1.2. ANR Project IMPEX

Project acronym: IMPEX.

Project title: Implicit and explicit semantics integration in proof based developments of discrete systems.

Duration: December 2013 – December 2018.

Coordinator: Dominique Méry.

Other partners: ENSEEIHT/IRIT Toulouse, Supélec, Telecom Sud Paris, Systerel. Pierre Castéran from LaBRI Bordeaux also contributed to the project.

Participants: Souad Kherroubi, Dominique Méry.

Abstract: Modeling languages provide techniques and tool support for the design, synthesis, and analysis of formal models that arise during system development. The semantics of these languages is well understood by their users and is therefore implicit in the models. The languages do not provide concepts for explicitly representing characteristics (domain knowledge) resulting from an analysis of the underlying application domain [69]. We suggest that ontologies are good candidates for defining domain theories and for uniquely identifying concepts encapsulating domain knowledge. The objective [50] is to offer rigorous mechanisms for handling domain knowledge in design models. The main results of the project are summarized in [18] and show the importance of three operations over models namely annotation, dependency and refactoring [38].

8.1.3. ANR Project Formedicis

Project acronym: Formedicis.

Project title: Formal methods for the development and the engineering of critical interactive systems.

Duration: January 2017 – December 2020.

Coordinator: Bruno d'Augsbourg (Onera).

Other partners: ENSEEIHT/IRIT Toulouse, ENAC, Université de Lorraine (Veridis).

Participants: Dominique Méry.

Abstract: For 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: for example, the investigation into the crash of the Rio-Paris flight AF 447 in 2009 pointed out a design issue in the Flight Director interface as one of the original causes of the crash. Formedicis aims at designing a formal hub language, in which designers can express their requirements concerning the interactive behavior that must be embedded inside applications, and at developing a framework for validating, verifying, and implementing critical interactive applications expressed in that language.

More information: http://www.agence-nationale-recherche.fr/Project-ANR-16-CE25-0007.

8.1.4. ANR Project DISCONT

Project acronym: DISCONT.

Project title: Correct integration of discrete and continuous models.

Duration: March 2018 – February 2022.

Coordinator: Paul Gibson (Telecom Sud Paris).

Other partners: ENSEEIHT/IRIT Toulouse, LACL, ClearSy, Université de Lorraine (Veridis).

Participants: Dominique Méry.

Abstract: Cyber-Physical Systems (CPSs) connect the real world to software systems through a network of sensors and actuators that interact in complex ways, depending on context and involving different spatial and temporal scales. Typically, a discrete software controller interacts with its physical environment in a closed-loop schema where input from sensors is processed and output is generated and communicated to actuators. We are concerned with the verification of the correctness of such discrete controllers, which requires correct integration of discrete and continuous models. Correctness should arise from a design process based on sound abstractions (including discretizations) and models of the relevant physical laws. DISCONT aims at bridging the gap between the discrete and continuous worlds of formal methods and control theory. We will lift the level of abstraction above that found in current bridging techniques and provide associated methodologies and tools. Our concrete objectives are to develop a formal hybrid model, elaborate refinement steps for control requirements, propose a rational design method and support tools, and validate them based on use cases from a range of application domains.

More information: https://fusionforge.int-evry.fr/www/discont/.

8.1.5. ANR Project PARDI

Project acronym: PARDI.

Project title: Verification of parameterized distributed systems.

Duration: January 2017 – December 2020.

Coordinator: Philippe Quéinnec (ENSEEIHT/IRIT Toulouse).

Other partners: Université Paris Sud/LRI, Université Nanterre/LIP6, Inria Nancy Grand Est (Veridis).

Participants: Marie Duflot-Kremer, Igor Konnov, Stephan Merz.

Abstract: 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: http://pardi.enseeiht.fr/.

8.1.6. Inria IPL HAC SPECIS

Project acronym: HAC SPECIS.

Project title: High-performance application and computers: studying performance and correctness in simulation.

Duration: June 2016 – June 2020.

Coordinator: Arnaud Legrand (CNRS & Inria Grenoble Rhône Alpes, Polaris).

Other partners: Inria Grenoble Rhône Alpes (Avalon), Inria Rennes Bretagne Atlantique (Myriads), Inria Bordeaux Sud Ouest (Hiepacs, Storm), Inria Saclay Île de France (Mexico), Inria Nancy Grand Est (Veridis).

Participants: Marie Duflot-Kremer, Stephan Merz.

Abstract: The goal of HAC SPECIS is to answer methodological needs of HPC application and runtime developers and to allow the study of 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. VeriDis contributes 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. Yann Duplouy joined the project in December 2018 as a post-doctoral researcher with the objective of designing and implementing a statistical model checker for SimGrid.

More information: http://hacspecis.gforge.inria.fr.

8.2. European Initiatives

8.2.1. FP7 & H2020 Projects

8.2.1.1. ERC Matryoshka

Program: ERC.

Project acronym: Matryoshka.

Duration: April 2017 – March 2022.

Coordinator: Jasmin Blanchette (VU Amsterdam).

Participants: Daniel El Oraoui, Mathias Fleury, Pascal Fontaine, Hans-Jörg Schurr, Sophie Tourret, Uwe Waldmann.

Abstract: Proof assistants are increasingly used to verify hardware and software and to formalize mathematics. However, despite some 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 – but only so much can be done when viewing automatic provers as black boxes. We propose to deliver much higher levels of automation to users of proof assistants by fusing and extending two lines of research: automatic and interactive theorem proving. 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. 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, and integrate them in proof assistants. Users 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.

More information: http://matryoshka.gforge.inria.fr/.

8.2.1.2. FET-Open CSA SC^2

Program: FET Open CSA.

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.

Participants: Pascal Fontaine, Thomas Sturm.

Abstract: 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. Before the project, the two communities were largely disjoint and unaware of the achievements of each other, despite strong reasons for them to discuss and collaborate. Researchers from the two communities rarely interacted, and also their tools lacked common, mutual interfaces for unifiying their strengths. The SC^2 project initiated 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. Now that the project is finished, we believe that these activities will continue to foster cross-fertilization of both fields and bring mutual improvements to the techniques and the software tools developed by both communities.

8.2.2. Collaborations in European Programs, Except FP7 & H2020

Program: Erasmus+.

Project acronym: PIAF.

Project title: Pensée Informatique et Algorithmique au Fondamental / Computational Thinking in and Algorithmic in Primary Education.

Coordinator: Université de Liège.

Other partners: Université du Luxembourg, Saarland University, ESPE Nancy.

Participant: Marie Duflot-Kremer.

Abstract: The goal of the PIAF project is threefold: creating a repository of skills related to computational and algorithmic thinking, designing activities aiming at the acquisition of these skills, and evaluating the impact of these activities on primary school children and their computational thinking capacities.

8.3. International Initiatives

8.3.1. Inria International Partners

Project acronym: KANASA.

Title: Kanazawa-Nancy Partnership for Satistifiability and Arithmetics.

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

Start 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.

In 2016/17, Vu Xuan Tung, then a PhD student from JAIST, spent one year in the VeriDis team, and Pascal Fontaine was a reviewer of his PhD thesis, defended in 2018. There were mutual visits in 2018, and the joint research evolves towards applying SMT techniques for detecting malware in obfuscated code.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

Cezary Kaliszyk.

Date: 17 May 2018 – 17 June 2018.

Institution: University of Innsbruck, Austria.

Host: Pascal Fontaine.

Cezary Kaliszyk is an assistant professor at the University of Innsbruck. He is an expert in and a precursor of the use of machine learning in an automated reasoning context. He is the principal investigator for the ERC Starting Grant SMART (Strong Modular Proof Assistance Reasoning Across Theories). His research interests cover machine learning for theorem proving, formalization of mathematics, logical and proof translations, automated reasoning and proof data management. During his stay in Nancy, we initiated a new direction of research for quantifier instantiation, that is, using machine learning as a means of filtering the numerous instances generated by heuristic instantiation procedures in SMT.

8.4.2. Internships

Alexis Grall

Date: 1 March 2018 – 31 August 2018

Institution: Université de Lorraine

Host: Dominique Méry

In his master thesis, Alexis Grall studied the localization of Event-B models and their tranformation into the DistAlgo programming language. The Event-B models are obtained for designing distributed algorithms such as the leader election or the sliding window protocol. The transformation is proved to be sound and to preserve the properties of the Event-B models.

Axel Palaude

Date: 1 May 2018 – 31 July 2018

Institution: ENS Rennes

Host: Igor Konnov, Stephan Merz

Axel Palaude extended the short counter-example property that underlies decidability results for the verification of threshold automata (cf. section 7.2) to the case of threshold automata with real-time constraints.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Organization of Scientific Events

Jasmin Blanchette co-organized the first *Verification and Deduction Mentoring Workshop* (VDMW 2018) as part of the Federated Logic Conferences (FLoC 2018) in Oxford, UK. He also coorganized two workshops at VU Amsterdam: the First European Workshop on *Higher-Order Automated Reasoning* (Matryoshka 2018) and the Fourth International *Workshop on Automated (Co)inductive Theorem Proving* (WAIT 2018).

Igor Konnov and Stephan Merz were organizers of the fifth *Workshop on Formal Reasoning in Distributed Algorithms* (FRIDA 2018) as part of the Federated Logic Conference (FLoC 2018) in Oxford, UK.

Stephan Merz was the main organizer of the TLA$^+$ Community Meeting as part of the Federated Logic Conference (FLoC 2018) in Oxford, UK.

Thomas Sturm co-organized two international interdisciplinary workshops on *Symbolic Methods for Biological Networks* at the University of Bonn, Germany.

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 2018, VTSA took place in August in Nancy, France.

9.1.2. Program Committees

9.1.2.1. Chair of Conference Program Committees

Igor Konnov served as a chair of the *Artifact Evaluation Committee* at *Computer-Aided Verification* (CAV 2018).

Dominique Méry was a co-chair of the program committee of the 8th International Conference on Model and Data Engineering (MEDI 2018), organized in Marrakesh, Morocco, in October 2018.

Uwe Waldmann co-chaired the program committee of Deduktionstreffen 2018, the annual meeting of the Interest Group for Deduction Systems (FGDedSys) of the AI Chapter of the German Society of Informatics.

9.1.2.2. Member of Conference Program Committees

Jasmin Blanchette served on the program committees of the *NASA Formal Methods* Symposium (NFM 2018), the Conference on *Computer-Aided Verification* (CAV 2018), the *International Conference on Tests and Proofs* (TAP 2018), the *International Joint Confrence on Automated Reasoning* (IJCAR 2018), the Interational Conferene on *Interactive Theorem Proving* (ITP 2018), the ACM SIGPLAN International Conference on *Certified Programs and Proofs* (CPP 2018), and the Conference on *Artificial Intelligence and Theorem Proving* (AITP 2018). He also served on the workshop committees for the *International Workshop on the Implementation of Logics* (IWIL 2018) and the *Deduktionstreffen* 2018.

Pascal Fontaine served on the program committees of the *International Joint Confrence on Automated Reasoning* (IJCAR 2018), the *International Workshop on the Implementation of Logics* (IWIL 2018), the *International Workshop on Practical Aspects of Automated Reasoning* (PAAR 2018), the *Satisfiability Checking and Symbolic Computation* Workshop (SC-Square 2018).

Igor Konnov served on the program committees of *ACM Symposium on Principles of Distributed Computing* (PODC 2018), *Formal Methods in Computer-Aided Design* (FMCAD 2018), *International Conference on Verification and Evaluation of Computer and Communication Systems* (VECoS 2018), *International Symposium on Formal Approaches to Parallel and Distributed Systems* (4PAD 2018), *Workshop on Methods and Tools for Rigorous System Design* (MeTRiD 2018), and *Workshop on Program Semantics, Specification, and Verification* (PSSV 2018).

Dominique Méry served on the program committees of the *International Conference on Engineering of Complex Computer Systems* (ICECCS 2018), the *International Symposium on Formal Methods* (FM 2018), the *International Conference on Formal Engineering Methods* (ICFEM 2018), the *International Conference on Integrated Formal Methods* (iFM 2018), the *International Conference on ASM, Alloy, B, TLA, VDM and Z* (ABZ 2018), the *Workshop on Formal Methods for Interactive Systems* (FMIS 2018), the *Workshop in Formal Models for Mastering Multifaceted Systems* (REMEDY 2018), the *Workshop on Formal Approaches for Advanced Computing Systems* (FAACS 2018), and the *Workshop on Software Engineering in Healthcare Systems* (SEHS 2018).

Stephan Merz served on the program committees of the *International Conference on ASM, Alloy, B, TLA, VDM and Z* (ABZ), the *International Conference on Formal Methods for Industrial Critical Systems* (FMICS), the *International Conference on Formal Engineering Methods* (ICFEM), the *International Symposium on Dependable Software Engineering: Theories, Tools, and Applications* (SETTA), the *International Workshop on Automated Verification of Critical Systems* (AVoCS), and the *International Workshop about Sets and Tools* (SETS).

Sorin Stratulat served on the program committees of the *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (SYNASC 2018), the *International Conference on Information Assurance and Security* (IAS 2018), and the *International Conference on Computational Intelligence in Security for Information Systems* (CISIS 2018).

Thomas Sturm served on the program committees of *Automated Deduction in Geometry* (ADG 2018), *Computer Algebra in Scientific Computation* (CASC 2018), the *International Joint Confrence on Automated Reasoning* (IJCAR 2018), the *Satisfiability Modulo Theories* Workshop (SMT 2018), the *Satisfiability Checking and Symbolic Computation* Workshop (SC-Square 2018).

Uwe Waldmann served on the program committee of the *International Joint Confrence on Automated Reasoning* (IJCAR 2018).

Christoph Weidenbach served on the program committee of the *International Joint Conference on Automated Reasoning* (IJCAR 2018) and the senior program committee of *International Joint Conference on Artificial Intelligence* (IJCAI 2018).

9.1.3. Journals

9.1.3.1. Member of Editorial Boards

Jasmin Blanchette and Stephan Merz served as guest editors for the special issue on *Interactive Theorem Proving* (ITP 2016) of the *Journal of Automated Reasoning*.

Dominique Méry is Book Reviews Editor for *Formal Aspects of Computing*.

Thomas Sturm is an editor of the *Journal of Symbolic Computation* (Elsevier) since 2003 and an editor of *Mathematics in Computer Science* (Springer) since 2013.

Christoph Weidenbach is a member of the editorial board of the *Journal of Automated Reasoning* (Springer). He also served as an editor on the special issue on *Automated Reasoning Systems* of JAR.

9.1.4. Invited Talks

Jasmin Blanchette was invited to give a seminar talk at the University of Edinburgh on the IsaFoL (Isabelle Formalization of Logic) project.

Marie Duflot-Kremer was an invited speaker at EduCode 2018 in Brussels, Belgium, where she presented the Class'Code project.

Pascal Fontaine was an invited speaker at Deduktionstreffen 2018 in Luxembourg. He was an invited lecturer at the EPIT 2018 Software Verification Spring School in Aussois, France and at the SAT-SMT-AR school 2018 in Manchester, UK.

Igor Konnov was invited to give a tutorial at the Dagstuhl Seminar 18211 “Formal Methods and Fault-Tolerant Distributed Computing: Forging an Alliance” in Dagstuhl, Germany. He was also invited to give talks at the Workshop on Verification of Distributed Systems, Essaouira, Morocco, and Helmut Veith Memorial Workshop in Obertauern, Austria. Furthermore, he gave a talk (together with Josef Widder) on one of the research highlights of the RISE project at Alpine Verification Meeting in Wagrain, Austria.

Stephan Merz was invited to give a seminar talk at EPFL Lausanne on the use of auxiliary variables for proving refinement between TLA$^+$ specifications.

Thomas Sturm was an invited speaker at ISSAC 2018 in New York. He was furthermore invited to give a lecture at the graduate school for mathematics at RWTH Aachen University, Germany, and a seminar talk at Johannes Kepler University Linz, Austria.

Uwe Waldmann was invited to give a tutorial on Saturation Theorem Proving at the SAT/SMT/AR Summer School 2018 in Manchester, UK.

Christoph Weidenbach gave an invited talk on Robust Automated Reasoning at the 2018 Innsbruck Symposium on Integration of Automated Deduction and Interactive Theorem Proving.

9.1.5. Leadership within the Scientific Community

Jasmin Blanchette is a regular member of the CADE (*Conference on Automated Deduction*) Inc. Board of Trustees. He is also a regular member of the steering committees for the ITP (*Interactive Theorem Proving*) and TAP (*Tests and Proofs*) conference series.

Marie Duflot-Kremer is an elected member of the council of SIF, the French association for computer science.

Pascal Fontaine is an SMT-LIB manager, together with Clark Barrett (Stanford University) and Cesare Tinelli (University of Iowa). He is a regular member of steering committees for the FroCoS (*Frontiers of Combining Systems*) conference series, and for the SC-Square (*Satisfiability Checking and Symbolic Computation* workshop series. He is ex-officio member of the CADE (*Conference on Automated Deduction*) Inc. Board of Trustees. He is an elected member of the steering committee for the SMT (*Satisfiability Modulo Theories*) workshop series.

Stephan Merz is a member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*, a member of the committee for the SIF thesis award (*Prix Gilles Kahn*), and a member of the steering committee of the workshop on Automated Verification of Critical Systems (AVoCS).

Thomas Sturm is a member at large of the steering committee of the ACM 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.

9.1.6. Scientific Expertise

Dominique Méry and Stephan Merz served as experts for ANR, the French national research agency.

Christoph Weidenbach served as an expert for the German Science Foundation (DFG).

9.1.7. Research Administration

Marie Duflot-Kremer is an elected member of the council of LORIA. She was a member of the hiring committee for an associate professor at Université Paris-Est-Créteil.

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 2018, he was a member of the hiring committees of senior researchers at Inria and of junior researchers at Inria Paris. He is also a member of the *bureau* of the computer science committee of the doctoral school IAEM Lorraine and of the executive committee of the project on citizens' trust in the digital world (DigiTrust) funded by *Lorraine Université d'Excellence*.

Uwe Waldmann is a member of the admissions committee for scholarships of the International Max-Planck Research School for students aiming at a master's degree.

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

9.2. Teaching, Supervision, PhD Committees

9.2.1. Teaching

Licence: Marie Duflot-Kremer is the head of the first year for computer science students at the Faculty for Science and Technology of Université de Lorraine.

Licence: Marie Duflot-Kremer, Algorithmique et Programmation 1, 60 HETD L1 Mathématiques, Informatiques, Sciences pour l'Ingénieur, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Introduction au Web, 30 HETD L1 Mathématiques, Informatiques, Sciences pour l'Ingénieur, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Méthodologie du Travail Universitaire, 24 HETD, L1 Informatique, Université de Lorraine, France.

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

Licence: Marie Duflot-Kremer, Programmation Web, 10 HETD, L3 Informatique, Université de Lorraine, France.

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

Licence: Sorin Stratulat, Bases de données, 32 HETD, L1, ISFATES, France.

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

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

Master: Marie Duflot-Kremer and Stephan Merz, Elements of model checking, 40 HETD, M2 Informatique and Master Erasmus Mundus DESEM, Université de Lorraine, France.

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

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

Master: Pascal Fontaine is the head of the MIAGE degree at Université de Lorraine.

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: Sorin Stratulat, Analyse et conception de logiciels, 105.5 HETD, M1 Informatique, Université de Lorraine, France.

Master: Sorin Stratulat, Génie Logiciel, 30 HETD, M2 Informatique, Université de Lorraine, France.

Master: Uwe Waldmann, Automated Reasoning I, 90 HETD, Universität des Saarlandes, Germany.

Master: Sophie Tourret and Uwe Waldmann, Automated Reasoning II, 60 HETD, Universität des Saarlandes, Germany.

9.2.2. Supervision

HdR: Pascal Fontaine, Satisfiability Modulo Theories, Université de Lorraine, 8 October 2018.

PhD: Souad Kherroubi, Un cadre formel pour l'intégration de connaissances du domaine dans la conception des systèmes: Application au formalisme Event-B, Université de Lorraine, 21 December 2018. Supervised by Dominique Méry.

PhD: Andreas Teucke, *An Approximation and Refinement Approach to First-Order Automated Reasoning*, Saarland University. Supervised by Christoph Weidenbach, defended in May 2018.

PhD in progress: Martin Bromberger, Arithmetic Reasoning, Saarland University. Supervised by Thomas Sturm and 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: Alexis Grall, Integration of a modeling language and a language for programming distributed systems, Université de Lorraine. Supervised by Horatiu Cirstea and Dominique Méry, since October 2018.

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 Thomas Sturm and Christoph Weidenbach, since November 2013.

Jasmin Blanchette supervises two PhD students at VU Amsterdam and was a supervisor of Anders Schlichtkrull at TU Denmark, who defended in 2018. Igor Konnov co-supervises three PhD students at TU Wien. These PhD students are not members of VeriDis.

9.2.3. Thesis committees

Pascal Fontaine served as a reviewer in the thesis committees for Ahmed Irfan at Fondazione Bruno Kessler and University of Trento, Italy, for Vu Xuan Tung at JAIST, Japan, and for Mêton Mêton Atindehou at Université Catholique de Louvain (UCL), Belgium.

Stephan Merz served as a reviewer for the PhD theses of Bin Fang (Univ. Paris Diderot and East China Normal University) and Hai Nguyen Van (Univ. Paris Saclay). He was an examiner for the PhD committees of Evgeny Kotelnikov (Chalmers Univ.) and for the habilitation thesis of Nikolai Kosmatov (CEA Saclay).

Thomas Sturm served as a reviewer in the thesis committee for Ulrich Loup at RWTH Aachen University, Germany.

9.3. Popularization

9.3.1. Articles and Contents

In addition to the creation of unplugged computer science activities, Marie Duflot-Kremer produces documents to help others, together with videos already produced in collaboration with Inria, master and practise on their own those activities.

An article was accepted at the Educode conference on the analysis of unplugged vs. computer based programming learning [32].

9.3.2. Education

Marie Duflot-Kremer is involved in various training activities for high school teachers. She is involved in two IREM (Institute for Research on Mathematics Education) groups that produced a training session, she gave workshops in the regional APMEP (Association of Math Teachers from Public Education) day and during the “Journée ISN” (organized for teachers involved in computer science courses in high school). She also gave a conference talk and a workshop on a one day training session for teachers at the Science Museum “Le Vaisseau” in Strasbourg.

Thomas Sturm and Christoph Weidenbach co-organized the scientific track of the training program of the German team for the International Olympiad in Informatics (IOI).

9.3.3. Interventions

Marie Duflot-Kremer is involved in many outreach events where computer science is shared with a very wide audience, from 3 to 80+ years old, including “Journée des cordées de la réussite” and “Journée d’immersion” (for high school students), Fête de La Science (locally and in Paris with Inria including a theater play performed at Cité des Sciences), Math en Jeans (a program where high school students discover research through simple mathematics or computer science problems).

She gave a two-day seminar at Université Paris Nanterre to Law teachers presenting computer networks and computer security, and a talk at the “Journée GDR IA” to show to AI researchers how to present their research to a wide audience through unplugged activities.

Marie Duflot-Kremer was also one of the trainers at a three-day summer school on computer science outreach held by Société Informatique de France. During these three days the trainees discovered the concept, practised existing activities and even created their own (on subject as diverse as information leakage, Turing machines or binary integer encoding).

Concerning events organized by Inria, she took part in the Ada Lovelace Day organized by Inria Nancy – Grand Est (NGE), on three aspects: organization of the day (both scientific and practical), training of colleagues prior to the event, and supervising workshops during the event. She is also part of the FAN (Formation des Ambassadeurs du Numérique) project organised by Inria NGE and “Les Petits Débrouillards” that will, in addition to the Class’Code MOOC, train people involved in education (in school or outside) through 5 days of training seminar. She also took part in two events related to Class’Code in Bordeaux (May) and Poitiers (November), introducing the motivations of unplugged activities and their practical aspects in workshops.

9.3.4. Internal Action

Marie Duflot-Kremer is part of the “Info Sans Ordi” group affiliated to Société Informatique de France, where people share and design new unplugged activities to introduce computer science concepts.

9.3.5. Creation of Media or Tools for Science Outreach

Marie Duflot-Kremer and the Inria media team recorded three new videos presenting unplugged activities, that complement the 10 videos already existing and available on the Pixees Youtube account. She is a member of the GT7F working group (led by Interstice/Inria) that has produced a card game presenting important computer science figures (to be released in early 2019).

10. 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] P. FONTAINE. *Satisfiability Modulo Theories: state-of-the-art, contributions, project*, Université de lorraine, October 2018, Habilitation à diriger des recherches, https://tel.archives-ouvertes.fr/tel-01968404

Articles in International Peer-Reviewed Journals

[12] N. AZMY, S. MERZ, C. WEIDENBACH. *A Machine-Checked Correctness Proof for Pastry*, in "Science of Computer Programming", June 2018, vol. 158, pp. 64-80 [*DOI :* 10.1016/J.SCICO.2017.08.003], https://hal.inria.fr/hal-01768758

[13] J. C. BLANCHETTE, M. FLEURY, P. LAMMICH, C. WEIDENBACH. *A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality*, in "Journal of Automated Reasoning", 2018, vol. 61, n^{o} 1-4, pp. 333–365 [*DOI :* 10.1007/S10817-018-9455-7], https://hal.inria.fr/hal-01904579

[14] I. DRAMNESC, T. JEBELEAN, S. STRATULAT. *Mechanical Synthesis of Sorting Algorithms for Binary Trees by Logic and Combinatorial Techniques*, in "Journal of Symbolic Computation", 2019, vol. 90, n^{o} 3–41, https://hal.archives-ouvertes.fr/hal-01590654

[15] S. MERZ, H. VANZETTO. *Encoding TLA+ into unsorted and many-sorted first-order logic*, in "Science of Computer Programming", June 2018, vol. 158, pp. 3-20 [*DOI :* 10.1016/J.SCICO.2017.09.004], https://hal.inria.fr/hal-01768750

[16] M. ROMERO, M. DUFLOT-KREMER, T. VIÉVILLE. *Le jeu du robot : analyse d'une activité d'informatique débranchée sous la perspective de la cognition incarnée*, in "Review of science, mathematics and ICT education", 2018, https://hal.inria.fr/hal-01950335

Invited Conferences

[17] T. STURM. *Thirty Years of Virtual Substitution*, in "ISSAC 2018 - 43rd International Symposium on Symbolic and Algebraic Computation", New York, United States, July 2018, vol. 18 [*DOI :* 10.1145/3208976.3209030], https://hal.inria.fr/hal-01889817

International Conferences with Proceedings

[18] Y. AIT AMEUR, I. AIT-SADOUNE, P. CASTÉRAN, J. P. GIBSON, K. HACID, S. KHERROUBI, D. MÉRY, L. MOHAND OUSSAID, N. K. SINGH, L. VOISIN. *On the Importance of Explicit Domain Modelling in Refinement-Based Modelling Design. Experiments with Event-B*, in "ABZ 2018 - 6th International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z", Southampton, United Kingdom, M. BUTLER, A. RASCHKE, T. S. HOANG, K. REICHL (editors), Lecture Notes in Computer Science, Springer, June 2018, vol. 10817, pp. 425–430 [*DOI :* 10.1007/978-3-319-91271-4_35], https://hal.archives-ouvertes.fr/hal-01797538

[19] A. BENTKAMP, S. CRUANES, J. C. BLANCHETTE, U. WALDMANN. *Superposition for Lambda-Free Higher-Order Logic*, in "IJCAR 2018 - 9th International Joint Conference on Automated Reasoning", Oxford, United Kingdom, July 2018, https://hal.inria.fr/hal-01904595

[20] J. C. BLANCHETTE. *Formalizing the Metatheory of Logical Calculi and Automatic Provers in Isabelle/HOL (Invited Talk)*, in "CPP 2019 - The 8th ACM SIGPLAN International Conference on Certified Programs and Proofs", Cascais, Portugal, CPP 2019 - The 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, January 2019 [*DOI :* 10.1145/3293880.3294087], https://hal.archives-ouvertes.fr/hal-01937136

[21] J. C. BLANCHETTE, N. PELTIER, S. ROBILLARD. *Superposition with Datatypes and Codatatypes*, in "IJCAR 2018 - 9th International Joint Conference on Automated Reasoning", Oxford, United Kingdom, July 2018, https://hal.inria.fr/hal-01904588

[22] M. BROMBERGER. *A Reduction from Unbounded Linear Mixed Arithmetic Problems into Bounded Problems*, in "IJCAR 2018 - 9th International Joint Conference on Automated Reasoning", Oxford, United Kingdom, D. GALMICHE, S. SCHULZ, R. SEBASTIANI (editors), Lecture Notes in Computer Science, Springer, July 2018, vol. 10900, pp. 329-345, https://hal.inria.fr/hal-01942228

[23] M. FLEURY, J. C. BLANCHETTE, P. LAMMICH. *A verified SAT solver with watched literals using imperative HOL*, in "CPP 2018 - The 7th ACM SIGPLAN International Conference on Certified Programs and Proofs", Los Angeles, United States, ACM Press, January 2018 [*DOI :* 10.1145/3167080], https://hal.inria.fr/hal-01904647

[24] P. FONTAINE, M. OGAWA, T. STURM, V. KHANH TO, X. TUNG VU. *Wrapping Computer Algebra is Surprisingly Successful for Non-Linear SMT*, in "SC-square 2018 - Third International Workshop on Satisfiability Checking and Symbolic Computation", Oxford, United Kingdom, July 2018, https://hal.inria.fr/hal-01946733

[25] H. HONG, T. STURM. *Positive Solutions of Systems of Signed Parametric Polynomial Inequalities*, in "CASC 2018 - International Workshop on Computer Algebra in Scientific Computing", Lille, France, LNCS, September 2018, vol. 11077, pp. 238 - 253 [*DOI :* 10.1007/978-3-319-99639-4_17], https://hal.inria.fr/hal-01889827

[26] I. KONNOV, J. WIDDER. *ByMC: Byzantine Model Checker*, in "ISoLA 2018 - 8th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation", Limassol, Cyprus, Lecture Notes in Computer Science, October 2018, vol. 11246, pp. 327-342 [*DOI :* 10.1007/978-3-030-03424-5_22], https://hal.inria.fr/hal-01909653

[27] J. KUKOVEC, I. KONNOV, J. WIDDER. *Reachability in Parameterized Systems: All Flavors of Threshold Automata*, in "CONCUR 2018 - 29th International Conference on Concurrency Theory", Beijing, China, September 2018 [*DOI :* 10.4230/LIPIcs.CONCUR.2018.19], https://hal.inria.fr/hal-01871142

[28] J. KUKOVEC, T.-H. TRAN, I. KONNOV. *Extracting Symbolic Transitions from $TLA+$ Specifications*, in "Abstract State Machines, Alloy, B, TLA, VDM, and Z. ABZ 2018", Southampton, United Kingdom, M. BUTLER, A. RASCHKE, T. S. HOANG, K. REICHL (editors), Lecture Notes in Computer Science, June 2018, vol. 10817, pp. 89-104 [*DOI :* 10.1007/978-3-319-91271-4_7], https://hal.inria.fr/hal-01871131

[29] S. LENGLET, A. SCHMITT. *HOπ in Coq*, in "CPP 2018 - The 7th ACM SIGPLAN International Conference on Certified Programs and Proofs", Los Angeles, United States, January 2018, 14 p. [*DOI :* 10.1145/3167083], https://hal.inria.fr/hal-01614987

[30] D. MÉRY. *Modelling by Patterns for Correct-by-Construction Process*, in "ISOLA 2018 - 8th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation", Limassol, Cyprus, Leveraging Applications of Formal Methods, Verification and Validation. Modeling - 8th International Symposium, ISoLA 2018, Springer, November 2018, vol. 11244, pp. 399-423, https://hal.inria.fr/hal-01933971

[31] A. REYNOLDS, H. BARBOSA, P. FONTAINE. *Revisiting Enumerative Instantiation*, in "TACAS 2018 - 24th International Conference on Tools and Algorithms for the Construction and Analysis of Systems", Thessaloniki, Greece, D. BEYER, M. HUISMAN (editors), LNCS, Springer, April 2018, vol. 10806, 20 p. , https://hal.archives-ouvertes.fr/hal-01877055

[32] M. ROMERO, B. LILLE, T. VIÉVILLE, M. DUFLOT-KREMER, C. DE SMET, D. BELHASSEIN. *Analyse comparative d'une activité d'apprentissage de la programmation en mode branché et débranché*, in "Educode - Conférence internationale sur l'enseignement au numérique et par le numérique", Bruxelles, Belgium, August 2018, https://hal.inria.fr/hal-01861732

[33] A. SCHLICHTKRULL, J. C. BLANCHETTE, D. TRAYTEL. *A Verified Prover Based on Ordered Resolution*, in "CPP 2019 - The 8th ACM SIGPLAN International Conference on Certified Programs and Proofs", Cascais, Portugal, CPP 2019 - The 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, January 2019 [*DOI :* 10.1145/3293880.3294100], https://hal.archives-ouvertes.fr/hal-01937141

[34] A. SCHLICHTKRULL, J. C. BLANCHETTE, D. TRAYTEL, U. WALDMANN. *Formalizing Bachmair and Ganzinger's Ordered Resolution Prover*, in "IJCAR 2018 - 9th International Joint Conference on Automated Reasoning", Oxford, United Kingdom, July 2018, https://hal.inria.fr/hal-01904610

[35] N. SCHNEPF, R. BADONNEL, A. LAHMADI, S. MERZ. *Generation of SDN policies for protecting Android environments based on automata learning*, in "NOMS 2018 - IEEE/IFIP Network Operations and Management Symposium", Taipei, Taiwan, Proceedings of the IEEE/IFIP Network Operations and Management Symposium (IEEE/IFIP NOMS), IEEE, April 2018 [*DOI :* 10.1109/NOMS.2018.8406153], https://hal.archives-ouvertes.fr/hal-01892390

[36] N. Schnepf, R. Badonnel, A. Lahmadi, S. Merz. *Rule-Based Synthesis of Chains of Security Functions for Software-Defined Networks*, in "AVOCS 2018 - 18th International Workshop on Automated Verification of Critical Systems", Oxford, United Kingdom, Proceedings of the International Workshop on Automated Verification of Critical Systems, July 2018, https://hal.archives-ouvertes.fr/hal-01892423

[37] N. Schnepf, R. Badonnel, A. Lahmadi, S. Merz. *Synaptic: A formal checker for SDN-based security policies*, in "NOMS 2018 - IEEE/IFIP Network Operations and Management Symposium", Taipei, Taiwan, IEEE, April 2018 [*DOI :* 10.1109/NOMS.2018.8406122], https://hal.archives-ouvertes.fr/hal-01892397

[38] N. K. Singh, Y. Ait Ameur, D. Méry. *Formal Ontological Driven Model Refactoring*, in "ICECCS 2018 - 23rd International Conference on Engineering of Complex Computer Systems", Melbourne, Australia, IEEE, December 2018, https://hal.inria.fr/hal-01939006

[39] S. Stratulat. *Validating Back-links of FOLID Cyclic Pre-proofs*, in "CL&C'18 - Seventh International Workshop on Classical Logic and Computation", Oxford, United Kingdom, July 2018, vol. 281, pp. 39–53, https://hal.archives-ouvertes.fr/hal-01883826

Books or Proceedings Editing

[40] E. H. Abdelwahed, L. Bellatreche, D. Benslimane, M. Golfarelli, S. Jean, D. Méry, K. Nakamatsu, C. Ordonez (editors). *New Trends in Model and Data Engineering*, Springer, Marrakesh, Morocco, October 2018, vol. Communications in Computer and Information Science, n^{o} 929 [*DOI :* 10.1007/978-3-030-02852-7], https://hal.inria.fr/hal-01933975

[41] E. H. Abdelwahed, L. Bellatreche, M. Golfarelli, D. Méry, C. Ordonez (editors). *Model and Data Engineering*, Lecture Notes in Computer Science, Springer, Marrakech, Morocco, October 2018, vol. 11163, https://hal.inria.fr/hal-01933977

[42] R. Laleau, D. Méry, S. Nakajima, E. Troubitsyna (editors). *Proceedings Joint Workshop on Handling IMPlicit and EXplicit knowledge in formal system development (IMPEX) and Formal and Model-Driven Techniques for Developing Trustworthy Systems (FM&MDD)*, EPTCS, May 2018, vol. 271, https://arxiv.org/abs/1805.04636 [*DOI :* 10.4204/EPTCS.271], https://hal.inria.fr/hal-01933762

Research Reports

[43] A. J. Reynolds, H. Barbosa, P. Fontaine. *Revisiting Enumerative Instantiation*, University of Iowa ; Inria, March 2018, https://hal.inria.fr/hal-01744956

Other Publications

[44] N. Bertrand, I. Konnov, M. Lazic, J. Widder. *Verification of Randomized Distributed Algorithms under Round-Rigid Adversaries*, November 2018, Experiments presented in this paper were carried out using the Grid5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations, see grid5000.fr, https://hal.inria.fr/hal-01925533

[45] R. Chen, C. Cohen, J.-J. Levy, S. Merz, L. Thery. *Formal Proofs of Tarjan's Algorithm in Why3, Coq, and Isabelle*, October 2018, https://arxiv.org/abs/1810.11979 - working paper or preprint, https://hal.inria.fr/hal-01906155

[46] I. KONNOV, J. KUKOVEC, T. H. TRAN. *BmcMT: Bounded Model Checking of TLA + Specifications with SMT*, July 2018, TLA+ Community Meeting 2018, https://hal.inria.fr/hal-01899719

[47] I. KONNOV, S. MERZ. *Model Checking of Fault-Tolerant Distributed Algorithms: from Classics towards Contemporary*, June 2018, BCRB 2018 - DSN Workshop on Byzantine Consensus and Resilient Blockchains, https://hal.inria.fr/hal-01899723

[48] I. STOILKOVSKA, I. KONNOV, J. WIDDER, F. ZULEGER. *Verifying Safety of Synchronous Fault-Tolerant Algorithms by Bounded Model Checking*, November 2018, working paper or preprint, https://hal.inria.fr/hal-01925653

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] 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

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

[53] 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

[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] 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

[56] M. BEN-OR. *Another Advantage of Free Choice: Completely Asynchronous Agreement Protocols (Extended Abstract)*, in "PODC", 1983, pp. 27–30

[57] 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

[58] 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

[59] R. CHEN, J.-J. LEVY. *A Semi-automatic Proof of Strong connectivity*, in "Proc. 9th Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE 2017)", A. PASKEVICH, T. WIES (editors), Lecture Notes in Computer Science, Springer, July 2017, pp. 49-65

[60] 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

[61] 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

[62] A. CIMATTI, A. GRIGGIO, B. J. SCHAAFSMA, R. SEBASTIANI. *The MathSAT5 SMT Solver*, in "Tools and Algorithms for the Construction and Analysis of Systems - 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings", N. PITERMAN, S. A. SMOLKA (editors), Springer, 2013, vol. 7795, pp. 93–107

[63] B. DUTERTRE, L. DE MOURA. *A Fast Linear–Arithmetic Solver for DPLL(T)*, in "Proceedings of CAV", LNCS, Springer, August, 17–20 2006, vol. 4144, pp. 81–94

[64] M. J. FISCHER, N. A. LYNCH, M. S. PATERSON. *Impossibility of Distributed Consensus with one Faulty Process*, in "J. ACM", 1985, vol. 32, n^{o} 2, pp. 374–382

[65] 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

[66] I. V. KONNOV, H. VEITH, J. WIDDER. *On the completeness of bounded model checking for threshold-based distributed algorithms: Reachability*, in "Inf. Comput.", 2017, vol. 252, pp. 95–109

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

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

[69] 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

[70] Y. J. SONG, R. VAN RENESSE. *Bosco: One-Step Byzantine Asynchronous Consensus*, in "DISC", LNCS, 2008, vol. 5218, pp. 438–450

[71] 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^{o} 3-4, pp. 483 - 502 [*DOI :* 10.1007/S11786-017-0319-Z], https://hal.inria.fr/hal-01648690

[72] L. M. DE MOURA, N. BJØRNER. *Z3: An Efficient SMT Solver*, in "Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings", C. R. RAMAKRISHNAN, J. REHOF (editors), Lecture Notes in Computer Science, Springer, 2008, vol. 4963, pp. 337–340