IN PARTNERSHIP WITH:
CNRS

Max-Planck-Institut für Informatik Saarbrücken

Université de Lorraine

Activity Report 2019

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

1. Team, Visitors, External Collaborators **1**

2. Overall Objectives **2**

3. Research Program **3**

3.1. Automated and Interactive Theorem Proving 3

3.2. Formal Methods for Developing and Analyzing Algorithms and Systems 4

4. Application Domains **5**

5. Highlights of the Year **5**

6. New Software and Platforms **5**

6.1. Redlog 5

6.2. SPASS 6

6.3. SPASS-SATT 6

6.4. veriT 7

6.5. SPIKE 7

6.6. TLAPS 8

6.7. Apalache 8

6.8. IMITATOR 9

6.9. ByMC 9

7. New Results **10**

7.1. Automated and Interactive Theorem Proving 10

7.1.1. Combination of Satisfiability Procedures 10

7.1.2. Quantifier Handling in SMT 10

7.1.3. Higher-Order SMT 11

7.1.4. Proofs for SMT 11

7.1.5. Clause Learning from Simple Models 11

7.1.6. SPASS-SATT 11

7.1.7. Extension of a Highly Efficient Prover to λ-free Higher-Order Logic 12

7.1.8. Extension of the Superposition Calculus with λ-Abstractions 12

7.1.9. Automated Reasoning over Biological Networks 12

7.1.10. Towards an Improved Encoding of TLA+ Proof Obligations 13

7.1.11. Formal Proofs of Tarjan's Algorithm 13

7.1.12. Implementation of an Efficient Validation of FOLID Cyclic Induction Reasoning 13

7.2. Formal Methods for Developing and Analyzing Algorithms and Systems 13

7.2.1. Synthesis of Security Chains for Software Defined Networks 13

7.2.2. Satisfiability Techniques for Reliability Assessment 14

7.2.3. Statistical Model Checking of Distributed Programs 14

7.2.4. Parameterized Verification of Threshold-Guarded Fault-Tolerant Distributed Algorithms 14

7.2.5. Symbolic Model Checking of TLA+ Specifications 15

7.2.6. Incremental Development of Systems and Algorithms 15

8. Bilateral Contracts and Grants with Industry **15**

9. Partnerships and Cooperations **15**

9.1. Regional Initiatives 15

9.2. National Initiatives 16

9.2.1. PIA2 ISITE LUE 16

9.2.2. ANR International Project ProMiS 16

9.2.3. ANR International Project SYMBIONT 16

9.2.4. ANR Project Formedicis 17

9.2.5. ANR Project DISCONT 17

9.2.6. ANR Project PARDI 18

9.2.7. Inria IPL HAC SPECIS 18

9.2.8. DFG Transregional Research Center 248 CPEC 18
9.3. European Initiatives 19
9.3.1. FP7 & H2020 Projects 19
9.3.2. Collaborations in European Programs, Except FP7 & H2020 19
9.4. International Research Visitors 20
9.4.1. Visits of International Scientists 20
9.4.2. Visits to International Teams 21
10. Dissemination **21**
10.1. Promoting Scientific Activities 21
10.1.1. Organization of Scientific Events 21
10.1.2. Program Committees 21
10.1.2.1. Chair of Conference Program Committees 21
10.1.2.2. Membership in Conference Program Committees 21
10.1.3. Journals 22
10.1.4. Invited Talks 23
10.1.5. Leadership within the Scientific Community 23
10.1.6. Scientific Expertise 23
10.1.7. Research Administration 24
10.2. Teaching, Supervision, Thesis Committees 24
10.2.1. Teaching 24
10.2.2. Supervision 24
10.2.3. Thesis Committees 25
10.3. Popularization 25
10.3.1. Responsibilities at Inria or Beyond 25
10.3.2. Articles and Contents 26
10.3.3. Education 26
10.3.4. Interventions 26
10.3.5. Creation of Media or Tools for Science Outreach 26
11. Bibliography **26**

Project-Team VERIDIS

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

VeriDis is a joint research group of CNRS, Inria, Max-Planck-Institut für Informatik, and Université de Lorraine. It consists of members of the Mosel research group at LORIA, Nancy, France, and members of the Automation of Logic group at 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, until September 2019]
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

Étienne André [Univ. de Lorraine, Professor, from September 2019, HDR]
Marie Duflot-Kremer [Univ. de Lorraine, Associate Professor]
Pascal Fontaine [Univ. de Lorraine, Associate Professor, until October 2019, HDR]
Dominique Méry [Univ de Lorraine, Professor]
Sorin Stratulat [Univ de Lorraine, Associate Professor]

Post-Doctoral Fellows

Martin Bromberger [Max-Planck Institut für Informatik, from December 2019]
Yann Duplouy [Inria]
Hamid Rahkooy [CNRS, from June 2019]
Sophie Tourret [Max-Planck Institut für Informatik]
Marco Voigt [Max-Planck Institut für Informatik, from August 2019]

PhD Students

Martin Bromberger [Max-Planck Institut für Informatik, until December 2019]

Antoine Defourné [Inria, from March 2019]
Margaux Duroeulx [Univ. de Lorraine]
Daniel El Ouraoui [Inria]
Alberto Fiori [Max-Planck Institut für Informatik]
Mathias Fleury [Max-Planck Institut für Informatik]
Alexis Grall [Univ. de Lorraine]
Pierre Lermusiaux [Univ. de Lorraine]
Nicolas Schnepf [Inria, joint with Resist team, until September 2019]
Hans-Jörg Schurr [Inria]
Marco Voigt [Max-Planck Institut für Informatik, until July 2019]

Interns and Apprentices

Guillaume Ambal [Ecole Normale Supérieure Lyon, from March 2019 until June 2019]
Manon Blanc [Inria, from June 2019 until July 2019]
Pierre Henry [Univ. de Lorraine, from April 2019 until June 2019]
Brandon Hornbeck [Univ. de Lorraine, from April 2019 until June 2019]
Kseniia Iankina [Univ. de Lorraine, from March 2019 until July 2019]
Ali Kumail [Univ. de Lorraine, from March 2019 until July 2019]
Viktor Sergeev [Inria, from March 2019 until July 2019]

Administrative Assistants

Sophie Drouot [Inria]
Sylvie Hilbert [CNRS]
Jennifer Müller [Max-Planck Institut für Informatik]

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 development and analysis of concurrent and distributed algorithms and systems, based on mathematically precise and practically applicable development methods. The techniques that we develop are intended 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.

Within this context, we work on techniques for *automated theorem proving* for expressive languages based on first-order logic, with support for theories (fragments of arithmetic, set theory etc.) that are relevant for specifying algorithms and systems. 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 fundamental undecidability of the problem, this cannot be achieved in general. Nevertheless, we have observed important advances in automated deduction in recent years, to which we have contributed. These advances suggest that a substantially higher degree of automation can be achieved over what is available in today's tools supporting deductive verification. Our techniques are developed within SMT (satisfiability modulo theories) solving and superposition reasoning, the two main frameworks of contemporary automated reasoning that have complementary strengths and weaknesses, and we are interested in making them converge when appropriate. Techniques developed within the symbolic computation domain, such as algorithms for quantifier elimination for appropriate theories, are also relevant, and we are working on integrating them into our portfolio of techniques. In order to handle expressive input languages, we are working on techniques that encompass tractable fragments of higher-order logic, for example for specifying inductive or co-inductive data types, for automating proofs by induction, or for handling collections defined through a characteristic predicate.

Since full automatic verification remains elusive, another line of our research targets *interactive proof platforms*. We intend these platforms to benefit from our work on automated deduction by incorporating powerful automated backends and thus raise the degree of automation beyond what current proof assistants can offer. Since most conjectures stated by users are initially wrong (due to type errors, omitted hypotheses or overlooked border cases), it is also important that proof assistants be able to detect and explain such errors rather than letting users waste considerable time in futile proof attempts. Moreover, increased automation must not come at the expense of trustworthiness: skeptical proof assistants expect to be given an explanation of the proof found by the backend prover that they can certify.

Our methodological and foundational research is accompanied by the development of *efficient software tools*, several of which go beyond pure research prototypes: they have been used by others, have been integrated in proof platforms developed by other groups, and participate in international competitions. We also validate our work on proof techniques by applying them to the *formal development of algorithms and systems*. We mainly target high-level descriptions of concurrent and distributed algorithms and systems. This class of algorithms is by now ubiquitous, ranging from multi- and many-core algorithms to large networks and cloud computing, and their formal verification is notoriously difficult. Targeting high levels of abstraction allows the designs of such systems to be verified before an actual implementation has been developed, contributing to reducing the costs of formal verification. The potential of distributed systems for increased resilience to component failures makes them attractive in many contexts, but also makes formal verification even more important and challenging. Our work in this area aims at identifying classes of algorithms and systems for which we can provide guidelines and identify patterns of formal development that makes verification less an art and more an engineering discipline. We mainly target components of operating systems, distributed and cloud services, and networks of computers or mobile devices.

Beyond formal verification, we pursue applications of some of the symbolic techniques that we are developing in other domains. We have observed encouraging success in using techniques of symbolic computation for the qualitative analysis of biological and chemical regulation networks described by systems of ordinary differential equations that were previously only accessible to large-scale simulation. This work is being pursued within a large-scale interdisciplinary collaboration. It aims for our work grounded in verification having an impact on the sciences, beyond engineering, which will feed back into our core formal methods community.

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 [56] 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, i.e. by embedding decision procedures into the superposition framework or by allowing an SMT solver to accept axiomatizations for plug-in theories. We also develop specific decision procedures for theories such as non-linear real arithmetic that are important when reasoning about certain classes of (e.g., real-time) systems but that also have interesting applications beyond verification.

We rely on interactive theorem provers for reasoning about specifications at a high level of abstraction when fully automatic verification is not (yet) feasible. An interactive proof platform should help verification engineers lay out the proof structure at a sufficiently high level of abstraction; powerful automatic plug-ins should then discharge the resulting proof steps. Members of VeriDis have ample experience in the specification and subsequent machine-assisted, interactive verification of algorithms. In particular, we participate in a project at the joint Microsoft Research-Inria Centre on the development of methods and tools for the formal proof of TLA$^+$ [66] 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* [55], [57], [70] 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

[1] Satisfiability Modulo Theories [58]

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.

Our work on symbolic procedures for solving polynomial constraints finds applications beyond verification. In particular, we have been working in interdisciplinary projects with researchers from mathematics, computer science, system biology, and system medicine on the analysis of molecular interaction networks in order to infer the principal qualitative properties of models. Our techniques complement numerical analysis techniques and are validated against collections of models from computational biology.

5. Highlights of the Year

5.1. Highlights of the Year

5.1.1. Awards

Christoph Weidenbach received the Skolem test-of-time award of CADE, the international conference on automated deduction, for his paper *Towards an Automated Analysis of Security Protocols* [72].

Martin Bromberger, Mathias Fleury, Simon Schwarz and Christoph Weidenbach received the best student paper award at CADE 27 for their paper *SPASS-SATT: A CDCL(LA) Solver* .

BEST PAPER AWARD:

[31]
M. BROMBERGER, M. FLEURY, S. SCHWARZ, C. WEIDENBACH. *SPASS-SATT: A CDCL(LA) Solver*, in "27th International Conference on Automated Deduction (CADE-27)", Natal, Brazil, P. FONTAINE (editor), Lecture Notes in Computer Science, 2019, vol. 11716, pp. 111-122 [*DOI :* 10.1007/978-3-030-29436-6_7], https://hal.inria.fr/hal-02405524

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 started to go for major revisions and improvements of Redlog's software architecture, which are still under way.

Redlog, as well as the underlying Reduce, depends on a quite minimalistic Lisp 1 dialect called 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.

Scientifically we are currently improving on Parametric Gaussian Elimination in Reduce/Redlog, which has various applications in our bilateral interdisciplinary ANR/DFG project SYMBIONT (Symbolic Methods for Biological Networks), e.g., classification of real singularities of systems of implicit ordinary differential equations.

- 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.

- 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: SPASS-SATT participated in the SMT competition 2019 in the quantifier free integer and rational linear arithmetic categories. It scored first on rational linear arithmetic and second on integer linear arithmetic. (The winner of the latter category was a portfolio solver that includes SPASS-SATT.) The main improvements are due to an advanced translation to clause normal form, a close interaction between the theory and the SAT solvers, and a new transformation turning unbounded integer problems into bounded integer problems.

- 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. 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 2019 have been focused on quantifier handling, higher logic, and proof production.

The veriT solver participated in the SMT competition SMT-COMP 2019 with good results. In particular, it took the bronze medal in the QF_UF division, solving as many problems as the two leading solvers but taking somewhat more time.

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, and it is integrated within *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

6.5. 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
- URL: https://github.com/sorinica/spike-prover/wiki

6.6. TLAPS

TLA+ proof system

KEYWORD: Proof assistant

SCIENTIFIC 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.

FUNCTIONAL DESCRIPTION: TLAPS is a proof assistant for the TLA+ specification language.

NEWS OF THE YEAR: Work in 2019 focused on providing support for reasoning about TLA+'s ENABLED and action composition constructs. We also prepared a minor release, fixing some issues and switching to Z3 as the default SMT back-end solver.

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

6.7. Apalache

Abstraction-based Parameterized TLA+ Checker

KEYWORD: Model Checker

SCIENTIFIC DESCRIPTION: Apalache is a symbolic model checker that works under the following assumptions:

(1) As in TLC, all specification parameters are fixed and finite, e.g., the system is initialized integers, finite sets, and functions of finite domains and co-domains. (2) As in TLC, all data structures evaluated during an execution are finite, e.g., a system specification cannot operate on the set of all integers. (3) Only finite executions up to a given bound are analysed.

Apalache translates bounded executions of a TLA+ specifications into a set of quantifier-free SMT constraints. By querying the SMT solver, the model checker either finds a counterexample to an invariant, or proves that there is no counterexample up to given computation length.

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.

NEWS OF THE YEAR: In 2019, we have simplified the set of rewriting rules, which are used in the translation from TLA+ to SMT. We have shown that the rules are sound, that is, that the translator produces a set of SMT constraints that are equisatisfiable to the given TLA+ formula. We have conducted the experiments on 10 TLA+ specifications of distributed algorithms. When running bounded model checking, Apalache outperforms TLC in some cases. When checking inductive invariants, Apalache runs significantly faster than TLC. These results were reported at ACM OOPSLA 2019.

- Partner: Technische Universität Wien
- Contact: Igor Konnov
- Publications: hal-01899719v1 - hal-01871131v1 - hal-02280888v1
- URL: https://forsyte.at/research/apalache/

6.8. IMITATOR

KEYWORDS: Verification - Parametric model - Parameter synthesis - Model Checking - Model Checker - Timed automata

FUNCTIONAL DESCRIPTION: IMITATOR is a software tool for parametric verification and robustness analysis of real-time systems with parameters. It relies on the formalism of networks of parametric timed automata, augmented with integer variables and stopwatches.

- Participants: Etienne Andre and Jaime Eduardo Arias Almeida
- Partner: Loria
- Contact: Etienne Andre
- Publications: The Inverse Method - Formalizing Time4sys using parametric timed automata - Minimal-Time Synthesis for Parametric Timed Automata - A benchmark library for parametric timed model checking
- URL: https://www.imitator.fr/

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., $n > 3t$.

ByMC supports a parallel mode, which allows one to run verification experiments in an MPI cluster such as Grid5000 and Vienna Scientific Cluster.

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: In 2019, we have shown how to apply ByMC to randomized fault-tolerant consensus algorithms such as randomized consensus by Ben-Or and RS-BOSCO. This result was presented at CONCUR 2019.

- 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, Antoine Defourné, Daniel El Ouraoui, Alberto Fiori, Mathias Fleury, Pascal Fontaine, Stephan Merz, Hamid Rahkooy, Hans-Jörg Schurr, Sorin Stratulat, Thomas Sturm, Sophie Tourret, Marco Voigt, Uwe Waldmann, Christoph Weidenbach.

7.1.1. Combination of Satisfiability Procedures

Joint work with Christophe Ringeissen (Inria Nancy – Grand Est, Pesto) and Paula Chocron (Insikt Intelligence, 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 [59]. 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 [60] 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 and 2019, we have been improving the framework and unified both results. This was published in the Journal of Automated Reasoning in 2019 [19].

7.1.2. Quantifier Handling in SMT

Joint work with 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.

In 2019, we investigated machine learning techniques for predicting the usefulness of an instance in order to decrease the number of instances passed to the SMT solver. For this, we proposed a meaningful way to characterize the state of an SMT solver, to collect instantiation learning data, and to integrate a predictor in the core of a state-of-the-art SMT solver. This ultimately leads to more efficient SMT solving for quantified problems.

7.1.3. Higher-Order SMT

Joint work with Haniel Barbosa, Andrew Reynolds, Cesare Tinelli (Univ. of Iowa), and Clark Barrett (Stanford)

SMT solvers have throughout the years been able to cope with increasingly expressive formulas, from ground logics to full first-order logic (FOL). In contrast, the extension of SMT solvers to higher-order logic (HOL) was mostly unexplored. We proposed a pragmatic extension for SMT solvers to support HOL reasoning natively without compromising performance on FOL reasoning, thus leveraging the extensive research and implementation efforts dedicated to efficient SMT solving. We showed how to generalize data structures and the ground decision procedure to support partial applications and extensionality, as well as how to reconcile quantifier instantiation techniques with higher-order variables. We also discussed a separate approach for redesigning an SMT solver for higher-order logic from the ground up via new data structures and algorithms. We applied our pragmatic extension to the CVC4 SMT solver and discussed a redesign of the veriT SMT solver. Our evaluation showed that they are competitive with state-of-the-art HOL provers and often outperform the traditional encoding into FOL.

This result was published at CADE 2019 [27]. We are also currently investigating extending the CCFV algorithm to higher-order logic.

7.1.4. Proofs for SMT

We have previously developed a framework for processing formulas in automatic theorem provers, with generation of detailed proofs that can be checked by external tools, including skeptical proof assistants. 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. In 2019, the format of proof output was further improved, while also improving the reconstruction procedure in the proof assistant Isabelle/HOL. This allowed the tactic using SMT with proofs to be regularly suggested by Sledgehammer as the fastest method to automatically solve proof goals. This was the subject of a workshop publication [36].

7.1.5. Clause Learning from Simple Models

The goal of this research is to guide inferences in expressive logics via simple models. Intuitively, a model is simple if computations with respect to the model can be done in polynomial time. We have shown that for first-order logic, models built from ground literals are sufficient to guide resolution inferences between non-ground clauses [35]. We have also investigated the expressivity of model representation formalisms in general [41]. Model representation formalisms built on atoms with only linear variable occurrences have the finite model property. Hence, they cannot represent infinite models.

7.1.6. SPASS-SATT

We have further developed our CDCL(T) solver SPASS-SATT. It is the combination of our SAT solver SPASS-SAT with highly efficient theory solvers for linear arithmetic [31]. SPASS-SATT showed good performance at the SMT competition 2019 where it won the category on linear rational arithmetic and scored second on linear integer arithmetic. The winner of the linear integer arithmetic category was a portfolio solver including SPASS-SATT. Our main improvements are due to an advanced clause normal form translation, a close interaction between the theory solvers and the SAT solver SPASS-SAT, and and a new transformation turning unbounded integer problems into bounded integer problems.

7.1.7. Extension of a Highly Efficient Prover to λ-free Higher-Order Logic

Joint work with Simon Cruanes (Aesthetic Integration), Stephan Schulz (DHBW Stuttgart), and Petar Vukmirović (VU Amsterdam).

Superposition-based provers, such as E, SPASS, and Vampire, are among the most successful reasoning systems for first-order logic. They serve as backends in various frameworks, including software verifiers, automatic higher-order theorem provers, and one-click "hammers" in proof assistants. Decades of research have gone into refining calculi, devising efficient data structures and algorithms, and developing heuristics to guide proof search. This work has mostly focused on first-order logic with equality, with or without arithmetic.

To obtain better performance, we propose to start with a competitive first-order prover and extend it to full higher-order logic one feature at a time. Our goal is a *graceful* extension, in keeping with the zero-overhead principle: *What you don't use, you don't pay for.*

As a stepping stone towards full higher-order logic, we initially restricted our focus to a higher-order logic without λ-expressions. Compared with first-order logic, its distinguishing features are partial application and applied variables. Our vehicle is E, a prover developed primarily by Schulz. It is written in C and offers good performance. E regularly scores among the top systems at the CASC competition, and usually is the strongest open source prover in the relevant divisions. It also serves as a backend for competitive higher-order provers.

Our experiments show that the λ-free higher-order version of E is practically as fast as E on first-order problems and can also prove higher-order problems that do not require synthesizing λ-terms. As a next step, we plan to add support for λ-terms and higher-order unification. This work is described in a TACAS 2019 conference paper [42]; an extended version of this paper has been invited to a special issue of the *International Journal on Software Tools for Technology Transfer*.

7.1.8. Extension of the Superposition Calculus with λ-Abstractions

Joint work with Alexander Bentkamp (VU Amsterdam) and Petar Vukmirović (VU Amsterdam).

We designed a superposition calculus for a clausal fragment of extensional polymorphic higher-order logic that includes anonymous functions but excludes Booleans. The inference rules work on $\beta\eta$-equivalence classes of λ-terms and rely on higher-order unification to achieve refutational completeness.

We implemented the calculus in the Zipperposition prover. Our empirical evaluation includes benchmarks from the TPTP (Thousands of Problems for Theorem Provers) and interactive verification problems exported from Isabelle/HOL. The results appear promising and suggest that an optimized implementation inside a competitive prover such as E, SPASS, or Vampire would outperform existing higher-order automatic provers. This research was presented at the CADE 2019 conference [28].

7.1.9. Automated Reasoning over Biological Networks

[54] study toricity of steady state ideals of biological models. From a computational point of view, models identified as toric allow to employ tools from toric geometry for a complexity reduction step. From a scientific point of view, toric models are known to have scale invariant multistationarity in the space of linear conserved quantities. This can be interpreted as a dimension reduction of the multistationarity problem. We propose a generalization of the notion of toricity, compatible with our above remarks, in terms of the geometry of the variety instead of the syntactic shape of generators of the ideal. We consider 129 models from the BioModels repository [67], for which ODEbase [2] provides input data directly usable for symbolic computation. While the existing literature was mostly limited to the complex numbers, we use real quantifier elimination methods to treat also the real case, which is clearly the relevant domain from a scientific point of view. In practice, our real computations in Redlog [4] can compete with our complex ones. In theory we show that our real algorithms are in EXPTIME while Gröbner bases, which are typically used when working with ideal generators, are EXPSPACE-complete [68]. To our knowledge, this is the first time that such a comprehensive set of biomodels has been systematically processed using symbolic methods.

[2] http://odebase.cs.uni-bonn.de/

7.1.10. Towards an Improved Encoding of TLA+ Proof Obligations

We reconsider the encoding of proof obligations that arise in proofs about TLA$^+$ specifications in multi-sorted first-order logic, and specifically their translations to SMT solvers. Our previous work [69] relied on type inference for identifying expressions having atomic types such as integers but did not exploit more complex types, even if such types were constructed during type inference. A more pervasive use of types for translating set-theoretic expressions to the input language of SMT solvers appears promising in order to reduce the use of type injections and quantifiers and thus simplify the proof obligations passed to the solver, but it raises non-trivial soundness and completeness issues. Techniques of gradual typing designed for programming languages where type inference is not fully possible statically may be helpful in this context. A related problem is support for instantiation hints for quantified formulas given by the user. A first paper will be presented at JFLA 2020.

7.1.11. Formal Proofs of Tarjan's Algorithm

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

We consider Tarjan's classical algorithm for computing strongly connected components in a graph as a case study of intermediate complexity for comparing interactive proof assistants. Representing the algorithm as a functional program (rather than its more conventional imperative representation), we proved its correctness in three different proof assistants (Coq, Isabelle/HOL, and Why3). The proofs are based on essentially the same formulation of the algorithm and of its invariants, allowing us to compare differences due to idiosyncracies of the proof assistants, such as their ability to handle mutually recursive function definitions, proving termination beyond syntactic criteria, and their degree of automation. Our results were presented at ITP 2019 [33].

7.1.12. Implementation of an Efficient Validation of FOLID Cyclic Induction Reasoning

Checking the soundness of cyclic induction reasoning for first-order logic with inductive definitions ($\mathrm{FOL}_{\mathrm{ID}}$) is decidable but the standard checking method is based on an exponential complement operation for Büchi automata. We devised a polynomial method "semi-deciding" this problem; its most expensive steps are reminiscent of the comparisons with multiset path orderings. In practice, it has been integrated in the CYCLIST prover and successfully checked all the proofs included in its distribution. The work was presented at the CiSS2019 conference (Circularity in Syntax and Semantics) and the software is available at https://members.loria.fr/SStratulat/files/e-cyclist.zip.

7.2. Formal Methods for Developing and Analyzing Algorithms and Systems

Participants: Étienne André, Marie Duflot-Kremer, Yann Duplouy, Margaux Duroeulx, Igor Konnov, Dominique Méry, Stephan Merz, Nicolas Schnepf, Christoph Weidenbach.

7.2.1. Synthesis of Security Chains for Software Defined Networks

Joint work with Rémi Badonnel and Abdelkader Lahmadi (Inria Nancy – Grand Est, Resist).

The PhD thesis of Nicolas Schnepf focuses on applying techniques based on formal methods in the area of network communications, and in particular for the construction, verification, and optimization of chains of security functions in the setting of software-defined networks (SDN). The main objective is to prevent applications from disrupting the functioning of the network or services, for example by launching denial of service attacks, port scanning or similar activities.

We designed techniques for formally verifying security chains using SMT solving and symbolic model checking. Furthermore, we developed and prototypically implemented an approach for (i) learning a Markov chain characterizing the network behavior of an Android application based on its observed communications, (ii) inferring appropriate security functions from the structure of that Markov chain and thresholds set by the network operator, using techniques of logic programming, (iii) combining security functions for individual applications into larger security chains, and (iv) optimizing the deployment of security chains for a given SDN infrastructure using techniques of (linear or non-linear) optimization or optimizing SMT solvers. Two papers were presented at IM 2019 [39], [38], the PhD thesis [12] was defended in September 2019, and a journal paper is in preparation.

7.2.2. Satisfiability Techniques for Reliability Assessment

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

In the context of the PhD thesis of Margaux Durœulx, funded by the Lorraine University of Excellence program, we explore the applicability of satisfiability techniques for assessing the reliability of complex systems. In particular, we consider component-based systems modeled using fault trees that can be seen as a visual representation of the structure function indicating which combinations of component failures lead to system failures. We rely on SAT solvers to compute minimal tie sets, i.e., minimal sets of components whose functioning ensures that the overall system works. These tie sets are instrumental for a probabilistic reliability assessment. In 2019, we have extended this idea to dynamic fault trees where the order of component failures needs to be taken into account in order to determine the failure status of the overall system [34].

7.2.3. Statistical Model Checking of Distributed Programs

Yann Duplouy joined the HAC SPECIS project (cf. section 9.2) in December 2018 as a post-doctoral researcher with the objective of designing and implementing a statistical model checker within the SimGrid framework. So far he added to SimGrid the possibility to use stochastic profiles, introducing probabilities in the model of the network. He also developed a prototype tool that can be interfaced with the SimGrid simulators to perform statistical model checking on the actual programs simulated using the SimGrid framework. He now validates this prototype on concrete case studies, including the Bit Torrent protocol with probabilistic failures of the nodes.

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

Joint work with Nathalie Bertrand (Inria Rennes Bretagne – Atlantique, SUMO), Marijana Lazić (TU Munich) and 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 [62], [63], [64].

Our previous results apply to asynchronous algorithms. It is well-known that distributed consensus cannot be solved in purely asynchronous systems [61]. However, when an algorithm is provided with a random coin, consensus becomes solvable (e.g., the algorithm by Ben-Or, 1993). In [29], 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 and randomized one-step consensus (RS-BOSCO [71]). The verification of the latter algorithm required us to run experiments in Grid5000. This paper was presented at CONCUR 2019.

Another way of making consensus solvable is to impose synchrony on the executions of a distributed system. In [40] 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. This paper was presented at TACAS 2019. The paper was invited to the special issue of TACAS 2019, to appear in the *International Journal on Software Tools for Technology Transfer* in 2020.

7.2.5. 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 [66]. 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.

Following our results in 2018 [65], we have extended the symbolic model checker for TLA$^+$. In [22], we have defined the translation process from TLA$^+$ to SMT as a series of rewriting rules. Furthermore, we have proven soundness of this translation. Our experiments show that APALACHE runs faster than TLC when proving inductive invariants. APALACHE also implements bounded model checking, which has to be improved, in order to make it competitive with TLC. The paper [22] was presented at ACM OOPSLA 2019.

7.2.6. Incremental Development of Systems and Algorithms

Joint work with Rosemary Monahan (NUI Maynooth, Ireland) and Mohammed Mosbah (LaBRI, Bordeaux).

The development of distributed algorithms and, more generally, of distributed systems, is a complex, delicate, and challenging process. The approach based on refinement applies a design methodology that starts from the most abstract model and leads, in an incremental way, to a distributed solution. The use of a proof assistant gives a formal guarantee on the conformance of each refinement with the model preceding it. Our main result during 2019 is the development of a distributed pattern [26] handling the dynamicity of the topology of networks.

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contracts with Industry

8.1.1. Logic4Business

The Max Planck Institute for Informatics (MPI-INF) and Logic 4 Business GmbH (L4B) have signed a cooperation contract. Its subject is the application of automated reasoning methods to product complexity management, in particular in the car industry. MPI-INF is providing software and know-how, L4B is providing real-world challenges.

9. Partnerships and Cooperations

9.1. Regional Initiatives

Antoine Defourné's PhD thesis and Yann Duplouy's post-doctoral research are co-funded by Région Grand Est.

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

9.2. National Initiatives

9.2.1. PIA2 ISITE LUE

Project acronym: ISITE LUE - Digitrust

Project title: Lorraine Université d'Excellence, Citizen Trust in the Digital World

Duration: 2016 – 2020

Coordinator: Marine Minier

Participants: Margaux Durœulx, Stephan Merz

Abstract: Digitrust is one of the "impact" projects within the excellence funding acquired by University of Lorraine and supports research into different aspects related to the trustworthiness and security of digital systems. It funds the PhD thesis of Margaux Durœulx on the use of SAT techniques for assessing system reliability.

9.2.2. ANR International Project ProMiS

Project acronym: ProMiS.

Project title: Provable Mitigation of Side Channel through Parametric Verification

Duration: November 2019 – April 2022.

Coordinators: Étienne André and Jun Sun (Singapore Management University, Singapore).

Other partners: École Centrale Nantes, Singapore University of Technology and Design.

Participants: Étienne André.

Abstract: ProMiS is an international project, funded by ANR in France and by NRF in Singapore under the PRCI program.

The Spectre vulnerability has recently been reported, which affects most modern processors. The idea is that attackers can extract information about the private data using a timing attack. It is an example of side channel attacks, where secure information flows through side channels unintentionally. How to systematically mitigate such attacks is an important and yet challenging research problem.

We propose to automatically synthesize mitigation of side channel attacks (e.g., timing or cache) using well-developed verification techniques. The idea is to reduce this problem to the parameter synthesis problem of a given formalism (for instance, parametric timed automata). Given a program or system with design parameters which can be tuned to mitigate side channel attacks, our approach will automatically generate provably secure valuations of the parameters. We plan to deliver a toolkit which can be automatically applied to real-world systems.

9.2.3. 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, Hamid Rahkooy.

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/.

9.2.4. 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, Horatiu Cirstea.

Abstract: During the last 30 years, the aerospace domain has successfully devised rigorous methods and tools for the development of safe functionally-correct software. During this process, interactive software has received a relatively lower amount of attention. However, Human-System Interactions (HSI) are important for critical systems and especially in aeronautics: 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.

9.2.5. 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), until February 2019; Dominique Méry, since March 2019.

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

Participants: Dominique Méry, Zheng Cheng.

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 and models of the relevant physical laws. The systems are generally characterized by differential equations with solutions in continuous domains; discretization steps are therefore of particular importance for assessing the correctness of CPSs. 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 step-wise 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/.

9.2.6. ANR Project PARDI

Project acronym: PARDI.

Project title: Verification of parameterized distributed systems.

Duration: January 2017 – December 2021.

Coordinator: Philippe Quéinnec (ENSEEIHT/IRIT Toulouse).

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

Participants: 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/.

9.2.7. 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 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.

9.2.8. DFG Transregional Research Center 248 CPEC

Project acronym: CPEC.

Project title: Foundations of Perspicuous Software Systems.

Duration: January 2019 – December 2022.

Coordinators: Holger Hermanns (Saarland University, Germany) and Raimund Dachselt (University of Dresden, Germany).

Other partners: Max Planck Institute for Software Systems, Saarbrücken.

Participants: Alberto Fiori, Sophie Tourret, Christoph Weidenbach.

Abstract: With cyber-physical technology increasingly impacting our lives, it is very important to ensure that humans can understand them. Systems lack support for making their behaviour plausible to their users. And even for technology experts it is nowadays virtually impossible to provide scientifically well-founded answers to questions about the exact reasons that lead to a particular decision, or about the responsibility for a malfunctioning. The root cause of the problem is that contemporary systems do not have any built-in concepts to explicate their behaviour. They calculate and propagate outcomes of computations, but are not designed to provide explanations. They are not perspicuous. The key to enable comprehension in a cyber-physical world is a science of perspicuous computing.

More information: https://www.perspicuous-computing.science/.

9.3. European Initiatives

9.3.1. FP7 & H2020 Projects

9.3.1.1. ERC Matryoshka

Program: ERC.

Project acronym: Matryoshka.

Duration: April 2017 – March 2022.

Coordinator: Jasmin Blanchette (VU Amsterdam).

Participants: Antoine Defourné, Daniel El Oraoui, Mathias Fleury, Pascal Fontaine, Stephan Merz, 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. The purpose of Matryoshka is 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 is 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/.

9.3.2. Collaborations in European Programs, Except FP7 & H2020

Program: Erasmus+.

Project acronym: PIAF.

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

Program: ERASMUS+.

Project acronym: ARC.

Project title: Automated reasoning in the class.

Coordinator: West University of Timisoara (Romania).

Other partners: Johaness Kepler University Linz (Austria), RWTH Aachen University (Germany), Eszterhazy Karoly University (Hungary), Université de Lorraine.

Participant: Sorin Stratulat.

Abstract: The main objective of the project is to improve the education of computer science students in fields related to computational logic, by creating innovative and advanced learning material that uses automated reasoning and by training a large number of academic staff in using this in a modern way. Thus indirectly the project objectives include the effects of increased software reliability: virus elimination, online safety, better detection of negative online phenomena (fake news, cyberbullying, etc.), and other.

9.4. International Research Visitors

9.4.1. Visits of International Scientists

Maria Paola Bonacina.

Date: 11 February 2019 – 16 February 2019.

Institution: Università degli Studi di Verona, Italy.

Host: Pascal Fontaine.

Maria Paola Bonacina is a professor at the Università degli Studi di Verona, Italy. She is well known in the community for her numerous works in the field of automated reasoning, notably in SMT, combination of theories, and procedures for first-order logic. During her one-week stay in Nancy, we particularly discussed SGGS (semantically-guided goal-sensitive theorem proving) as a means of inspiration for instantiation in SMT. We also worked on a review paper on combination of theories, published in 2019 [49].

Armin Biere.

Date: 27 May 2019 – 29 May 2019.

Institution: Johannes Keppler Universität, Linz, Austria.

Host: Christoph Weidenbach.

Armin Biere is professor at the University of Linz. He is a leading researcher in the SAT community. During his stay we discussed recent developments in SAT solving. In particular, resolution based inference and reduction mechanisms beyond subsumption resolution.

9.4.1.1. Internships

Manon Blanc

Date: 1 June 2019 – 31 July 2019

Institution: ENS Cachan

Host: Pascal Fontaine

In her bachelor thesis, Manon Blanc studied and experimentally evaluated two different subtropical methods for handling polynomial constraints within SMT.

Mehran Aghabozorgi

Date: 5 August 2019 – 7 October 2019

Institution: Isfahan University of Technology, Iran

Host: Christoph Weidenbach

Mehran worked on algorithms enhancing SAT pre- and inprocessing. He implemented blocked clause elimination as well as a variable elimination algorithm aiming at smaller clause sets.

9.4.2. Visits to International Teams

9.4.2.1. Research Stays Abroad

Thomas Sturm visited the University of Bonn (Institute of Computer Science II) for 4 weeks during 2019, and the University of Kassel (Mathematical Institute). Topics included perspectives for SMT Solving in symbolic reaction network analysis, toricity of steady state varieties, scaling methods for systems of ordinary differential equations (ODE), and logic approaches for the classification of real singularities of ODE.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Organization of Scientific Events

10.1.1.1. Membership in Organizing Committees

Jasmin Blanchette co-organized the *2nd Deduction Mentoring Workshop* (DeMent 2019). He also coorganized the *VeriDis Group Retreat + Second European Workshop on Higher-Order Automated Reasoning* (Matryoshka 2019).

Pascal Fontaine co-organized the Third Workshop on Mathematical Logic and its Applications, in Nancy.

Igor Konnov and Stephan Merz were organizers of the sixth *Workshop on Formal Reasoning in Distributed Algorithms* (FRIDA 2019), colocated with DISC 2019 in Budapest, Hungary.

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 2019, VTSA took place in August in Esch sur Alzette, Luxembourg.

10.1.2. Program Committees

10.1.2.1. Chair of Conference Program Committees

Jasmin Blanchette co-chaired the program committee of the *9th ACM SIGPLAN International Conference on Certified Programs and Proofs* (CPP 2020).

Pascal Fontaine served as the chair of the *Conference on Automated Deduction* (CADE-27).

Dominique Méry co-chaired the program committee of the *13th International Symposium on Theoretical Aspects of Software Engineering* (TASE 2019).

Stephan Merz was Tool Exhibition Chair at the *3rd World Congress for Formal Methods* (FM Week 2019).

10.1.2.2. Membership in Conference Program Committees

Jasmin Blanchette served on the program committees of the *27th International Conference on Automated Deduction* (CADE-27), *19th Conference on Formal Methods in Computer-Aided Design* (FM-CAD 2019), *21st International Symposium on Principles and Practice of Declarative Programming* (PPDP 2019), *12th International Symposium on Frontiers of Combining Systems* (FroCoS 2019), *26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (TACAS 2020), and *4th Conference on Artificial Intelligence and Theorem Proving* (AITP 2019). He also served on the following workshop committees: *Second Workshop on Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements* (ARCADE 2019) and *Deduktionstreffen 2019*.

Pascal Fontaine served on the program committees of the *International Symposium on Frontiers of Combining Systems* (FroCoS 2019), the *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (TABLEAUX 2019), and the *International Conference on Theory and Applications of Satisfiability Testing* (SAT 2019). He also served on the committee of *Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements* (ARCADE 2019).

Dominique Méry served on the program committees of the *16th International Colloquium on Theoretical Aspects of Computing* (ICTAC 2019), the *8th Workshop on Formal Methods for Interactive Systems* (FMIS 2019), the *Workshop on Practical Formal Verification for Software Dependability* (AFFORD'19), the *24th International Conference on Engineering of Complex Computer Systems* (ICECCS 2019), the *Workshop on Formal Methods for Autonomous Systems* (FMAS 2019), the *23rd International Symposium on Formal Methods* (FM 2019), the *Workshop on Models and Data Engineering for Cyber-Physical Systems* (CPS@MEDI 2019), the *Workshop on Formal Models for Mastering Heterogeneous Multifaceted Systems* (REMEDY 2019), the *3rd Workshop on Formal Approaches for Advanced Computing Systems* (FAACS 2019), the *21st International Conference on Formal Engineering Methods* (ICFEM 2019), and the *9th International Conference on Model and Data Engineering* (MEDI 2019).

Stephan Merz served on the program committees of the *19th International Workshop on Automated Verification of Critical Systems* (AVoCS 2019), the *Doctoral Symposium, Formal Methods* (DS-FM 2019), the *5th International Workshop on Formal Integrated Development Environment* (F-IDE 2019), and the *21st International Conference on Formal Engineering Methods* (ICFEM 2019).

Sorin Stratulat served on the program committees of the *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (SYNASC 2019), the *International Conference on Information Assurance and Security* (IAS 2019), and the *International Conference on Computational Intelligence in Security for Information Systems* (CISIS 2019).

Thomas Sturm served on the program committees of the *21st International Workshop on Computer Algebra in Scientific Computing* (CASC 2019) and the *4th International Workshop on Satisfiability Checking and Symbolic Computation* (SC-SQUARE 2019).

Uwe Waldmann served on the program committees of the *27th International Conference on Automated Deduction* (CADE-27) and the *International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (TABLEAUX 2019).

Christoph Weidenbach served on the program committees of the *International Conference on Automated Deduction* (CADE 27), and the *International Symposium on Frontiers of Combining Systems* (FroCoS 2019). He also served on the committee of *Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements* (ARCADE 2019).

10.1.3. Journals

10.1.3.1. 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* [16].

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

Thomas Sturm has been an editor of the *Journal of Symbolic Computation* (Elsevier) since 2003 and an editor of *Mathematics in Computer Science* (Springer) since 2013. He edited a special issue of the Journal of Symbolic Computation on *Symbolic Computation and Satisfiability Checking*.

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

10.1.4. Invited Talks

Jasmin Blanchette gave a keynote talk at the *8th ACM SIGPLAN International Conference on Certified Programs and Proofs* (CPP 2019).

Dominique Méry was an invited speaker at the *16th International Colloquium on Theoretical Aspects of Computing* (ICTAC 2019).

Stephan Merz was invited to the meeting of IFIP Working Group 2.3 in October 2019 in Los Altos, California. He was an invited speaker at the meeting of the AFSEC group of GDR GPL in December 2019 in Toulouse.

Sorin Stratulat was an invited speaker at FROM 2019 (Working Formal Methods Symposium) in Timisoara, Romania, where he presented an efficient way to validate cyclic pre-proofs for first-order logic with inductive definitions.

Christoph Weidenbach was invited to give a lecture at the SAT/SMT/AR summer school 2019 in Lisbon, Portugal. He was invited to present at the computer science lecture series at the University of Bonn, Germany.

10.1.5. Leadership within the Scientific Community

Jasmin Blanchette served as 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.

Pascal Fontaine is an SMT-LIB manager, together with Clark Barrett (Stanford University) and Cesare Tinelli (University of Iowa). He was a regular member of the steering committee for the FroCoS (*Frontiers of Combining Systems*) conference series until September 2019, and he is a member of the steering committee for the SC-Square (*Satisfiability Checking and Symbolic Computation* workshop series. He was ex-officio member of the CADE (*Conference on Automated Deduction*) Inc. Board of Trustees until July 2019. 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* and a member of the steering committee of the workshop on Automated Verification of Critical Systems (AVoCS).

Thomas Sturm has been elected the chair 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.

10.1.6. Scientific Expertise

Dominique Méry was a member of the committee for the thesis award of GDR GPL (*Génie de la Programmation et du Logiciel*) 2019.

Stephan Merz contributed an assessment of candidates for a professorship at TU Vienna, Austria.

Thomas Sturm served as an external expert on the appointment committee for a professorship in Computer Algebra at the University of Kassel, Germany. He is taking an advisory role as a “project partner” in the UK EPSRC Project EP/R019622/1 *Embedding Machine Learning within Quantifier Elimination Procedures*.

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

10.1.7. Research Administration

Dominique Méry participated in the evaluation committees of HCERES for the LIASD (*Laboratoire d'informatique avancée de Saint-Denis*) and the doctoral school of École Polytechnique.

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 2019, he was a member of the hiring committees of senior researchers at Inria and of junior researchers at Inria Rennes Bretagne Atlantique. 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 coordinates the scientific affairs at MPI-INF.

10.2. Teaching, Supervision, Thesis Committees

10.2.1. Teaching

Licence : Marie Duflot-Kremer, Algorithmique et Programmation 1, 60 HETD, L1, Université de Lorraine, France

Licence : Marie Duflot-Kremer, Algorithmique et Programmation 2, 10 HETD, L1, Université de Lorraine, France

Diplôme inter universitaire : Marie Duflot-Kremer, formation d'enseignants du secondaire à la spécialité NSI, 43 HETD, Université de Lorraine, France

Licence : Marie Duflot-Kremer, Introduction au Web, 20 HETD, L1, Université de Lorraine, France

Licence : Marie Duflot-Kremer, Accompagnement Algorithmique 1, 26 HETD, L1, Université de Lorraine, France

Licence : Marie Duflot-Kremer, Programmation Web, 5 HETD, L3, 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, Algorithmes distribués, 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.

Licence : Sorin Stratulat, Algorithmique des structures de contrôle, 45 HETD, L1 Informatique, ISFATES, Université de Lorraine, France.

Licence : Sorin Stratulat, Algorithmique des structures de données, 45 HETD, L1 Informatique, ISFATES, Université de Lorraine, France.

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

Master: Sorin Stratulat, Génie Logiciel, 20 HETD, M2 Informatique, Université de Lorraine, France.

Master: Sophie Tourret, Concrete Semantics with Isabelle/HOL, 6 ECTS, Saarland University, Germany.

Master: Uwe Waldmann, Automated Reasoning, 9 ECTS, Saarland University, Germany.

Master: Christoph Weidenbach, Decision Procedures, 6 ECTS, Saarland University, Germany.

10.2.2. Supervision

PhD: Martin Bromberger, Decision Procedures for Linear Arithmetic. Saarland University, 10 December 2019. Supervised by Thomas Sturm and Christoph Weidenbach.

PhD: Nicolas Schnepf, Orchestration et vérification de fonctions de sécurité pour des environnements intelligents. Université de Lorraine, 30 September 2019. Supervised by Rémi Badonnel, Abdelkader Lahmadi, and Stephan Merz.

PhD: Marco Voigt, Decidable Fragments of First-Order Logic and of First-Order Linear Arithmetic with Uninterpreted Predicates. Saarland University, 31 July 2019. Supervised by Thomas Sturm and Christoph Weidenbach.

PhD in progress: Antoine Defourné, SMT for TLAPS, Université de Lorraine. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, since March 2019.

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: Alberto Fiori, Clause Learning from Simple Models, Saarland University. Supervised by Christoph Weidenbach, since August 2018.

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

PhD in progress: Pierre Lermusiaux, Analysis of properties of interactive critical systems, Université de Lorraine. Supervised by Horatiu Cirstea and Pierre-Etienne Moreau, since October 2017.

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

10.2.3. Thesis Committees

Pascal Fontaine served as a reviewer in the thesis committees for Martin Bromberger (Universität des Saarlandes, Germany), Albin Coquereau (Université Paris-Saclay, France), Hakan Metin (Sorbonne Université, France), Simon Robillard (Chalmers University of Technology, Sweden) and Stefano Varotti (University of Trento, Italy).

Stephan Merz served as an examiner in the PhD committee of The Anh Pham (ENS Rennes) and as the president in the PhD committee of Renaud Vilmart (Univ. of Lorraine). He was a member of the PhD committee of Nicolas Schnepf as the thesis advisor.

Thomas Sturm served as a reviewer in the thesis committees for Martin Bromberger and Marco Voigt (Saarland University, Germany).

Christoph Weidenbach served as a reviewer in the thesis committees for Martin Bromberger and Marco Voigt (Saarland University, Germany).

10.3. Popularization

10.3.1. Responsibilities at Inria or Beyond

- Marie Duflot-Kremer is the deputy vice-president for outreach activities in the supervisory council of SIF (*Société Informatique de France*) and a member of the scientific committee of *Fondation Blaise Pascal*.
- Christoph Weidenbach is a member of the advisory committee for the German computer science competitions for pupils. Together with his colleagues at Saarland university he organizes the “Computer Science Research Days” for the most talented high-school students out of the competition every year. In addition, he organizes the final training for the German Informatics Olympiad team and coaches the Saarland University student teams for the ICPC.

10.3.2. Articles and Contents

Marie Duflot-Kremer is a member of the ERASMUS+ project PIAF (cf. section 9.3) with collleagues from Liège, Luxembourg and Saarbrücken. This projects aims at studying how computational thinking can be introduced in primary education (with kids rangng from 5 to 12 years old). The goal is first to agree on a shared reference document on computational thinking competences, and then to produce and test educational scenarios and didactical resources. So far this year the reference document has been designed (and a paper has been submitted recently to a conference). The work on the scenarios and the resources is ongoing.

10.3.3. Education

Marie Duflot-Kremer intervenes in the training of teachers:

- for primary school, half a day for training teachers on how to add a bit of computer science in their teaching;
- for secondary school, she took part (43 hours in 2019) in the advanced training of high school teachers who deliver a newly introduced computer science course (Numérique et Sciences Informatiques), and one day training for math teachers about unplugged activities;
- half a day of training for members of INSPE (Institut National Supérieur du Professorat et de l'Education), the people in charge of training teachers.

10.3.4. Interventions

Marie Duflot-Kremer takes part every year in several events, including local ones such as *Fête de la Science* (for which she trains 3rd year students to handle the workshops and gave this year a talk/show on "informagics"), Pint of Science, a talk for the local phase of the *Tournoi Français des Jeunes Mathématiciennes et Mathématiciens* and the local NSI (*Numérique et Sciences Informatiques*) day for secondary school teachers.

She is also invited for workshops and talks in events outside of the Nancy region, like national or regional days of APMEP (*Association des Professeurs de Mathématiques de l'Enseignement Public*) in Dijon and Grenoble, the SETT conference in Namur or the video game creation competition for kids in Manosque.

10.3.5. Creation of Media or Tools for Science Outreach

As a member of the national group *Informatique Sans Ordinateur* (ISO), Marie Duflot-Kremer takes part in creating new popularization activities and publishing online documentation to help people reproduce unplugged computer science activities. She also proposed and supervised a project of master students in cognitive sciences who created an escape game presenting various computer science concepts to kids from 13 years old. The documentation is available on her webpage. [4]

11. Bibliography

Major publications by the team in recent years

[1] T. Bouton, D. C. B. de Oliveira, D. Déharbe, P. Fontaine. *veriT: an open, trustable and efficient SMT-solver*, in "Proc. Conference on Automated Deduction (CADE)", Montreal, Canada, R. Schmidt (editor), Lecture Notes in Computer Science, Springer, 2009, vol. 5663, pp. 151-156

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

[4] https://members.loria.fr/MDuflot/.

[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] M. BROMBERGER. *Decision Procedures for Linear Arithmetic*, Saarland University, December 2019, https://hal.inria.fr/tel-02427371

[12] N. SCHNEPF. *Orchestration and verification of security functions for smart devices*, Université de Lorraine, September 2019, https://hal.univ-lorraine.fr/tel-02351769

[13] M. VOIGT. *Decidable Fragments of First-Order Logic and of First-Order Linear Arithmetic with Uninterpreted Predicates*, Universität des Saarlandes, July 2019, https://hal.inria.fr/tel-02406821

Articles in International Peer-Reviewed Journals

[14] A. BENTKAMP, J. C. BLANCHETTE, D. KLAKOW. *A Formal Proof of the Expressiveness of Deep Learning*, in "Journal of Automated Reasoning", August 2019, vol. 63, n^{o} 2, pp. 347-368 [*DOI :* 10.1007/s10817-018-9481-5], https://hal.inria.fr/hal-02296014

[15] J. C. Blanchette, L. Gheri, A. Popescu, D. Traytel. *Bindings as Bounded Natural Functors*, in "Proceedings of the ACM on Programming Languages", January 2019, vol. 3, n^o POPL, pp. 1-34 [*DOI :* 10.1145/3290335], https://hal.archives-ouvertes.fr/hal-01989726

[16] J. C. Blanchette, S. Merz. *Selected Extended Papers of ITP 2016: Preface*, in "Journal of Automated Reasoning", February 2019, vol. 62, n^o 2, pp. 169-170 [*DOI :* 10.1007/s10817-018-9470-8], https://hal.inria.fr/hal-02395177

[17] R. Bradford, J. H. Davenport, M. England, H. Errami, V. Gerdt, D. Grigoriev, C. Hoyt, M. Košta, O. Radulescu, T. Sturm, A. Weber. *Identifying the parametric occurrence of multiple steady states for some biological networks*, in "Journal of Symbolic Computation", May 2020, vol. 98, pp. 84-119 [*DOI :* 10.1016/j.jsc.2019.07.008], https://hal.inria.fr/hal-02397154

[18] M. Bromberger, T. Sturm, C. Weidenbach. *A complete and terminating approach to linear integer solving*, in "Journal of Symbolic Computation", July 2019, forthcoming [*DOI :* 10.1016/j.jsc.2019.07.021], https://hal.inria.fr/hal-02397168

[19] P. Chocron, P. Fontaine, C. Ringeissen. *Politeness and Combination Methods for Theories with Bridging Functions*, in "Journal of Automated Reasoning", 2019, forthcoming [*DOI :* 10.1007/s10817-019-09512-4], https://hal.inria.fr/hal-01988452

[20] 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, pp. 3-41 [*DOI :* 10.1016/j.jsc.2018.04.002], https://hal.archives-ouvertes.fr/hal-01590654

[21] I. Konnov. *Handbook of Model Checking by Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem (eds), published by Springer International Publishing AG, Cham, Switzerland, 2018*, in "Formal Aspects of Computing", 2019, pp. 455-456 [*DOI :* 10.1007/s00165-019-00486-z], https://hal.inria.fr/hal-02398334

[22] I. Konnov, J. Kukovec, T.-H. Tran. *TLA+ Model Checking Made Symbolic*, in "Proceedings of the ACM on Programming Languages", 2019, vol. 3, n^o OOPSLA, pp. 123:1–123:30 [*DOI :* 10.1145/3360549], https://hal.archives-ouvertes.fr/hal-02280888

[23] M. Romero, M. Duflot, T. Viéville. *The robot game : analysis of a computational thinking unplugged activity under the perspective of embodied cognition*, in "Review of science, mathematics and ICT education", June 2019, vol. 13, n^o 1 [*DOI :* 10.26220/rev.3089], https://hal.inria.fr/hal-02144467

[24] N. Schnepf, R. Badonnel, A. Lahmadi, S. Merz. *Rule-Based Synthesis of Chains of Security Functions for Software-Defined Networks*, in "Electronic Communications of the EASST", 2019, vol. 076, https://hal.inria.fr/hal-02397981

Articles in Non Peer-Reviewed Journals

[25] J. H. Davenport, M. England, A. Griggio, T. Sturm, C. Tinelli. *Symbolic computation and satisfiability checking*, in "Journal of Symbolic Computation", July 2019, Invited Editorial, forthcoming [*DOI :* 10.1016/j.jsc.2019.07.017], https://hal.inria.fr/hal-02397190

Invited Conferences

[26] D. MÉRY. *Verification by Construction of Distributed Algorithms*, in "Theoretical Aspects of Computing - ICTAC 2019 - 16th International Colloquium", Mammamet, Tunisia, R. M. HIERONS, M. MOSBAH (editors), Theoretical Aspects of Computing - ICTAC 2019 - 16th International Colloquium, Hammamet, Tunisia, October 31 - November 4, 2019, Proceedings, Springer, October 2019, n^{o} 11884, pp. 22-38 [*DOI :* 10.1007/978-3-030-32505-3_2], https://hal.inria.fr/hal-02400379

International Conferences with Proceedings

[27] H. BARBOSA, A. REYNOLDS, D. EL OURAOUI, C. TINELLI, C. BARRETT. *Extending SMT Solvers to Higher-Order Logic*, in "CADE-27", Natal, Brazil, Lecture Notes in Computer Science, Springer, August 2019, vol. 11716, pp. 35-54 [*DOI :* 10.1007/978-3-030-29436-6_3], https://hal.archives-ouvertes.fr/hal-02300986

[28] A. BENTKAMP, J. C. BLANCHETTE, S. TOURRET, P. VUKMIROVIĆ, U. WALDMANN. *Superposition with Lambdas*, in "CADE-27", Natal, Brazil, August 2019, pp. 55-73 [*DOI :* 10.1007/978-3-030-29436-6_4], https://hal.inria.fr/hal-02296038

[29] N. BERTRAND, I. KONNOV, M. LAZIC, J. WIDDER. *Verification of Randomized Consensus Algorithms under Round-Rigid Adversaries*, in "CONCUR 2019 - 30th International Conference on Concurrency Theory", Amsterdam, Netherlands, August 2019, pp. 1-16 [*DOI :* 10.4230/LIPICS.CONCUR.2019.33], https://hal.inria.fr/hal-02191348

[30] 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, 2019 [*DOI :* 10.1145/3293880.3294087], https://hal.archives-ouvertes.fr/hal-01937136

[31] *Best Paper*
M. BROMBERGER, M. FLEURY, S. SCHWARZ, C. WEIDENBACH. *SPASS-SATT: A CDCL(LA) Solver*, in "27th International Conference on Automated Deduction (CADE-27)", Natal, Brazil, P. FONTAINE (editor), Lecture Notes in Computer Science, 2019, vol. 11716, pp. 111-122 [*DOI :* 10.1007/978-3-030-29436-6_7], https://hal.inria.fr/hal-02405524.

[32] G. BUSANA, B. DENIS, M. DUFLOT, S. HIGUET, L. KATAJA, Y. KREIS, C. LADURON, C. MEYERS, Y. PARMENTIER, R. REUTER, A. WEINBERGER. *PIAF : développer la Pensée Informatique et Algorithmique dans l'enseignement Fondamental*, in "Didapro 8 - DIDASTIC - L'informatique, objets d'enseignements – enjeux épistémologiques, didactiques et de formation", Lille, France, Actes du 8e colloque international francophone sur la didactique de l'informatique (Didapro 8 - DIDASTIC), 2020, Session poster, https://hal.archives-ouvertes.fr/hal-02424418

[33] R. CHEN, C. COHEN, J.-J. LEVY, S. MERZ, L. THÉRY. *Formal Proofs of Tarjan's Strongly Connected Components Algorithm in Why3, Coq and Isabelle*, in "ITP 2019 - 10th International Conference on Interactive Theorem Proving", Portland, United States, J. HARRISON, J. O'LEARY, A. TOLMACH (editors), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019, vol. 141, pp. 13:1 - 13:19 [*DOI :* 10.4230/LIPICS.ITP.2019.13], https://hal.inria.fr/hal-02303987

[34] M. DUROEULX, N. BRINZEI, M. DUFLOT, S. MERZ. *Integrating satisfiability solving in the assessment of system reliability modeled by dynamic fault trees*, in "29th European Safety and Reliability Conference,

ESREL 2019", Hannover, Germany, Research Publishing Services, September 2019 [*DOI :* 10.3850/981-973-0000-00-0], https://hal.inria.fr/hal-02262205

[35] A. Fiori, C. Weidenbach. *SCL: Clause Learning from Simple Models*, in "27th International Conference on Automated Deduction", Natal, Brazil, P. Fontaine (editor), Lecture Notes in Computer Science, 2019, vol. 11716, pp. 233-249 [*DOI :* 10.1007/978-3-030-29436-6_14], https://hal.inria.fr/hal-02405550

[36] M. Fleury, H.-J. Schurr. *Reconstructing veriT Proofs in Isabelle/HOL*, in "PxTP 2019 - Sixth Workshop on Proof eXchange for Theorem Proving", Natal, Brazil, August 2019, vol. 301, pp. 36-50, https://arxiv.org/abs/1908.09480 [*DOI :* 10.4204/EPTCS.301.6], https://hal.inria.fr/hal-02276530

[37] 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

[38] N. Schnepf, R. Badonnel, A. Lahmadi, S. Merz. *A Tool Suite for the Automated Synthesis of Security Function Chains*, in "IFIP/IEEE IM 2019 - IFIP/IEEE International Symposium on Integrated Network Management", Washington, United States, April 2019, https://hal.inria.fr/hal-02111658

[39] N. Schnepf, R. Badonnel, A. Lahmadi, S. Merz. *Automated Factorization of Security Chains in Software-Defined Networks*, in "IFIP/IEEE IM 2019 - IFIP/IEEE International Symposium on Integrated Network Management", Washington, United States, April 2019, https://hal.inria.fr/hal-02111656

[40] I. Stoilkovska, I. Konnov, J. Widder, F. Zuleger. *Verifying Safety of Synchronous Fault-Tolerant Algorithms by Bounded Model Checking*, in "TACAS 2019 - International Conference on Tools and Algorithms for the Construction and Analysis of Systems", Prague, Czech Republic, April 2019 [*DOI :* 10.1007/978-3-030-17465-1_20], https://hal.inria.fr/hal-01925653

[41] A. Teucke, M. Voigt, C. Weidenbach. *On the Expressivity and Applicability of Model Representation Formalisms*, in "12th International Symposium on Frontiers of Combining Systems (FroCoS 2019)", London, United Kingdom, A. Herzig, A. Popescu (editors), Lecture Notes in Computer Science, Springer, 2019, vol. 11715, pp. 22-39 [*DOI :* 10.1007/978-3-030-29007-8_2], https://hal.inria.fr/hal-02406605

[42] P. Vukmirović, J. C. Blanchette, S. Cruanes, S. Schulz. *Extending a Brainiac Prover to Lambda-Free Higher-Order Logic*, in "TACAS 2019 - 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems", Prague, Czech Republic, April 2019, pp. 192-210, https://hal.inria.fr/hal-02178274

[43] P. Vukmirović, J. C. Blanchette, S. Cruanes, S. Schulz. *Faster, Higher, Stronger: E 2.3*, in "TACAS 2019", Prague, Czech Republic, LNAI, August 2019, vol. 11716, pp. 495-507 [*DOI :* 10.1007/978-3-030-29436-6_29], https://hal.inria.fr/hal-02296188

[44] C. Weidenbach. *The Challenge of Unifying Semantic and Syntactic Inference Restrictions*, in "2nd International Workshop on Automated Reasoning: Challenges, Applications, Directions, Exemplary Achievements (ARCADE 2019)", Natal, Brazil, Electronic Proceedings in Theoretical Computer Science, 2019, https://hal.inria.fr/hal-02406673

National Conferences with Proceedings

[45] P. LERMUSIAUX, H. CIRSTEA, P.-E. MOREAU. *Pattern eliminating transformations*, in "CIEL 2019 - 8ème Conférence en IngénieriE du Logiciel", Toulouse, France, June 2019, https://hal.inria.fr/hal-02186325

Conferences without Proceedings

[46] H. BARBOSA, J. C. BLANCHETTE, M. FLEURY, P. FONTAINE, H.-J. SCHURR. *Better SMT Proofs for Easier Reconstruction*, in "Conference on Artificial Intelligence and Theorem Proving (AITP 2019)", Obergurgl, Austria, April 2019, https://hal.archives-ouvertes.fr/hal-02381819

[47] J. C. BLANCHETTE, D. E. OURAOUI, P. FONTAINE, C. KALISZYK. *Machine Learning for Instance Selection in SMT Solving*, in "Conference on Artificial Intelligence and Theorem Proving (AITP 2019)", Obergurgl, Austria, April 2019, https://hal.archives-ouvertes.fr/hal-02381430

Scientific Books (or Scientific Book chapters)

[48] P. BAUMGARTNER, U. WALDMANN. *Hierarchic Superposition Revisited*, in "Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday", C. LUTZ, U. SATTLER, C. TINELLI, A.-Y. TURHAN, F. WOLTER (editors), Lecture Notes in Computer Science, Springer, 2019, vol. 11560, pp. 15-56 [*DOI :* 10.1007/978-3-030-22102-7_2], https://hal.inria.fr/hal-02402941

[49] M. P. BONACINA, P. FONTAINE, C. RINGEISSEN, C. TINELLI. *Theory Combination: Beyond Equality Sharing*, in "Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday", C. LUTZ, U. SATTLER, C. TINELLI, A.-Y. TURHAN, F. WOLTER (editors), Theoretical Computer Science and General Issues, Springer, June 2019, vol. 11560, pp. 57-89, https://hal.inria.fr/hal-02194001

[50] S. MERZ. *Formal specification and verification*, in "Concurrency: the Works of Leslie Lamport", D. MALKHI (editor), ACM Books, Association for Computing Machinery, 2019, vol. 29, pp. 103-129 [*DOI :* 10.1145/3335772.3335780], https://hal.inria.fr/hal-02387780

Books or Proceedings Editing

[51] P. FONTAINE (editor). *Automated Deduction – CADE-27 : 27th International Conference on Automated Deduction, Natal, Brazil, August 27–30, 2019, Proceedings*, Lecture Notes in Artificial Intelligence, Springer, Natal, Brazil, 2019, vol. 11716, forthcoming, https://hal.inria.fr/hal-02194007

[52] D. MÉRY, S. QIN (editors). *2019 International Symposium on Theoretical Aspects of Software Engineering (TASE)*, IEEE, Guillin, China, November 2019, https://hal.inria.fr/hal-02400510

Other Publications

[53] N. BERTRAND, I. KONNOV, M. LAZIC, J. WIDDER. *Verification of Randomized Distributed Algorithms under Round-Rigid Adversaries*, April 2019, 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

[54] D. GRIGORIEV, A. IOSIF, H. RAHKOOY, T. STURM, A. WEBER. *Efficiently and Effectively Recognizing Toricity of Steady State Varieties*, December 2019, https://arxiv.org/abs/1910.04100 - working paper or preprint [*DOI :* 10.04100], https://hal.inria.fr/hal-02397107

References in notes

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

[56] 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

[57] R. Back, J. von Wright. *Refinement calculus—A systematic introduction*, Springer Verlag, 1998

[58] 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

[59] 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

[60] 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

[61] 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

[62] I. 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

[63] 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

[64] 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

[65] 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

[66] L. Lamport. *Specifying Systems*, Addison-Wesley, Boston, Mass., 2002

[67] N. Le Novere, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro, J. L. Snoep, M. Hucka. *BioModels Database: a free, centralized*

database of curated, published, quantitative kinetic models of biochemical and cellular systems, in "Nucleic Acids Research", 2006, vol. 34, n^{o} suppl 1, pp. D689-D691, https://doi.org/10.1093/nar/gkj092

[68] E. W. MAYR, A. R. MEYER. *The complexity of the word problems for commutative semigroups and polynomial ideals*, in "Advances in Mathematics", 1982, vol. 46, n^{o} 3, pp. 305-329, https://doi.org/10.1016/0001-8708(82)90048-2

[69] 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

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

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

[72] C. WEIDENBACH. *Towards an Automatic Analysis of Security Protocols in First-Order Logic*, in "16th International Conference on Automated Deduction (CADE-16)", Trento, Italy, H. GANZINGER (editor), Lecture Notes in Computer Science, Springer, 1999, vol. 1632, pp. 314-328