RESEARCH CENTRE

Inria Centre
at Université de Lorraine

IN PARTNERSHIP WITH:

CNRS, Université de Lorraine,
Max-Planck-Institut für Informatik
Saarbrücken

2023 ACTIVITY REPORT

Project-Team VERIDIS

Modeling and Verification of Distributed Algorithms and Systems

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

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Proofs and Verification

Contents

Project-Team VERIDIS

Creation of the Project-Team: 2012 July 01

Keywords

Computer sciences and digital sciences

A2.1.1. – Semantics of programming languages

A2.1.4. – Functional programming

A2.1.7. – Distributed programming

A2.1.11. – Proof languages

A2.2. – Compilation

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

A4.5. – Formal methods for security

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 members, visitors, external collaborators

Research Scientists

- Stephan Merz [Team leader, INRIA, Senior Researcher, HDR]
- Engel Lefaucheux [INRIA, ISFP]
- Thomas Sturm [CNRS, Senior Researcher, HDR]
- Sophie Tourret [INRIA, Researcher]
- Uwe Waldmann [MAX PLANCK SOCIETY, Senior Researcher]
- Christoph Weidenbach [MAX PLANCK SOCIETY, Senior Researcher, HDR]

Faculty Members

- Horatiu Cirstea [UL, Professor, HDR]
- Marie Duflot-Kremer [UL, Associate Professor]
- Sergueï Lenglet [UL, Associate Professor, until Aug 2023]
- Pierre-Étienne Moreau [UL, Professor, HDR]
- Dominique Méry [UL, Professor, HDR]
- Victor Roussanaly [UL]
- Sorin Stratulat [UL, Associate Professor, HDR]

Post-Doctoral Fellows

- Martin Bromberger [Max Planck Society]
- Sibylle Möhle [Max Planck Society]

PhD Students

- Thomas Bagrel [Tweag, CIFRE]
- Ghilain Bergeron [INRIA, from Oct 2023]
- Alessio Coltellacci [INRIA]
- Rosalie Defourne [UL, ATER, until Aug 2023]
- Martin Desharnais [Max Planck Society]
- Hendrik Leidinger [Max Planck Society]
- Lorenz Leutgeb [Max Planck Society]
- Dylan Marinho [UL, until Oct 2023]
- Mohamed Amine Snoussi [Westinghouse, CIFRE, from May 2023]
- Vincent Trelat [INRIA, from Oct 2023]

Technical Staff

- Benjamin Loillier [INRIA, Engineer, from Feb 2023 until Sep 2023]
- Benjamin Loillier [UL, Engineer, until Jan 2023]

Interns and Apprentices

- Ghilain Bergeron [UL, Intern, from Mar 2023 until Sep 2023]
- Anton Danilkin [INRIA, Intern, from Jun 2023 until Aug 2023]
- Romain Nicolas [UL, Intern, from Apr 2023 until Jun 2023]
- Isaline Plaid [ENS DE LYON, Intern, from Jun 2023 until Jul 2023]
- Juliette Schilling [UL, Intern, from Apr 2023 until Jun 2023]

Administrative Assistants

- Juline Brevillet [UL, from Apr 2023]
- Sophie Drouot [INRIA, until Jun 2023]
- Sylvie Hilbert [CNRS, until Mar 2023]
- Jennifer Müller [Max Planck Society]
- Cecilia Olivier [INRIA, from Jul 2023]

External Collaborator

- Pascal Fontaine [Univ. Liège, HDR]

2 Overall objectives

The VeriDis project team includes members of the Formal Methods department 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 verified 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 (including fragments of arithmetic or of set theory) that are relevant for specifying algorithms and systems. Ideally, systems and their properties would be specified using 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 first-order logic reasoning based on superposition, 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.

Model checking is also an established and highly successful technique for verifying systems and for finding errors. Our contributions in this area more specifically target quantitative, in particular timed or probabilistic systems. A specificity of VeriDis is notably to consider partially specified systems, using *parameters*, in which case the verification problem becomes the synthesis of suitable parameter valuations.

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 verification platforms developed by other groups, and participate in international competitions. We also validate our work on verification 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 system verification, we pursue applications of some of the symbolic techniques that we develop in other domains. We have observed encouraging success in using techniques of symbolic computation for the qualitative analysis of biological and chemical networks described by systems of ordinary differential equations that were previously only accessible to large-scale simulation. Such networks include biological reaction networks as they occur with models for diseases such as diabetes or cancer. They furthermore include epidemic models such as variants and generalizations of SEIR[1] models, which are typically used for Influenza A or Covid-19. This work is being pursued within a large-scale interdisciplinary collaboration. It aims for our work grounded in verification to have 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

[1] Susceptible – Exposed – Infectious – Removed

[57], a theory solver for linear arithmetic [2], a CDCL[2] based satisfiability solver and a propositional converter to clausal normal form. Recently we have extended it to a Datalog hammer solving universal and existential queries with respect to a Horn Bernays-Schoenfinkel theory modulo linear arithmetic [62, 61].

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 [3] 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 [5].

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 specifications written in the TLA$^+$ [65] language. Our prover relies on a declarative proof language, and calls upon several automatic backends [4]. 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.

Members of VeriDis formalize a framework in the proof assistant Isabelle/HOL for representing the correctness and completeness of automated theorem provers. This work encompasses proof calculi such as ordered resolution or superposition, as well as concrete prover architectures such as Otter or DISCOUNT loops. It also covers the most recent splitting techniques that bring proof calculi closer to SMT solvers.

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 [3, 8], and in applying them to concrete use cases. In particular, the concept of *refinement* [54, 58, 69] 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.

[2] conflict-driven clause learning

[3] Satisfiability Modulo Theories [59]

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

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

Model checking The paradigm of model checking is based on automatically verifying properties over a formal model of a system, using mathematical foundations. Model checking, while useful and highly successful in practice, can encounter the infamous state space explosion problem. One direction of VeriDis therefore addresses the efficiency of model checking, by proposing new algorithms or heuristics to speed up analysis. We notably focus on the quantitative setting (time, probabilities), and more specifically on the parametric paradigm where some quantitative constants are unknown, and the goal becomes to synthesize suitable valuations. A recent application of the VeriDis team is that of *opacity* (in the more general field of cybersecurity), addressed using model checking. The team considers a novel definition of opacity in timed automata, where an attacker has only access to the execution time; several recent works address this direction.

3.3 Verification and Analysis of Dynamic Properties of Biological Systems

The unprecedented accumulation of information in biology and medicine during the last 20 years led to a situation where any new progress in these fields is dependent on the capacity to model and make sense of large data. Until recently, foundational research was concerned with simple models of 2 to 5 ordinary differential equations. The analysis of even such simple models was sufficiently involved that it resulted in one or several scientific publications for a single model. Much larger models are built today to represent cell processes, explain and predict the origin and evolution of complex diseases or the differences between patients in precision and personalized medicine. For instance, the biomodels.net model repository [66] contains thousands of hand-built models of up to several hundreds of variables. Numerical analysis of large models requires an exhaustive scan of the parameter space or the identification of the numerical parameters from data. Both are infeasible for large biological systems because parameters are largely unknown and because of the curse of dimensionality: data, even rich, become rapidly sparse when the dimensionality of the problem increases. On these grounds, VeriDis researchers aim at formal symbolic analysis instead of numerical simulation.

As an illustration of the approach, consider `BIOMD0000000716` in the above-mentioned BioModels database, which models the transmission dynamics of subtype H5N6 of the avian Influenza A virus in the Philippines in August 2017 [67]. This model describes four species (susceptible/infected bird or human) together with their dynamics. Using purely symbolic algorithms, we obtain a decomposition of the dynamics into three subsystems T_1, T_2, and T_3 with attractive manifolds $\mathscr{M}_1$, $\mathscr{M}_2$ and $\mathscr{M}_3$, and the constant factors appearing in the corresponding differential equations indicate that the system T_2 is 125 times slower than T_1, and that T_3 is another 125 times slower. This multiple time scale reduction emphasizes a cascade of successive relaxations of model variables. Figure 1(a) shows the surface of $\mathscr{M}_1$ projected into 3D space, with the line and the dot representing the submanifolds $\mathscr{M}_2$ and $\mathscr{M}_3$. Figure 1(b) illustrates the direction field of T_1 projected into 2D space. The curve corresponds to $\mathscr{M}_1$, indicating that the population of susceptible birds relaxes and that these variables reach quasi-steady state values. Figure 1(c) represents the direction field of T_2 on $\mathscr{M}_1$ projected into 2D space. The line corresponds to

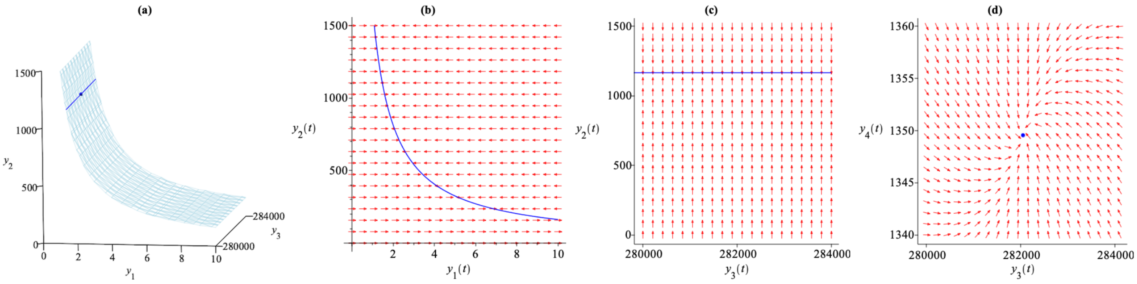

Figure 1: Illustration of the analysis of an epidemic model of avian Influenza A.

$\mathscr{M}_2$, showing the relaxation of the population of infected birds. Finally, figure 1(d) shows the direction field of T_3 on $\mathscr{M}_2$ projected into 2D space. The dot corresponds to $\mathscr{M}_3$, indicating the relaxation of the populations of susceptible and infected humans to a stable steady state.

The computation time is less than a second. The computation is based on massive SMT solving over various theories, including QF_LRA for tropicalizations, QF_NRA for testing Hurwitz conditions on eigenvalues, and QF_LIA for finding sufficient differentiability conditions for hyperbolic attractivity of critical manifolds. Gröbner reduction techniques are used for final algebraic simplification [53]. Observe that numerical simulation would not be able to provide such a global analysis of the overall system, even in the absence of symbolic parameters, as is the case in our rather simple example.

4 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 underlie 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, systems biology, and system medicine on the analysis of reaction networks and epidemic models in order to infer principal qualitative properties. Our techniques complement numerical analysis techniques and are validated against collections of models from computational biology.

The team uses extensions of timed automata (such as parametric timed automata [55]) as an underlying formalism to solve practical questions. Our work on parametric timed automata is partly motivated by applications in cybersecurity, notably within the ANR-NRF ProMiS project. Foundational decidability results and novel notions of non-interference and opacity for this class of automata allow us, for example, to determine the maximal frequency of attacker actions for the attack to succeed (i.e., so that these actions remain invisible to the external observer). Several software artefacts were implemented by the team in this domain [56].

5 Highlights of the year

Pascal Fontaine obtained an Amazon Research Award, and Stephan Merz received a research award from Oracle Corporation.

Jasmin Blanchette, Qi Qiu and Sophie Tourret received the best paper award at CADE-29, the 29th International Conference on Automated Deduction, for their paper about verified given clause procedures [23] .
An article on the use of interactive proof assistants for formalizing mathematical developments was published in Communications of the ACM [12].

6 New software, platforms, open data

6.1 New software

6.1.1 Redlog

Name: Reduce Logic System

Keywords: Computer algebra system (CAS), First-order logic, Constraint solving, Quantifier Elimination

Functional Description: Redlog is an integral part of the interactive computer algebra system Reduce. It supplements Reduce's comprehensive collection of powerful symbolic computation methods 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 (quantified satisfiability solving), and many more.

News of the Year: The year 2023 corresponds to a transition in the development of Redlog. Parts of the code are more than 25 years old, and that version 1 of the underlying computer algebra system Reduce has been published even more than 50 years ago. During the last five years, the code has been revised and simplified aiming at better long-term maintainability. The existing code base remains available and will continue to be supported. New developments, however, will focus on the successor system Logic1, which we will be announced during 2024, and several parts of the Redlog core have been rewritten in view of this new system.

URL: https://www.redlog.eu/

Contact: Thomas Sturm

Participants: Thomas Sturm, Andreas Dolzmann, Melanie Achatz, Marek Kosta, Aless Lasaruk, Herbert Melenk, Winfried Neun, Andreas Seidl, Christoph Zengler, Volker Weispfenning

6.1.2 SPASS Workbench

Name: SPASS Automated Reasoning Workbench

Keywords: Decision, Linear Systems Solver

Functional Description: The SPASS Workbench is a collection of tools for various reasoning tasks in logic. It currently comprises the first-order theorem prover SPASS, a decision procedure for linear (mixed) arithmetic SPASS-IQ, a satisfiability modulo theory (SMT) solver for linear (mixed) arithmetic, a propositional satisfiability (SAT) solver SPASS-SAT and a propositional conjunctive normal form converter SPASS-CNF.

News of the Year: In 2023, work focused on the development of a new solver SPASS-SPL for a fragment we call SUPERLOG, which is the first-order Bernays Schoenfinkel class extended with linear arithmetic. A particular application domain for this solver will be the verification of supervisors, i.e. electronic control units used in embedded systems.

URL: https://www.mpi-inf.mpg.de/departments/automation-of-logic/software/spass-workbench/

Publications: hal-03531893, hal-03531889, hal-03531894

Contact: Christoph Weidenbach

Participants: Martin Bromberger, Christoph Weidenbach

6.1.3 E-Cyclist

Keyword: Cyclic proofs

Functional Description: Checking the soundness of cyclic induction reasoning for first-order logic with inductive definitions (FOLID) 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 in a paper presented at the CiSS2019 conference (Circularity in Syntax and Semantics). E-Cyclist is an extension of the Cyclist prover (http://www.cyclist-prover.org/) that integrates this method. It successfully checked all the proofs included in the Cyclist distribution. The implementation details have been presented at SCSS 2021 (ID HAL: hal-02464242).

News of the Year: Functionality for the reconstruction of proofs found by E-Cyclist in the Coq proof assistant was significantly extended in 2023.

URL: `https://members.loria.fr/SStratulat/files/e-cyclist.zip`

Contact: Sorin Stratulat

6.1.4 TLAPS

Name: TLA+ proof system

Keyword: Proof assistant

Functional Description: TLAPS is a platform for developing and mechanically verifying proofs about specifications written in the TLA+ language. The TLA+ proof language is hierarchical and explicit, allowing a user to decompose the overall proof into proof steps that can be checked independently. TLAPS consists of a proof manager that interprets the proof language and generates a collection of proof obligations that are sent to backend verifiers. The current backends include the tableau-based prover Zenon for first-order logic, Isabelle/TLA+, an encoding of TLA+ set theory as an object logic in the logical framework Isabelle, an SMT backend designed for use with any SMT-lib compatible solver, and an interface to a decision procedure for propositional temporal logic.

News of the Year: A new SMT backend was developed in 2023 based on the thesis of Rosalie Defourné and is currently being integrated in the main code base on TLAPS. A new release based on the Dune build system and integrating recent developments such as support for reasoning about the ENABLED operator or a new Isabelle backend is in preparation.

URL: `https://tla.msr-inria.inria.fr/tlaps/content/Home.html`

Contact: Stephan Merz

Participants: Damien Doligez, Stephan Merz

Partner: Microsoft

6.1.5 veriT

Keywords: Automated deduction, Formula solving, Verification

Functional Description: VeriT is an open, trustable and efficient SMT (Satisfiability Modulo Theories) solver. It comprises a propositional satisfiability (SAT) solver, an efficient decision procedure for uninterpreted symbols based on congruence closure, a simplex-based decision procedure for linear arithmetic, and instantiation-based quantifier reasoning.

News of the Year: Like the previous recent years, efforts in 2023 have been focused on higher-order logic, and better proof production. We also initiated a profound code refactoring phase, to better accommodate the role of the solver as a platform for testing new ideas.

We target applications where validation of formulas is crucial, such as proof about specifications written in the B or TLA$^+$ languages, and we 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*.

URL: `http://www.veriT-solver.org`

Contact: Pascal Fontaine

Participants: Pascal Fontaine, Sophie Tourret

Partner: Université de Lorraine

6.2 New platforms

6.2.1 ODEbase

Participants: Thomas Sturm.

Name: Online Database of Biomodels Involving Ordinary Differential Equations

Keywords: Automated reasoning, Dynamical systems, Interdisciplinary research, Qualitative analysis

Scientific Description: Symbolic Computation and Automated Reasoning allow qualitative answers to biological questions. Qualitative methods analyze dynamical input systems as formal objects, in contrast to investigating only a subset of the state space, as is the case with numerical simulation. A common format used in mathematical modeling of biological processes is the Systems Biology Markup Language SBML. However, symbolic tools and libraries have a different set of requirements for their input data than their numerical counterparts. The use of SBML data in Symbolic Computation and Automated Reasoning requires significant pre-processing that combines automated translation steps with human interaction and expertise. ODEbase provides pre-processed input data derived from established existing biomodels.

Functional Description: SBML, which is technically an XML instance, has been designed as a very liberal format, and contributors of models are primarily researchers with their key expertise in the natural sciences. This creates a situation where SBML features are used in unexpected ways in general. A sound presentation of corresponding models outside the SMBL framework then requires expertise in the life sciences as well as mathematical competence, primarily in algebra and in dynamical systems. Technically we use a set of Python tools, which we have developed for the semi-automatic conversion of SBML models. Since the conversion process is not fully automatic and our resources are limited, we focus on models that we identify as interesting for Symbolic Computation and Automated Reasoning approaches. Our principal source of models is the renowned online database biomodels.net.

News of the Year: New models were integrated, and ODEbase comprises 662 models at the time of writing. A sign of recognition of ODEbase by the relevant scientific communities is the fact that several publications in the reporting period refer to its use as the principal source of data, e.g. [63, 68, 71].

URL: https://odebase.org

Publications: hal-03651751

Contact: Thomas Sturm

Partners: Christoph Lüders, University of Bonn, Germany, Ovidiu Radulescu, University of Montpellier, France.

7 New results

7.1 Automated and Interactive Theorem Proving

Participants: Martin Bromberger, Alessio Coltellacci, Rosalie Defourné, Martin Desharnais, Pascal Fontaine, Hendrik Leidinger, Lorenz Leutgeb, Stephan Merz, Sibylle Möhle, Hans-Jörg Schurr, Simon Schwarz, Sorin Stratulat, Vincent Trélat, Sophie Tourret, Marco Voigt, Uwe Waldmann, Christoph Weidenbach.

7.1.1 Contributions to SMT Techniques

Quantifier Handling in Higher-Order SMT. *Joint work with Haniel Barbosa (Univ. Federal de Miras Gerais, Brazil).*

SMT solvers have throughout the years been able to cope with increasingly expressive logics, from ground formulas to full first-order logic (FOL). In the past, we proposed a pragmatic extension for SMT solvers to support higher-order logic reasoning natively without compromising performance on FOL reasoning, thus leveraging the extensive research and implementation efforts dedicated to efficient SMT solving. However, the higher-order SMT solvers resulting from this work are not as effective as we would expect given their performances in first-order logic. We believe this comes from the fact that only the core of the SMT solver has been extended, ignoring in particular the modules for quantifier instantiation.

This motivated us to start working on an extension of the main quantifier-instantiation approach (congruence closure with free variables, CCFV) to higher-order logic in 2020. We are working on an encoding of the CCFV higher-order problem into a set of SAT constraints. In previous years, we concentrated our efforts on the theory, to prove the soundness and completeness of our approach, and developed pseudo-code for all elements of CCFV computation. In 2022 and 2023, these algorithms were implemented in a C++ library, and they were tested on benchmarks from the SMT-lib collection. We started to integrate this library within a new SMT framework. The library will eventually be released under an open-source permissive license.

7.1.2 Automated reasoning techniques beyond SMT

Extensions of a formal framework for automated reasoning. We are part of a group developing a framework for formal refutational completeness proofs of abstract provers that implement automated reasoning calculi, especially calculi based on saturation such as ordered resolution and superposition. In previous work, we published a framework that fully captures the dynamic aspects of proof search with a saturation calculus. This framework covers clause splitting as supported by modern superposition provers with the help of a SAT solver. In particular, our formalization revealed some completeness issues with the theorem prover Vampire.

This year, we extended the Isabelle formalization by representations of the main loops of saturation-based theorem provers and their fairness conditions. In the process, we found and repaired several issues with the (in fact, our own) description of the Zipperposition loop, a novel loop that handles inferences producing an infinite stream of conclusions. In parallel, Martin Desharnais, for his PhD thesis, completed an instantiation of this framework for the superposition calculus. We also made progress on the Isabelle/HOL mechanization of the framework with clause splitting. Ghilain Bergeron contributed to this endeavor for his master thesis. This last piece of work is still ongoing, while the first one has been completed in 2022 and led to a paper presented at CADE in 2023 [23].

Effective Symbolic Model Construction. When automatic reasoning techniques are applied in order to prove a specific property of some system, a proof is a certificate of success and we have worked on explaining the gist of it [64]. If a proof attempt fails, automatic reasoning techniques may still terminate and implicitly provide a representation of a (counter) model to the property of interest. We have worked on effective representations for such counter models providing insights into why the desired property does not hold. This way, either the system, the formalization of it or the property can be debugged [47, 27].

Certification of $\mathrm{FOL}_{\mathrm{ID}}$ cyclic proofs. Cyclic induction is a powerful reasoning technique that consists in blocking the proof development of certain subgoals already encountered during the proof process. In the setting of first-order logic with inductive definitions and equality ($\mathrm{FOL}_{\mathrm{ID}}$), cyclic proofs can be built automatically by the CYCLIST prover, but their implementations are error-prone and the human validation may be tedious. On the other hand, cyclic induction is not yet integrated into certifying proof environments that support first-order logic and inductive definitions, such as Isabelle and Coq.

We have proposed in [20] a general procedure for certifying formula-based Noetherian induction reasoning in order to check $\mathrm{FOL}_{\mathrm{ID}}$ cyclic proofs using Coq. The output is a Coq script proving several theorems and lemmas for which the deductive part translates the E-CYCLIST proof steps *without* using proof reconstruction techniques. This approach also allows for finding errors in cyclic proofs in a very precise way, at the level of proof steps. We established a bridge between formula-based Noetherian induction and $\mathrm{FOL}_{\mathrm{ID}}$ cyclic induction, by identifying a class of pre-proofs certifiable by Coq when some ordering and derivability constraints are satisfied, such as those produced by the E-CYCLIST prover. The advantages of our approach are threefold:

1. The certification of cyclic $\mathrm{FOL}_{\mathrm{ID}}$ proofs is *mechanical*. Coq can validate every single step from the E-CYCLIST proofs, as well as the induction arguments; also, it helps to identify errors in a very precise way.

2. There is a great potential for *automation*. The methodology has already been used to automatically convert to Coq scripts implicit induction proofs [70].

3. Cyclic induction can be *directly* performed in Coq. A library of Coq functions is provided and can be reused to manage the induction part.

Proofs for TLA$^+$. In her PhD work, Rosalie Defourné defined improved encodings of the non-temporal theory of TLA$^+$ in the input languages of automated theorem provers for first-order and higher-order logic, including SMT solvers and Zipperposition. The SMT encoding relies on an axiomatization for the operators of TLA$^+$ set theory, annotated by triggers for finding relevant instances of these axioms. The previously existing encoding heavily relied on rewriting input formulas in order to simplify them before submitting them to the backend solver. Verifying its soundness required to not only inspect the axioms given to the solver, but also understanding the preprocessing techniques that were applied to the input formulas. Besides, optimizing the set of rewrite rules was delicate because properties such as confluence and termination had to be maintained. In contrast, the soundness of the new encoding can be determined directly by verifying the background axioms. Adding triggers does not endanger soundness, and benchmarking the new backend over an extensive corpus of existing TLA$^+$ proofs showed that it performs comparably or better than the old encoding. This work was presented in a conference publication [31] and in Rosalie Defourné's PhD thesis [45], which was defended in November.

7.2 Formal Methods for Developing and Analyzing Algorithms and Systems

Participants: Thomas Bagrel, Ghilain Bergeron, Martin Bromberger, Horatiu Cirstea, Marie Duflot-Kremer, Engel Lefaucheux, Serguei Lenglet, Benjamin Loillier, Dylan Marinho, Dominique Méry, Stephan Merz, Pierre-Etienne Moreau, Victor Roussanaly, Amine Snoussi, Christoph Weidenbach.

7.2.1 Contributions to Formal Methods of System Design

Formal framework for developing safety-critical interactive systems F3FLUID is a unified formal framework for the development of safety-critical interactive systems. This framework is based on the Formal Language of User Interface Design (FLUID) defined in the FORMEDICIS ANR project (2017–2022) for expressing high-level system requirements for interactive systems. FLUID is specifically designed for handling concepts of safety-critical interactive systems, including domain knowledge. A FLUID model is used as a source model for the generation of several target models in different modeling languages to support the formal verification methods, such as theorem proving and model checking.

The Event-B modeling language is used for checking functional behaviors, user interactions, safety properties, and domain properties. A FLUID model is transformed into an Event-B model, and then, the Rodin tool is used to check the internal consistency with respect to the given safety properties. In addition, an interactive cooperative objects (ICOs) model is derived from the Event-B model for animation, visualization and validation of dynamic behaviors, visual properties, and task analysis, using the ProB model checker. Finally, an industrial case study, complying with the ARINC 661 standard, Multi-Purpose Interactive Applications (MPIA), is used to illustrate the effectiveness of our F3FLUID framework for the development of safety-critical interactive systems. Moreover, we show [18] how formal ontologies can be used to model domain-specific knowledge, as well as how system models may refer to these ontologies through annotations. It relies on the Event-B refinement and proof state-based method, and the associated theories, to define a framework in which domain-specific knowledge ontologies are formalized as Event-B theories defining data types used to type Event-B system design models. Finally, this framework is deployed for the specific case of interactive critical systems. To illustrate the proposed approach, a case study of the Traffic Collision Avoidance System (TCAS) is developed.

A Secure Low-Latency Protocol for Mitigating High Computation Overhead in WIFI Networks The increase in popularity of wireless networks in industrial, embedded, medical and public sectors has made them an appealing attack surface for attackers who exploit the vulnerabilities in network protocols to launch attacks such as Evil Twin, Man-in-the-middle, sniffing, etc., which may result in economic and non-economic losses. To protect wireless networks against such attacks, IEEE 802.11 keep updating the protocol standards with new and more secure versions. There has always been a direct correlation between attacks and the improvement of protocol standards. As the sophistication of attacks increases, protocol standards tend to move towards higher security, resulting in a significant increase in both latency and computational overhead, and severe degradation in the performance of low-latency applications such as Industrial Internet of Things (IIoT), automotive, robotics, etc. The paper [16] highlights the importance of both latency and security in wireless networks from the implementation and performance perspectives. We review existing IEEE 802.11 protocols in terms of security offered and overhead incurred to substantiate the fact that there is a need for a protocol which in addition to providing optimum security against attacks also maintains the latency and overhead. We also propose a secure and low-latency protocol known as Secure Authentication Protocol (SAP) which operates in two phases, where the first phase is a one-time process implemented using asymmetric cryptography and the second phase is implemented using symmetric cryptography. The protocol is structured in a way that it maintains the original structure of IEEE 802.11 protocols and performs both phases using fewer messages than existing protocols. By simulating the protocol using the well-established OMNeT++ simulator, we demonstrated that the proposed protocol incurs a low computation overhead, making it ideal for low-latency applications. We extensively verified the security properties of the proposed protocol using formal verification through the widely-accepted Scyther tool. Finally, we perform a comparative analysis of SAP with existing IEEE 802.11 wireless network protocols to highlight the improvement.

Modeling hybrid systems by refinement. Whenever continuous dynamics and discrete control interact, hybrid systems arise. As hybrid systems become ubiquitous and more and more complex, analysis and synthesis techniques are in high demand to design safe hybrid systems. This is however challenging due to the nature of hybrid systems and their designs, and the question arises of how to formulate and reason about their safety problems. Previous work has demonstrated how to extend the discrete modeling language Event-B with support for continuous domains to integrate traditional refinement in hybrid system design. We have extended our strategy [50] that can coherently refine an abstract hybrid system design with safety constraints down to a concrete one, integrated with implementable discrete control, that can behave safely. We demonstrate our proposal on a smart heating system that regulates room temperature between two reference values. The main innovation lies in the fact that we combine the Event-B modelling language (events) and the B modelling language (operations). The transformation of events into operation is proved correct using a proof assistant. We also explore the frequency model approach of control theory using the Event-B modelling language. The frequency modelling style is not so usual and we provided a Python artefact for assisting with the use of the approach.

Semantics transformations. Any given programming language may come with several semantics definitions, such as big-step, small-step, or even abstract machines, which can be seen as an implementation of a language. They all describe identical behaviors of programs, but each may be better adapted for some purpose: for instance, small-step semantics are better suited to prove subject reduction.

To have access to all kinds of semantics at once, we develop transformations between semantics to be able to generate one from the other at no extra cost for the language designer. In a previous work [60], we developed a semantics format called zipper semantics, a small-step semantics from which we can automatically derive sound and complete abstract machines for non-deterministic languages such as process calculi. We build on this work in two ways: first, we propose a new format of zipper semantics, called leaf-first, in which we can express lazy scope extrusion or join patterns. Next, we make the non-deterministic abstract machines more efficient by reusing some design principles of abstract machines for the lambda-calculus, like environments and refocusing.

7.2.2 Automated Reasoning Techniques for Verification

Synthesis of inductive invariants for distributed algorithms. In joint work with Aman Goel (Amazon) and Karem Sakallah (University of Michigan), we investigated the use of symbolic model checking techniques for automatically computing inductive invariants for parameterized distributed algorithms. Specifically, the IC3PO model checker applies the well-known IC3 model checking algorithm to finite instances of the algorithm and, in case of success, retrieves inductive invariants for those instances. It then inspects these invariants for symmetries with respect to space (processes) and time (e.g., ballot numbers used during the algorithm) and expresses those symmetries by introducing quantifiers. The resulting formulas are then checked for larger input sizes until no new invariants are inferred. We applied the technique to two versions of the well-known Bakery algorithm that ensures mutual exclusion among N processes communicating via shared variables. For both versions, IC3PO generated invariants that were remarkably similar to, but more permissive than, human-written invariants used in a previous interactive proof of the algorithm. This work, presented at FORTE 2023 [34], suggests that automated invariant inference is becoming a viable alternative to labor-intensive human-written proofs. In ongoing work, we are investigating invariant synthesis for the significantly more complex Raft consensus algorithm. One of the main problems there is how to represent the sequential log maintained by the nodes in a first-order logic of limited expressiveness.

New primitives in PlusCal for modeling distributed algorithms. We designed an extension of the PlusCal algorithmic language for modeling distributed algorithms. Rather than introducing many new features that could break the design objectives of PlusCal being a lightweight front-end to writing TLA$^+$specifications, we added only a few orthogonal concepts inspired from those found in distributed programming languages while both remaining compatible with the existing language and keeping simple the generation of human-readable TLA$^+$ specifications. Compared to the original PlusCal language, Distributed PlusCal allows processes to consist of multiple threads that communicate via process-local variables, and it introduces communication channels that can be declared as preserving FIFO order or not. We illustrated Distributed PlusCal using two well-known algorithms and our preliminary findings indicate that the extensions provided help us express distributed algorithms in a natural way. Moreover, any overhead incurred in verification with respect to a specification written in TLA$^+$ is not different from that of ordinary PlusCal. This work was presented in a conference paper [29].

Validating traces of distributed systems. Programming distributed systems is error-prone, and while formal methods such as TLA$^+$ can help design such systems, the translation from the verified specification towards the final implementation is rarely certified. Nevertheless, we believe that we can still gain confidence in the fact that an implementation behaves according to a high-level specification by collecting traces of executions and checking that they are allowed by the specification. We have developed a framework for the instrumentation of Java programs allowing one to record events and variable updates that are meaningful at the level of the TLA$^+$ specification, and we use the TLA$^+$ model checker to verify that the recorded trace of an execution indeed corresponds to a possible execution of the high-level specification[4]. Although this cannot formally establish the correctness of an implementation, preliminary experience with this approach indicates that it can be surprisingly effective at discovering discrepancies between the TLA$^+$ specification and the implementation.

7.2.3 Model checking for timed and linear systems

Opacity of timed automata. Opacity of a system is a property describing the impossibility for an external attacker to deduce some private information by observing an execution of the system. In two publications [22, 11] we considered the opacity of systems modeled by timed automata where an attacker has access to the duration of the executions only. Our goal was to determine whether a system was opaque, and in some cases, under which conditions it could be made opaque. More precisely, the models we considered allowed for parameters representing unknown delays within the system, and we wished to identify for which parameter values opacity was achieved. Throughout this line of research we have considered many different variants of the notion of opacity, for instance by allowing the secret behaviour to expire, meaning that some private behaviour could stop being deemed private if the private information was not detected by the observer quickly enough. The paper [11] serves as a survey of the current state of this line of research.

Mathematical tools for the analysis of system. For the analysis of timed automata, including the analysis of opacity mentioned above, the notion of semi-linear sets (i.e. sets that can be defined by Presburger arithmetics) appears regularly. For instance, the durations of runs in a timed automata can be described as such a set. Presburger arithmetics however fails once parameters are included within the automata: the presence of parameters often leads to multiplications between integer variables, which cannot be represented in Presburger arithmetic. Peano arithmetic could of course represent those sets, but this arithmetic is well known to be undecidable and thus not a real option. Much previous research has studied the gap between Presburger and Peano arithmetic. To our knowledge, none of the existing work was able to handle the formulas that we generate in the analysis of parametric timed automata. In [51], we developed a parametric extension to Presburger arithmetic tailored for our needs. It shows how to handle some formulas beyond Presburger which we plan to use in future work.

Model checking of rounded linear loops. Loops are a fundamental staple of any programming language, and the study of loops plays a pivotal role in many subfields of computer science, including automated verification, abstract interpretation, program analysis, semantics, etc. A lot of work has been done in the linear dynamical systems community to represent program loops as simple linear systems, and to analyze which kind of properties formulated in Monadic Second Order Logic could be checked over the evolution of a loop. In [35], we modify the usual linear models used to account for the fact that computers rely on floating points. This change leads to a completely different approach to the problem, as the usual tools to tackle linear dynamical systems fail to apply. An interesting result is that, while termination of simple linear loops is a long-standing open problem in the usual setting, it becomes undecidable with floating points. Moreover, if we add some very simple restriction on the system, which changes nothing in the original setting, termination in our setting becomes decidable.

7.3 Verification and Analysis of Dynamic Properties of Biological Systems

Participants: Thomas Sturm.

[4] see GitHub repository

7.3.1 Approximate Conservation Laws

Section 3.3 showed an example for the analysis of the kinetics of a chemical reaction network at multiple timescales [53]. Although general in its implementation, this reduction method can fail in a number of cases. A major cause of failure is the degeneracy of the quasi-steady state, when the fast dynamics has a continuous variety of steady states. Typically, this happens when the fast truncated ODEs have first integrals, i.e. quantities that are conserved on any trajectory and that have to be fixed at arbitrary values depending on the initial conditions. The quasi-steady states are then no longer hyperbolic, because the Jacobian matrix of the fast part of the dynamics becomes singular. To address this issue, we have now proposed a concept of *approximate conservation laws*, which allows additional reductions.

Technically, this framework requires parametric versions of various established algorithms from Symbolic Computation. One simple example is the computation of the rank of a matrix with real parameters, which produces a formal finite case distinction where possible ranks are paired with necessary and sufficient conditions as Tarski formulas. This allows to identify critical cases with respect to the above-mentioned singularity of the Jacobian. Another example is the use of comprehensive Gröbner bases in the course of parametric computation of certain syzygy modules. From a practical point of view, a central issue with all such algorithms is the combinatorial explosion of the number of cases.

We use SMT solving as well as real quantifier elimination methods to detect inconsistent cases and prune the tree of case distinctions early. The decision procedures used are typically double exponential and can easily turn into a bottleneck preventing termination within reasonable time altogether, in particular when the degrees of polynomial terms get larger. Since the results remain correct also without the elimination of some redundant cases, we combine various methods and use suitable timeouts. This work has resulted in two articles, which have been accepted for publication in the SIAM Journal on Applied Dynamical Systems

8 Bilateral contracts and grants with industry

8.1 Bilateral contracts with industry

TLA$^+$Trace Validation

Duration: January 2023 – December 2024

Industrial Partner: Oracle Corporation

Team participants: Horatiu Cirstea, Benjamin Loillier, Stephan Merz

Summary: The objective of this work is to find discrepancies between traces of distributed programs collected during their execution and high-level specifications, written in TLA$^+$, of the algorithms that the programs are supposed to implement.

Type systems for the memory safety of functional programs

Duration: April 2022 – March 2025

Industrial Partner: Tweag

Team participants: Thomas Bagrel, Horatiu Cirstea

Summary: In his PhD work supported by a CIFRE contract, Thomas Bagrel studies type systems and corresponding constructions in functional programs, notably based on (anti-)patterns, for guaranteeing programs that are memory safe and can be compiled to efficient machine code.

Reengineering of protocols for industrial controllers

Duration: May 2023 – April 2026

Industrial Partner: Westinghouse France

Team participants: Amine Snoussi, Marie Duflot-Kremer, Engel Lefaucheux, Stephan Merz

Summary: In his PhD work supported by a CIFRE contract, Amine Snoussi aims at constructing formal models and simulations of protocols that are used for industrial controllers, in particular for the diagnosis and control of electronic components in nuclear power plants.

9 Partnerships and cooperations

Participants: Alessio Coltellacci, Marie Duflot-Kremer, Pascal Fontaine, Engel Lefaucheux, Sergueï Lenglet, Dominique Méry, Stephan Merz, Sorin Stratulat, Sophie Tourret, Vincent Trélat.

9.1 International research visitors

9.1.1 Visits to international teams

Research stays abroad

Sergueï Lenglet

Visited institution: University of Wrocław

Country: Poland

Dates: from 8 May 2023 to 12 May 2023, and from 4 Dec 23 to 8 Dec 2023.

Context of the visit: Research collaboration on program semantics.

Mobility program/type of mobility: Partenariat Hubert Curien.

9.2 European initiatives

9.2.1 Other European Programs

COST EuroProofNet.

Program: COST

Title: European Research Network on Formal Proofs (COST action CA20111)

Duration: October 2021 – October 2025

Coordinator: Inria

Inria contact: Frédéric Blanqui, Stephan Merz

Team participants: Pascal Fontaine (WG leader, management committee), Alessio Coltellacci, Stephan Merz, Sophie Tourret

Summary: EuroProofNet is the European research network on digital proofs. EuroProofNet aims at boosting the interoperability and usability of proof systems. The action now gathers more than 400 researchers from 43 different countries; it is coordinated by a core group chaired by Frédéric Blanqui. EuroProofNet organizes meetings and schools, and provides grants to its members for short-term scientific missions in another country.

AiRobo

Program: Erasmus+

Title: Artificial Intelligence based Robotics

Duration: December 2023 – November 2026

Coordinator: West University of Timisoara (Romania)

Partners: University of Macedonia (Thessaloniki, Greece), RWTH (Aachen, Germany), Eszterházy Károly Catholic University (Eger, Hungary), Université de Lorraine.

Inria contact: Sorin Stratulat

Summary: AiRobo is an innovative project, focused on both research and teacher training in the fields of robotics, artificial intelligence and formal verification. Within the project, a comprehensive set of didactic support materials will be developed: the AiRobo book, seven robotic applications in different fields (three of them are addressed to people with disabilities and migrants), tools and video tutorials, as well as scientific publications at international conferences. These materials will be used as support in the process of teaching courses at partner universities and even in other universities around the world.

9.3 National initiatives

ANR Project BiSoUS

Title: Better Synthesis for Underspecified Quantitative Systems

Duration: March 2023 – February 2027

Coordinator: Didier Lime, École Centrale de Nantes & LS2N

Partner Institutions:

- IRISA, Rennes
- LIPN, Université Sorbonne Paris Nord (Paris 13)
- LS2N École Centrale de Nantes (coordinator)
- LMF, Université Paris-Saclay

Team participants: Marie Duflot-Kremer, Engel Lefaucheux

Summary: Computer systems are ubiquitous and identifying their possible defects is crucial already at the earliest stages of their development, when many aspects, including the environments or the execution platforms, have not been fixed. Verification must then be performed on underspecified models and should give answers as understandable as possible. In this project, we aim at developing verification techniques for underspecified models that take this explainability constraint into account, by optimizing resources, such as energy or memory, and synthesizing more precise requirements on the underspecified aspects of the models under which the system behaves correctly. We depart from classical formalisms and consider their combined extensions with three complementary ingredients: costs/rewards for resource consumption; parameters for unknown quantitative characteristics; and games for representing all the behaviours of the underspecified system.

Keywords: Verification, Model checking, parametrised systems, games with guarantees

More information: BiSoUS Web site

ANR Project BLaSST

Title: Enhancing B Language Reasoners Using SAT and SMT Techniques

Duration: March 2022 – February 2026

Coordinator: Stephan Merz

Partner Institutions:

- Inria Nancy (coordinator)
- University of Artois & CRIL, Lens
- CLEARSY, Aix-en-Provence
- University of Liège, Belgium

Team participants: Pascal Fontaine, Stephan Merz, Vincent Trélat, Sophie Tourret

Summary: The BLaSST project targets bridging combinatorial and symbolic techniques in automatic theorem proving, in particular for proof obligations generated from B models. It focuses on advancing the state of the art in automated reasoning, in particular SAT and SMT techniques, and on making these techniques available to software verification. More specifically, encoding techniques, optimized resolution techniques, model generation, and lemma suggestion will be investigated. The expected scientific impact is a substantially higher degree of automation of solvers for expressive input languages by leveraging higher-order reasoning and enumerative instantiations over finite domains, as well as useful feedback for verification conditions that cannot be proved. The effectiveness of the techniques developed in the project will be quantified by applying them to benchmark sets provided by the industrial partner. The industrial impact of BLaSST will be a higher productivity of proof engineers. The collections of benchmarks and the reasoning engines will be made openly available under permissive open-source licenses.

Keywords: B method, deductive verification, SAT, SMT, higher-order logic

More information: BLaSST Web site

ANR Project DISCONT

Title: Correct integration of discrete and continuous models

Duration: March 2018 – September 2023

Coordinator: Dominique Méry

Partner Institutions:

- University of Lorraine (coordinator)
- ENSEEIHT & IRIT, Toulouse
- University Paris Est & LACL, Créteil
- CLEARSY, Aix-en-Provence

Team participants: Dominique Méry

Summary: 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.

Keywords: cyber-physical systems, discrete models, continuous models, refinement, verification, tools

More information: DISCONT Web site

ANR Project EBRP

Title: Enhancing EventB and RODIN: EventB-Rodin-Plus

Duration: January 2020 – January 2024

Coordinator: Dominique Méry

Partner Institutions:

- INPT-ENSEEIHT & IRIT, Toulouse
- CentraleSupelec & LRI
- University of Lorraine & LORIA
- University Paris-Est Créteil & LACL
- University of Düsseldorf, Germany
- University of Southampton, School of Electronics and Computer Science, United Kingdom

Team participants: Dominique Méry

Keywords: formal IDE, theory, proof managementr, cyber-physical systems, discrete models, continuous models, refinement, verification, tools

Summary: The purpose of EBRP is to enhance Event-B and the corresponding Rodin toolset. This will be done by engaging in some basic research dealing with various mathematical theories that are not currently available in Event-B and Rodin. The development of complex systems usually involves different scientific disciplines and skills. For instance, modeling behaviors and interactions of autonomous systems may require concepts from control theory such as differential equations, communication protocols, resource allocation, access control rules, etc. EBRP targets the definition of extension mechanisms for Event-B rather than defining domain-specific modeling languages, and implementing those mechanisms within Rodin.

More information: EBRP Web site

ANR Project ICSPA

Title: Interoperable and Confident Set-based Proof Assistants

Duration: January 2022 – December 2025

Coordinator: Catherine Dubois, ENSIIE & Samovar

Partner Institutions:

- ENSIIE & Samovar, Évry
- Inria (Nancy and Saclay research centers)
- University Paul Sabatier & IRIT, Toulouse

- University of Montpellier & LIRMM, Montpellier
- CLEARSY, Aix-en-Provence

Team participants: Alessio Coltellacci, Dominique Méry, Stephan Merz

Summary: The B, Event-B, and TLA$^+$ formal methods are based on different flavors of set theory. The ICSPA project aims at formally relating these different theories for allowing users (i) to independently certify proofs carried out using the automatic proof tools developed for these formal methods and (ii) to transfer developments, including their proofs, carried out in one of these languages to another one. The objectives are to reinforce confidence in developments carried out using these methods and to enable interoperability between them. The foundation for achieving these goals lies in the encoding of the set theories in the Dedukti logical framework developed at Inria Saclay, which implements the $\lambda\Pi$-calculus modulo theory.

Keywords: B method, TLA$^+$, set theory, logical framework

More information: ICSPA Web site

10 Dissemination

Participants: Thomas Bagrel, Martin Bromberger, Horatiu Cirstea, Alessio Coltellacci, Marie Duflot-Kremer, Pascal Fontaine, Engel Lefaucheux, Sergueï Lenglet, Dominique Méry, Stephan Merz, Sibylle Möhle, Victor Roussanaly, Simon Schwarz, Sorin Stratulat, Thomas Sturm, Sophie Tourret, Vincent Trélat, Uwe Waldmann, Christoph Weidenbach.

10.1 Promoting scientific activities

10.1.1 Scientific events: organization

General chair, scientific chair

- Dominique Méry: 9th International Conference on Rigorous State-Based Methods (ABZ 2023) [44], 27th International Conference on Engineering of Complex Computer Systems (ICECCS 2023).
- Sorin Stratulat: 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2023).
- Thomas Sturm: SC-Square (8th International Workshop on Satisfiability Checking and Symbolic Computation), including edition of the proceedings [42].

Member of organizing committees

- Engel Lefaucheux: 8th International Workshop on Synthesis of Complex Parameters (SynCoP 2023, Paris, France)
- Stephan Merz: ETAPS Mentoring Workshop (Paris, France), 10th Workshop Formal Reasoning in Distributed Algorithms (FRIDA, L'Aquila, Italy), TLA$^+$ Community Meeting (Paris, France).

10.1.2 Scientific events: selection

Member of conference program committees

- Horatiu Cirstea: RuleML+RR (7th International Joint Conference on Rules and Reasoning), SLE (ACM SIGPLAN International Conference on Software Language Engineering).

- Pascal Fontaine: CADE (International Conference on Automated Deduction).
- Engel Lefaucheux: FORMATS (21st International Conference on Formal Modeling and Analysis of Timed Systems).
- Sergueï Lenglet: ICE (16th Interaction and Concurrency Experience).
- Dominique Méry: FMAS (Fifth Workshop on Formal Methods for Autonomous Systems), HEDA (3rd International Health Data Workshop), ICFEM (24rd International Conference on Formal Engineering Methods), MEDI (12th International Conference on Model and Data Engineering), TASE (17th Theoretical Aspects of Software Engineering Conference).
- Stephan Merz: ABZ (9th International Conference on Rigorous State-based Methods), AFADL (22èmes Journées Approches Formelles dans l'Assistance au Développement Logiciel), ICFEM 2023 (24th International Conference on Formal Engineering Methods), SEFM2023 (Software Engineering and Formal Methods 2023), VMCAI 2023 (Verification, Model Checking and Abstract Interpretation).
- Sorin Stratulat: SYNASC (25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing), ICEUTE (14th International Conference on EUropean Transnational Educational), CISIS (16th International Conference on Computational Intelligence in Security for Information Systems).
- Thomas Sturm: CASC (25th International Workshop on Computer Algebra in Scientific Computing).
- Sophie Tourret: IJCAI (32st International Joint Conference on Artificial Intelligence) and CADE (29th International Conference on Automated Deduction).

10.1.3 Journal

Member of editorial boards

- Dominique Méry is a member of the editorial boards of the journals *Formal Aspects of Computing* and *Science of Computer Programming*.
- Thomas Sturm is an editor of the *Journal of Symbolic Computation* and of *Mathematics in Computer Science*.
- Christoph Weidenbach is an editor of the *Journal of Automated Reasoning*.

Special issues edited

- Thomas Sturm has edited a special issue of the Springer journal Mathematics in Computer Science on Computer Algebra in Scientific Computing [43].

10.1.4 Invited talks

- Horatiu Cirstea and Thomas Sturm were invited plenary speakers and Stephan Merz gave an invited tutorial at SYNASC 2023.
- Stephan Merz gave an invited talk at the 2023 meeting of the working group on verification of GDR IM.
- Sophie Tourret gave an invited talk at the Vampire workshop 2023.

10.1.5 Leadership within the scientific community

- Pascal Fontaine is an elected CADE Trustee. In the COST action EuroProofNet, he was workgroup vice-leader until October 2023, and he is in the management committee as one Belgian representative. He was member of the committee for the William McCune PhD Award 2023.
- Dominique Méry is a member of the IFIP Working Group 1.3 on *Foundations of System Specifications*.
- Stephan Merz is a member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*.
- Thomas Sturm has become a member of the steering committee of the workshop series SC-Square (Satisfiability Checking and Symbolic Computation)
- Sophie Tourret is an elected CADE Trustee. She is also a member of the Association for Automated Reasoning (AAR) board of trustees as the editor of the AAR newsletter.

10.1.6 Research administration

- Marie Duflot-Kremer is a member of the jury of CAPES NSI, the French hiring exam for becoming a computer science teacher in secondary schools.
- Pierre-Étienne Moreau is the director of Télécom Nancy.

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- Master: Horatiu Cirstea, Advanced software engineering, 40 HETD, M2 Informatique, Université de Lorraine, France.
- Master: Horatiu Cirstea, Software engineering & Design patterns, 80 HETD, M1 informatique, Université de Lorraine, France.
- Master: Horatiu Cirstea, Software engineering, 40 HETD, 2A ENSEM, Université de Lorraine, France.
- Licence: Horatiu Cirstea, Algorithms and programming 3, 60 HETD, L2, Université de Lorraine, France.
- Licence: Marie Duflot-Kremer, Algorithms and programming 1, 80 HETD, L1, Université de Lorraine, France.
- Licence: Marie Duflot-Kremer, Individual support for algorithms and programming, 30 HETD, L1, Université de Lorraine, France.
- Master: Marie Duflot-Kremer, Using unplugged activities to pass on computer science concepts to students, master for future teachers.
- Licence : Marie Duflot-Kremer, data bases, 20 HETD, L2 and L3, Université de Lorraine
- Master: Marie Duflot-Kremer and Stephan Merz, Elements of model checking, 24 HETD, M2 Informatique, Université de Lorraine, France.
- Master: Marie Duflot-Kremer and Stephan Merz, Distributed algorithms, 30 HETD, M1 Informatique, Université de Lorraine, France.
- Classe préparatoire universitaire: Engel Lefaucheux, Algorithms and programming (2 and 3), 8 HETD, Université de Lorraine.
- Licence: Engel Lefaucheux, Algorithms and programming 2, 20 HETD, L2, Université de Lorraine.

- Classe préparatoire des INP: Engel Lefaucheux, Langages et Automates, 34.5 HETD, Université de Lorraine
- BUT 1: Sergueï Lenglet, Introduction to databases, 110 HETD, Université de Lorraine – IUT Charlemagne, France.
- BUT 1: Sergueï Lenglet, Exploitation of databases, 60 HETD, Université de Lorraine – IUT Charlemagne, France.
- BUT 2: Sergueï Lenglet, Functional programming, 24 HETD, Université de Lorraine – IUT Charlemagne, France.
- Master: Dominique Méry, Formal Modeling for Software-based Systems 40 HETD, M2 Informatique, Université de Lorraine, France.
- Master: Dominique Méry, Models and algorithms, M1 Telecom Nancy, 48 HETD, Université de Lorraine, France.
- Master: Dominique Méry, Formal Modeling for Software-based Systems, M2 Telecom Nancy, 24 HETD, Université de Lorraine, France.
- Master: Stephan Merz and Sophie Tourret, Secure Coding, M1 Mines Nancy, 26 HETD, Université de Lorraine, France
- Master: Sophie Tourret, Decision Procedures for Program Verification, M2 Informatique (academic year 2022-2023), Université de Lorraine, France.
- Master: Uwe Waldmann, Automated Reasoning, 60 HETD, Universität des Saarlandes, Germany.
- Master: Martin Bromberger, Sibylle Möhle, Simon Schwarz, Christoph Weidenbach, Automated Reasoning I, 60 HETD, Universität des Saarlandes, Germany.
- Master: Sorin Stratulat, Software design, 30 HETD, M2 Informatique, Université de Lorraine, France.
- Licence: Sorin Stratulat, Algorithms and programming, 105 HETD, L1 Informatique, Université de Lorraine, France.
- Licence: Sorin Stratulat, Logic for computer science, 26 HETD, L1 Informatique, Université de Lorraine, France.
- Licence: Victor Roussanaly, Data bases, L3 Polytech Nancy, 60 HETD, Université de Lorraine, France.
- Licence: Victor Roussanaly, Object-oriented programming, L3 Polytech Nancy, 42 HETD, Université de Lorraine, France.
- Master: Victor Roussanaly, Introduction to cryptography, M2 Polytech Nancy, 15 HETD, Université de Lorraine, France.
- Master: Thomas Sturm, Algorithmic Quantifier Elimination, 40 HETD, Universität des Saarlandes, Germany.

10.2.2 Supervision

- PhD: Rosalie Defourné, *Encoding TLA$^+$ Set Theory for Automatic Proof*, Université de Lorraine [45]. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, November 7, 2023.
- PhD: Dylan Marinho, *Detecting timing attacks using formal methods*, Université de Lorraine. Supervised by Étienne André, October 3, 2023.

- PhD: Fajar Haifani, *On a Notion of Abduction and Relevance for First-Order Logic Clause Sets*, MPI for Informatics, Saarland University, Sarrebruck, Allemagne. Supervised by Sophie Tourret and Christoph Weidenbach, March 9, 2023.

- PhD in progress: Thomas Bagrel, *Type systems for memory safety in functional programming languages*, Université de Lorraine (CIFRE with Tweag company). Supervised by Horatiu Cirstea, since April 2022.

- PhD in progress: Ghilain Bergeron, *Generating distributed programs from formal specifications*, INRIA. Supervised by Horatiu Cirstea and Stephan Merz, since October 2023.

- PhD in progress: Martin Desharnais, *Verification in Isabelle/HOL of automated reasoning results*, MPI for Informatics, Saarland University, Sarrebruck, Allemagne. Supervised by Jasmin Blanchette, Sophie Tourret and Christoph Weidenbach, since August 2021.

- PhD in progress: Hendrik Leidinger, *SCL in First-Order Logic with Equality*, Universität des Saarlandes. Supervised by Christoph Weidenbach, since August 2020.

- PhD in progress: Lorenz Leutgeb, *Reasoning with SCL*, Universität des Saarlandes. Supervised by Christoph Weidenbach, since October 2021.

- PhD in progress: Simon Schwarz, *Automatic Reasoning for Security*, Universität des Saarlandes. Supervised by Christoph Weidenbach, since October 2022.

- PhD in progress: Amine Snoussi, *Formal Reengineering of Communication Protocols for Controllers*. Université de Lorraine (CIFRE with Westinghouse France). Supervised by Marie Duflot-Kremer and Stephan Merz, since May 2023.

- PhD in progress: Vincent Trélat, *Higher-Order SMT Solving for Proof Obligations in Set Theory*. Université de Lorraine. Supervised by Stephan Merz and Sophie Tourret, since October 2023.

10.2.3 Juries

- Stephan Merz was a reviewer of the PhD theses of Samira Aït Bensaïd (Saclay), Jürgen König (Paderborn), Alexandrina Korneva (Saclay), and Gautier Raimondi (Rennes).

10.3 Popularization

10.3.1 Internal or external Inria responsibilities

- 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*, which supports projects on popularization activities.

- Marie Duflot-Kremer is a member of the Interstices editorial board, a Web site launched by Inria that publishes popularization articles.

- Christoph Weidenbach is the head of the steering committee of the German Computer Science Competition for High School Students (BWINF) and a co-organizer and the president of the jury of the final round that took place in Karlsruhe in September 2023. Thomas Sturm was a member of that jury.

- Thomas Sturm collaborated in the scientific part of the German team for the International Olympiad in Informatics (IOI).

10.3.2 Education

- Marie Duflot-Kremer gave a half day training with Universcience for secondary school teachers on artificial intelligence in February in Paris
- Marie Duflot-Kremer gave a one day training on resenting computer science concepts to secondary school computer science teachers in April in Besançon
- Marie Duflot-Kremer gave a talk at a seminar on graphs for secondary school math teachers in Paris in June

10.3.3 Interventions

- Marie Duflot-Kremer co-organized a one week seminar "Les Cigognes" for girls in secondary school to help them discover research in mathematics and computer science in Ramonchamp in October 2023
- Marie Duflot-Kremer co-organized "Journée Sciences et Médias", a one day conference on links beetween sciences and journalism in Paris in May 2023
- Marie Duflot-Kremer organized a stand at the “Fête de la science” event (October 2023), to present unplugged computer science activities to the general public with the help of bachelor and master students. She also gave an invited talk for Fête de la Science in Saint Dié.
- Marie Duflot-Kremer set up a stand at different scientific events, like "Nocturnes de l’histoire" and "semaine des mathématiques" in March, "matinées filles maths et sciences" in April, "Salon de la culture et des Jeux Mathématiques" in May
- Marie Duflot-Kremer took part in several meetings with secondary school students to present computer science and research within the *Chiche* program.

11 Scientific production

11.1 Major publications

[1] T. Bouton, D. C. B. de Oliveira, D. Déharbe and P. Fontaine. ‘veriT: an open, trustable and efficient SMT-solver’. In: *Proc. Conference on Automated Deduction (CADE).* Ed. by R. Schmidt. Vol. 5663. Lecture Notes in Computer Science. Montreal, Canada: Springer, 2009, pp. 151–156.

[2] M. Bromberger, T. Sturm and C. Weidenbach. ‘A complete and terminating approach to linear integer solving’. In: *Journal of Symbolic Computation* 100 (Sept. 2020), pp. 102–136. DOI: 10.1016/j.jsc.2019.07.021. URL: https://hal.inria.fr/hal-02397168.

[3] D. Cansell and D. Méry. ‘The Event-B Modelling Method - Concepts and Case Studies’. In: *Logics of Specification Languages.* Ed. by D. Bjoerner and M. Henson. Monographs in Theoretical Computer Science. Springer, Feb. 2008, pp. 33–140. URL: https://hal.inria.fr/inria-00579550.

[4] D. Cousineau, D. Doligez, L. Lamport, S. Merz, D. Ricketts and H. Vanzetto. ‘TLA+ Proofs’. In: *18th International Symposium On Formal Methods - FM 2012.* Ed. by D. Giannakopoulou and D. Méry. Vol. 7436. Lecture Notes in Computer Science. Paris, France: Springer, 2012, pp. 147–154.

[5] A. Dolzmann and T. Sturm. ‘Redlog: Computer algebra meets computer logic’. In: *ACM SIGSAM Bull.* 31.2 (1997), pp. 2–9.

[6] H. Errami, M. Eiswirth, D. Grigoriev, W. M. Seiler, T. Sturm and A. Weber. ‘Detection of Hopf bifurcations in chemical reaction networks using convex coordinates’. In: *Journal of Computational Physics* 291 (Mar. 2015), pp. 279–302. DOI: 10.1016/j.jcp.2015.02.050. URL: https://hal.archives-ouvertes.fr/hal-03044741.

[7] E. Kruglov and C. Weidenbach. ‘Superposition Decides the First-Order Logic Fragment Over Ground Theories’. In: *Mathematics in Computer Science* 6.4 (2012), pp. 427–456.

[8] S. Merz. 'The Specification Language TLA+'. In: *Logics of specification languages*. Ed. by D. Bjoerner and M. Henson. Monographs in Theoretical Computer Science. Springer, 2008, pp. 401–452. URL: https://hal.inria.fr/inria-00338330.

[9] T. Sturm and A. Tiwari. 'Verification and synthesis using real quantifier elimination'. In: *Proc. ISSAC 2011*. San Jose, United States: ACM Press, June 2011, p. 329. DOI: 10.1145/1993886.1993935. URL: https://hal.archives-ouvertes.fr/hal-03142063.

[10] C. Weidenbach, D. Dimova, A. Fietzke, M. Suda and P. Wischnewski. 'SPASS Version 3.5'. In: *22nd International Conference on Automated Deduction (CADE-22)*. Ed. by R. Schmidt. Vol. 5663. LNAI. Montreal, Canada: Springer, 2009, pp. 140–145.

11.2 Publications of the year

International journals

[11] É. André, E. Lefaucheux, D. Lime, D. Marinho and J. Sun. 'Configuring Timing Parameters to Ensure Execution-Time Opacity in Timed Automata'. In: *Electronic Proceedings in Theoretical Computer Science* 392 (31st Oct. 2023), pp. 1–26. DOI: 10.4204/eptcs.392.1. URL: https://hal.science/hal-04312156.

[12] A. Bentkamp, J. Blanchette, V. Nummelin, S. Tourret, P. Vukmirović and U. Waldmann. 'Mechanical Mathematicians'. In: *Communications of the ACM* 66.4 (23rd Mar. 2023), pp. 80–90. DOI: 10.1145/3557998. URL: https://inria.hal.science/hal-04298600.

[13] A. Bentkamp, J. Blanchette, S. Tourret and P. Vukmirović. 'Superposition for Higher-Order Logic'. In: *Journal of Automated Reasoning* 67.1 (21st Jan. 2023), p. 10. DOI: 10.1007/s10817-022-09649-9. URL: https://inria.hal.science/hal-04298616.

[14] J. Blanchette and P. Vukmirović. 'SAT-Inspired Higher-Order Eliminations'. In: *Logical Methods in Computer Science* 19.2 (2023). DOI: 10.48550/arXiv.2208.07775. URL: https://inria.hal.science/hal-04298561.

[15] G. Ebner, J. Blanchette and S. Tourret. 'Unifying Splitting'. In: *Journal of Automated Reasoning* 67.2 (28th Apr. 2023), p. 16. DOI: 10.1007/s10817-023-09660-8. URL: https://inria.hal.science/hal-04298584.

[16] V. Jain, U. Wetzker, V. Laxmi, M. S. Gaur, M. Mosbah and D. Méry. 'SAP: A Secure Low-Latency Protocol for Mitigating High Computation Overhead in WI-FI Networks'. In: *IEEE Access* 11 (2023), pp. 84620–84635. DOI: 10.1109/ACCESS.2023.3302529. URL: https://inria.hal.science/hal-04183865.

[17] H. Leidinger and C. Weidenbach. 'SCL(EQ): SCL for First-Order Logic with Equality'. In: *Journal of Automated Reasoning* 67.3 (30th June 2023), p. 22. DOI: 10.1007/s10817-023-09673-3. URL: https://inria.hal.science/hal-04313698.

[18] I. Mendil, Y. Aït-Ameur, N. K. Singh, G. Dupont, D. Méry and P. Palanque. 'Formal domain-driven system development in Event-B: Application to interactive critical systems'. In: *Journal of Systems Architecture* 135 (Feb. 2023), p. 102798. DOI: 10.1016/j.sysarc.2022.102798. URL: https://inria.hal.science/hal-03904803.

[19] L. Penet de Monterno, B. Charron-Bost and S. Merz. 'Synchronization modulo P in dynamic networks'. In: *Theoretical Computer Science* 942 (Jan. 2023), pp. 200–212. DOI: 10.1016/J.TCS.2022.11.033. URL: https://hal.science/hal-04289753.

[20] S. Stratulat. 'Mechanical certification of FOLID cyclic proofs'. In: *Annals of Mathematics and Artificial Intelligence* 95.5 (16th Feb. 2023), pp. 651–673. DOI: 10.1007/s10472-023-09832-7. URL: https://inria.hal.science/hal-03993176.

[21] P. Vukmirović, J. Blanchette and M. J. H. Heule. 'SAT-Inspired Eliminations for Superposition'. In: *ACM Transactions on Computational Logic* 24.1 (18th Jan. 2023), pp. 1–25. DOI: 10.1145/3565366. URL: https://inria.hal.science/hal-04298574.

International peer-reviewed conferences

[22] É. André, E. Lefaucheux and D. Marinho. 'Expiring opacity problems in parametric timed automata'. In: *2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS)*. 2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS). Toulouse, France: IEEE; IEEE, 14th June 2023, pp. 89–98. DOI: 10.1109/ICECCS59891.2023.00020. URL: https://hal.science/hal-04151207.

[23] ***Best Paper***
J. Blanchette, Q. Qiu and S. Tourret. 'Verified Given Clause Procedures'. In: *Automated Deduction – CADE 29*. CADE-29. Vol. 14132. Lecture Notes in Computer Science. Rome, Italy: Springer Nature Switzerland, 2nd Sept. 2023, pp. 61–77. DOI: 10.1007/978-3-031-38499-8_4. URL: https://inria.hal.science/hal-04298505.

[24] Y. Briefs, H. Leidinger and C. Weidenbach. 'KBO Constraint Solving Revisited'. In: Frontiers of Combining Systems - 14th International Symposium. Vol. 14279. Lecture Notes in Computer Science. Prague (CZ), Czech Republic: Springer Nature Switzerland, 2023, pp. 81–98. DOI: 10.1007/978-3-031-43369-6_5. URL: https://inria.hal.science/hal-04313806.

[25] M. Bromberger, M. Desharnais and C. Weidenbach. 'An Isabelle/HOL Formalization of the SCL(FOL) Calculus'. In: *Springer LNCS*. Automated Deduction - CADE 29 - 29th International Conference on Automated Deduction. Vol. 14132. Lecture Notes in Computer Science. Rome (IT), Italy: Springer Nature Switzerland, 2nd Sept. 2023, pp. 116–133. DOI: 10.1007/978-3-031-38499-8_7. URL: https://inria.hal.science/hal-04313741.

[26] M. Bromberger, C. Jain and C. Weidenbach. 'SCL(FOL) Can Simulate Non-Redundant Superposition Clause Learning'. In: Automated Deduction - CADE 29 - 29th International Conference on Automated Deduction. Vol. 14132. Lecture Notes in Computer Science. Rome (IT), Italy: Springer Nature Switzerland, 2023, pp. 134–152. DOI: 10.1007/978-3-031-38499-8_8. URL: https://inria.hal.science/hal-04313799.

[27] M. Bromberger, L. Leutgeb and C. Weidenbach. 'Symbolic Model Construction for Saturated Constrained Horn Clauses'. In: Frontiers of Combining Systems - 14th International Symposium, FroCoS 2023. Vol. 14279. Lecture Notes in Computer Science. Prague (CZ), Czech Republic: Springer Nature Switzerland, 2023, pp. 137–155. DOI: 10.1007/978-3-031-43369-6_8. URL: https://inria.hal.science/hal-04313817.

[28] M. Bromberger, S. Schwarz and C. Weidenbach. 'Exploring Partial Models with SCL'. In: Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning. Vol. 94. Manizales, Colombia, 4th June 2023, pp. 48–22. DOI: 10.29007/8BR1. URL: https://inria.hal.science/hal-04313819.

[29] H. Cirstea and S. Merz. 'Extending PlusCal for Modeling Distributed Algorithms'. In: *Lecture Notes in Computer Science*. 18th International Conference on Integrated Formal Methods (iFM 2023). Vol. 14300. Leiden, Netherlands: Springer, Nov. 2023, pp. 321–340. DOI: 10.1007/978-3-031-47705-8_17. URL: https://inria.hal.science/hal-04293883.

[30] M. D'aquin, R. Bunoiu, H. Cirstea, M. Lenczner, J. Lieber and F. Zamkotsian. 'Combining representation formalisms for reasoning upon mathematical knowledge'. In: *K-CAP '23: Proceedings of the 12th Knowledge Capture Conference 2023*. K-CAP '23: Knowledge Capture Conference 2023. Pensacola FL USA, United States: ACM, 2023, pp. 180–187. DOI: 10.1145/3587259.3627549. URL: https://hal.science/hal-04315073.

[31] R. Defourné. 'Encoding TLA$^+$ Proof Obligations Safely for SMT'. In: 9th International Conference on Rigorous State-Based Methods (ABZ 2023). Vol. 14010. Lecture Notes in Computer Science. Nancy, France: Springer Nature Switzerland, 15th May 2023, pp. 88–106. DOI: 10.1007/978-3-031-33163-3_7. URL: https://hal.science/hal-04299295.

[32] M. Dvorak and J. Blanchette. 'Closure Properties of General Grammars – Formally Verified'. In: *14th International Conference on Interactive Theorem Proving (ITP 2023)*. ITP 2023. Białystok, Poland: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. DOI: 10.4230/LIPIcs.ITP.2023.15. URL: https://inria.hal.science/hal-04298533.

[33] N. Faroß and S. Schwarz. 'Gröbner Bases for Boolean Function Minimization'. In: Proceedings of the 8th SC-Square Workshop. Tromsø, Norway, 28th July 2024. URL: https://inria.hal.science/hal-04315477.

[34] A. Goel, S. Merz and K. A. Sakallah. 'Towards an Automatic Proof of the Bakery Algorithm'. In: Formal Techniques for Distributed Objects, Components, and Systems. FORTE 2023. Vol. 13910. Lecture Notes in Computer Science. Lisbon, Portugal: Springer Nature Switzerland, June 2023, pp. 21–28. DOI: 10.1007/978-3-031-35355-0_2. URL: https://inria.hal.science/hal-04135287.

[35] E. Lefaucheux, J. Ouaknine, D. Purser and M. Sharifi. 'Model Checking Linear Dynamical Systems under Floating-point Rounding'. In: Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2023). Vol. 13993. Lecture Notes in Computer Science. Paris, France: Springer Nature Switzerland, 22nd Apr. 2023, pp. 47–65. DOI: 10.1007/978-3-031-30823-9_3. URL: https://inria.hal.science/hal-03843471.

[36] I. Mendil, P. Riviere, Y. Aït-Ameur, N. K. Singh, D. Méry and P. Palanque. 'Non-Intrusive Annotation-Based Domain-Specific Analysis to Certify Event-B Models Behaviours'. In: *2022 29th Asia-Pacific Software Engineering Conference (APSEC)*. 2022 29th Asia-Pacific Software Engineering Conference (APSEC). Japan, Japan: IEEE, Feb. 2023, pp. 129–138. DOI: 10.1109/APSEC57359.2022.00025. URL: https://hal.science/hal-04316165.

[37] S. Möhle. 'An Abstract CNF-to-d-DNNF Compiler Based on Chronological CDCL'. In: Frontiers of Combining Systems - 14th International Symposium, FroCoS 2023. Vol. 14279. Lecture Notes in Computer Science. Prague, Czech Republic: Springer Nature Switzerland, 2023, pp. 195–213. DOI: 10.1007/978-3-031-43369-6_11. URL: https://inria.hal.science/hal-04315486.

[38] V. Nummelin, J. Blanchette and S. Dahmen. 'Recurrence-Driven Summations in Automated Deduction'. In: *Frontiers of Combining Systems. FroCoS 2023.* FroCoS 2023. Vol. 14279. Lecture Notes in Computer Science. Prague, Czech Republic: Springer Nature Switzerland, 13th Sept. 2023, pp. 23–40. DOI: 10.1007/978-3-031-43369-6_2. URL: https://inria.hal.science/hal-04298450.

[39] A. Plank, S. Möhle and M. Seidl. 'Enumerative Level-2 Solution Counting for Quantified Boolean Formulas'. In: 29th International Conference on Principles and Practice of Constraint Programming - CP 2023. Toronto, Canada: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. DOI: 10.4230/LIPIcs.CP.2023.49. URL: https://inria.hal.science/hal-04327008.

[40] P. Vukmirović, J. Blanchette and S. Schulz. 'Extending a High-Performance Prover to Higher-Order Logic'. In: *Tools and Algorithms for the Construction and Analysis of Systems. TACAS 2023.* TACAS 2023. Vol. 13994. Lecture Notes in Computer Science. Paris, France: Springer Nature Switzerland, 20th Apr. 2023, pp. 111–129. DOI: 10.1007/978-3-031-30820-8_10. URL: https://inria.hal.science/hal-04298635.

National peer-reviewed Conferences

[41] T. Bagrel. 'Destination-passing style programming: a Haskell implementation'. In: 35es Journées Francophones des Langages Applicatifs (JFLA 2024). Saint-Jacut-de-la-Mer, France, Jan. 2024. URL: https://inria.hal.science/hal-04406360.

Edition (books, proceedings, special issue of a journal)

[42] E. Ábrahám and T. Sturm, eds. *Satisfiability Checking and Symbolic Computation 2023: Proceedings of the 8th SC-Square Workshop, co-located with the 48th International Symposium on Symbolic and Algebraic Computation (ISSAC 2023)*. Vol. 3455. CEUR-WS.org, 15th Aug. 2023. URL: https://inria.hal.science/hal-04307435.

[43] *Computer Algebra in Scientific Computing 2022* 17.3-4 (Sept. 2023). URL: https://hal.science/hal-03833048.

[44] U. Glässer, J. Creissac Campos, D. Méry and P. Palanque, eds. *Rigorous State-Based Methods - 9th International Conference, ABZ 2023, Nancy, France, May 30 - June 2, 2023, Proceedings.* Vol. 14010. Springer Nature Switzerland; Springer, 2023. DOI: 10.1007/978-3-031-33163-3. URL: https://inria.hal.science/hal-04183902.

Doctoral dissertations and habilitation theses

[45] A. Defourné. 'Encoding TLA+'s Set Theory for Automated Theorem Provers'. Université de lorraine, 7th Nov. 2023. URL: https://theses.hal.science/tel-04408971.

Reports & preprints

[46] M. Bromberger, C. Jain and C. Weidenbach. *SCL(FOL) Can Simulate Non-Redundant Superposition Clause Learning*. Max-Planck-Institut für Informatik, Saarbrücken, Germany, 22nd May 2023. DOI: 10.48550/ARXIV.2305.12926. URL: https://inria.hal.science/hal-04315211.

[47] M. Bromberger, L. Leutgeb and C. Weidenbach. *Symbolic Model Construction for Saturated Constrained Horn Clauses*. Max-Planck-Institut für Informatik, Saarbrücken, Germany, 24th July 2023. DOI: 10.48550/ARXIV.2305.05064. URL: https://inria.hal.science/hal-04315210.

[48] M. Bromberger, S. Schwarz and C. Weidenbach. *SCL(FOL) Revisited*. Max-Planck-Institut für Informatik, Saarbrücken, Germany, 14th Feb. 2023. DOI: 10.48550/ARXIV.2302.05954. URL: https://inria.hal.science/hal-04314223.

[49] Z. Cheng and D. Méry. *A Static Checker for Reference Tracking Systems via Laplace Transform and Transfer Functions*. 5th July 2023. URL: https://hal.science/hal-04152829.

[50] Z. Cheng and D. Méry. *From System Events to Software Operations for Refinement-based Modeling of Hybrid Systems **. 28th Aug. 2023. URL: https://hal.science/hal-04189025.

[51] E. Lefaucheux. *Testing equality of parametric semi-linear sets*. 27th July 2023. URL: https://inria.hal.science/hal-04172593.

Other scientific publications

[52] M. England, F. Boulier, T. Sadykov and T. Sturm. *Foreword. Special issue on CASC 2022*. 1st Sept. 2023. DOI: 10.1007/s11786-023-00565-8. URL: https://inria.hal.science/hal-04307396.

11.3 Cited publications

[53] N. Kruff, C. Lüders, O. Radulescu, T. Sturm and S. Walcher. 'Algorithmic Reduction of Biological Networks with Multiple Time Scales'. In: *Mathematics in Computer Science* 15.3 (Sept. 2021), pp. 499–534. DOI: 10.1007/s11786-021-00515-2. URL: https://hal.archives-ouvertes.fr/hal-03438176.

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

[55] R. Alur, T. A. Henzinger and M. Y. Vardi. 'Parametric real-time reasoning'. In: *Proc. 25th Annual ACM Symp. Theory of Computing*. Ed. by S. R. Kosaraju, D. S. Johnson and A. Aggarwal. San Diego, CA, USA: ACM, 1993, pp. 592–601.

[56] É. André. 'IMITATOR 3: Synthesis of Timing Parameters Beyond Decidability'. In: *Proc. 33rd Intl. Conf. Computer-Aided Verification (CAV 2021)*. 2021, pp. 552–565.

[57] L. Bachmair and H. Ganzinger. 'Rewrite-Based Equational Theorem Proving with Selection and Simplification'. In: *Journal of Logic and Computation* 4.3 (1994), pp. 217–247.

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

[59] C. Barrett, R. Sebastiani, S. A. Seshia and C. Tinelli. 'Satisfiability Modulo Theories'. In: *Handbook of Satisfiability*. Ed. by A. Biere, M. Heule, H. van Maaren and T. Walsh. Vol. 185. Frontiers in Artificial Intelligence and Applications. IOS Press, Feb. 2009. Chap. 26, pp. 825–885.

[60] M. Biernacka, D. Biernacki, S. Lenglet and A. Schmitt. 'Non-Deterministic Abstract Machines'. In: *33rd International Conference on Concurrency Theory (CONCUR 2022)*. Ed. by B. Klin, S. Lasota and A. Muscholl. Vol. 243. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 7:1–7:24. DOI: 10.4230/LIPIcs.CONCUR.2022.7. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.CONCUR.2022.7.

[61] M. Bromberger, I. Dragoste, R. Faqeh, C. Fetzer, M. Krötzsch and C. Weidenbach. 'A Datalog Hammer for Supervisor Verification Conditions Modulo Simple Linear Arithmetic'. In: *13th Intl. Symp. Frontiers of Combining Systems (FroCoS 2021)*. Vol. 12941. Lecture Notes in Computer Science. Springer, 2021, pp. 3–24.

[62] M. Bromberger, A. Fiori and C. Weidenbach. 'Deciding the Bernays-Schoenfinkel Fragment over Bounded Difference Constraints by Simple Clause Learning over Theories'. In: 22nd Intl. Conf. Verification, Model Checking, and Abstract Interpretation (VMCAI 2021). Vol. 12597. Lecture Notes in Computer Science. Springer, 2021, pp. 511–533.

[63] E. Feliu, O. Henriksson and B. Pascual-Escudero. *Dimension and Degeneracy of Solutions of Parametric Polynomial Systems Arising from Reaction Networks*. arXiv 2304.02302. 2023.

[64] F. Haifani, S. Tourret and C. Weidenbach. 'Generalized Completeness for SOS Resolution and its Application to a New Notion of Relevance'. In: *28th Intl. Conf. Automated Deduction (CADE 28)*. Vol. 12699. Lecture Notes in Computer Science. Springer, 2021, pp. 327–343.

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

[66] N. Le Novere, B. Bornstein, A. Broicher, M. Courtot, M. Donizelli, H. Dharuri, L. Li, H. Sauro, M. Schilstra, B. Shapiro et al. 'BioModels Database: A Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems'. In: *Nucleic acids res.* 34.suppl_1 (Jan. 2006), pp. D689–D691. DOI: 10.1093/nar/gkj092.

[67] H. Lee and A. Lao. 'Transmission Dynamics and Control Strategies Assessment of Avian Influenza A (H5N6) in the Philippines'. In: *Infectious Disease Modelling* 3 (2018), pp. 35–59. DOI: 10.1016/j.idm.2018.03.004.

[68] R. A. E. Manssour, A. Sattelberger and B. T. Tabuguia. *D-Algebraic Functions*. arXiv 2301.02512. 2023.

[69] C. Morgan. *Programming from Specifications*. 2nd edition. Prentice Hall, 1998.

[70] S. Stratulat. 'Mechanically Certifying Formula-based Noetherian Induction Reasoning'. In: *Journal of Symbolic Computation* 80, Part I (2017), pp. 209–249. DOI: 10.1016/j.jsc.2016.07.014. URL: https://hal.science/hal-01590649.

[71] J. Xu, S. Du, J. Yang, X. Ding, J. Paisley and D. Zeng. *Double Normalizing Flows: Flexible Bayesian Gaussian Process ODEs Learning*. arXiv:2309.09222. 2023.