RESEARCH CENTRE

Inria Nancy - Grand Est Center

IN PARTNERSHIP WITH:

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

2022

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

Inria

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]
- Ioannis Filippidis [INRIA, Starting Research Position, until Aug 2022]
- 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

- Étienne André [UL, Professor, until Aug 2022, HDR]
- Horatiu Cirstea [UL, Professor, HDR]
- Marie Duflot-Kremer [UL, Associate Professor]
- Sergueï Lenglet [UL, Associate Professor]
- Pierre-Étienne Moreau [UL, Professor, HDR]
- Victor Roussanaly [UL, ATER, from Oct 2022]
- Sorin Stratulat [UL, Associate Professor, HDR]

Post-Doctoral Fellows

- Martin Bromberger [Max Planck Society]
- Zheng Cheng [UL]
- Sibylle Möhle [Max Planck Society, from Aug 2022]
- Hamid Rahkooy [Max Planck Society, until Oct 2022]
- Guillaume Verdier [UL]

PhD Students

- Thomas Bagrel [UL, CIFRE, from Apr 2022]
- Rosalie Defourne [UL, ATER, from Sep 2022]
- Rosalie Defourne [INRIA, until Aug 2022]
- Martin Desharnais [Max Planck Society]
- Fajar Haifani [Max Planck Society]
- Hendrik Leidinger [Max Planck Society]
- Lorenz Leutgeb [Max Planck Society]
- Dylan Marinho [UL]
- Hans-Jörg Schurr [UL, ATER, until Aug 2022]
- Simon Schwarz [Max Planck Society, from Oct 2022]

Technical Staff

- Benjamin Loillier [INRIA, Engineer]

Interns and Apprentices

- Shapagat Bolat [Inria, Intern, from Mar 2022]
- Florent Krasnopol [Inria, Intern, from Jun 2022 until Jul 2022]
- Nurgul Osmonova [Inria, Intern, from Mar 2022 until Jun 2022]
- Vincent Trélat [UL, Intern, until Jul 2022]
- Zunaira Zaman [Inria, Intern, from Mar 2022 until Jun 2022]

Administrative Assistants

- Juline Brevillet [UL, from May 2022]
- Antoinette Courrier [UL, until Apr 2022]
- Sophie Drouot [INRIA]

Visiting Scientist

- Marian Ileana [University of Pitesti, from Nov 2022]

External Collaborators

- Jasmin Blanchette [Free University of Amsterdam, until Mar 2022]
- Pascal Fontaine [University of Liège, HDR]

2 Overall objectives

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

The objectives of VeriDis are to contribute to advances in verification techniques, including automated and interactive theorem proving, and to make them available for the development and analysis of concurrent and distributed algorithms and systems, based on mathematically precise and practically applicable development methods. The techniques that we develop are intended to assist designers of algorithms and systems in carrying out formally 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 (fragments of arithmetic, set theory etc.) 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, 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.

[1] Susceptible – Exposed – Infectious – Removed

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 [67], 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 [73, 72].

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$^+$ [81] 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* [64, 68, 84] 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

[2] conflict-driven clause learning
[3] Satisfiability Modulo Theories [69]

establish precise proof obligations, many of which can be discharged by automatic tools. This requires taking into account specific characteristics of certain classes of systems and tailoring the model to concrete computational models. Our research in this area is supported by carrying out case studies for academic and industrial developments. This activity benefits from and influences the development of our proof tools.

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

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

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 [82] 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 [83]. 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

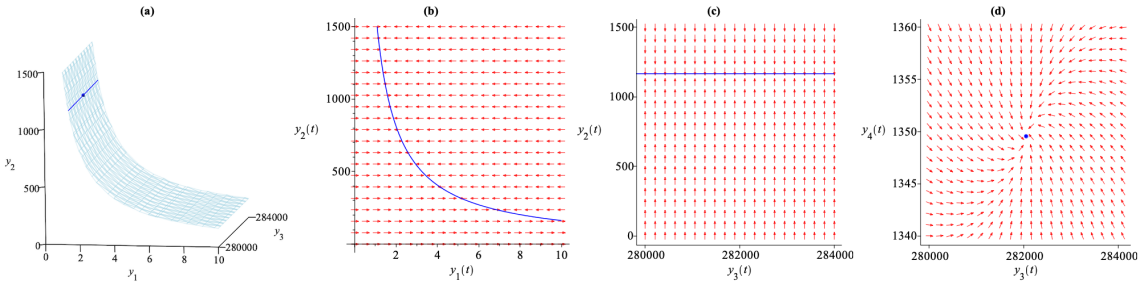

Figure 1: Illustration of the analysis of an epidemic model of avian Influenza A.

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 $\mathscr{M}_2$, 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 [61]. 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 underly mobile and low-power computing or overlay networks, peer-to-peer networks that provide services for telecommunication, or cloud computing services. Computing infrastructure must be highly available and is ideally invisible to the end user, therefore correctness is crucial. One should note that standard problems of distributed computing such as consensus, group membership or leader election have to be reformulated for the dynamic context of these modern systems. We are not ourselves experts in the design of distributed algorithms, but we work together with domain experts on designing formal models of these protocols, and on verifying their properties. These cooperations help us focus on concrete algorithms and ensure that our work is relevant to the distributed algorithm community.

Our work on symbolic procedures for solving polynomial constraints finds applications beyond verification. In particular, we have been working in interdisciplinary projects with researchers from mathematics, computer science, 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 [65]) 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 (cf. section 9.1). Foundational decidability results [11, 28] 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 [66].

5 Highlights of the year

5.1 Awards

Alexander Bentkamp, a former PhD student at VU Amsterdam co-supervised by Jasmin Blanchette and Uwe Waldmann, received the Ackermann award of the European Association for Computer Science Logic, the Beth Award of the Association for Logic, Language and Information, the McCune award at CADE, and the dissertation award of the Institute for Programming Research and Algorithmics (IPA) for his PhD thesis [70].

Fajar Haifani received the DL 2022 Best Student Paper Award for the paper "Connection-minimal Abduction in EL via translation to FOL" [43].

Sophie Tourret has been recognized as a distinguished PC member of IJCAI-ECAI 2022, for the quality of her reviews (ranked in the top 3%).

6 New software and platforms

6.1 New software

6.1.1 IMITATOR

Name: IMITATOR

Keywords: Verification, Parametric model, Parameter synthesis, Model Checking, Model Checker, Timed automata

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

News of the Year: Two new heuristics (merging and zone extrapolation) were implemented, with very interesting results in terms of efficiency. A very large extension of the syntax was performed, allowing new types (basic types, but also lists, arrays, stacks...), user defined functions over these types, etc. New applications to cybersecurity were analyzed.

URL: https://www.imitator.fr/

Publications: hal-03320626, hal-03772708, hal-00785289, hal-02153214, hal-02153342, hal-01961496

Contact: Etienne Andre

Participants: Etienne Andre, Jaime Eduardo Arias Almeida

Partner: Loria

6.1.2 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 methods from symbolic computation by supplying more than 100 functions on first-order formulas.

Redlog generally works with interpreted first-order logic in contrast to free first-order logic. Each first-order formula in Redlog must exclusively contain atoms from one particular Redlog-supported theory, which corresponds to a choice of admissible functions and relations with fixed semantics. Redlog-supported theories include Nonlinear Real Arithmetic (Real Closed Fields), Presburger Arithmetic, Parametric QSAT (quantified satisfiability solving), and many more.

News of the Year: Parts of the Redlog code are more than 25 years old now. Version 1 of the underlying computer algebra system Reduce has been published even more than 50 years ago. In 2018 we therefore started to go for major revisions and improvements of Redlog's software architecture, which are still under way.

Our principal design goal is more simplicity for the sake of better long-term maintainability. Technically we now favor keeping state spaces in explicit mutable data structures rather than on the recursion stack. Although not directly supported by the underlying Lisp system, we use object oriented ideas and approaches to the extent possible.

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

6.1.3 REDUCE F5

Name: A REDUCE Package for Computing Gröbner Bases Using F5

Keywords: Symbolic computation, Gröbner bases

Functional Description: F5 is a package for the computation of Gröbner Bases using Faugère's F5 algorithm. It uses multivariate rational functions as the coefficient field. The package is compatible with existing REDUCE term orderings, as used with the GROEBNER package and the CGB package.

From an Automated Reasoning viewpoint this offers a decision procedure for algebraically closed fields (including the complex numbers). The code has been designed with possible future generalization to the formal treatment of the coefficients as parameters and comprehensive Gröbner Bases in mind, which, in combination with other components of REDUCE, namely Redlog, would yield quantifier elimination and decision procedures for algebraically closed fields (including the complex numbers) as well as for real closed fields (including the real numbers).

News of the Year: REDUCE F5 has been launched as a part of the regular REDUCE distribution on SourceForge.

URL: https://sourceforge.net/p/reduce-algebra/code/HEAD/tree/trunk/packages/f5/

Contact: Thomas Sturm

Participants: Alexander Demin, Thomas Sturm, Hamid Rahkooy

6.1.4 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. In preparation is a solver SPASS-SPL for a fragment we call SUPERLOG, which is the first-order Bernays Schoenfinkel class extended with linear arithmetic.

News of the Year: We now published the SAT solver SPASS-SAT and the propositional CNF converter SPASS-CNF.

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

6.1.5 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).

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

Contact: Sorin Stratulat

6.1.6 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.

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

Contact: Stephan Merz

Participants: Damien Doligez, Stephan Merz, Ioannis Filippidis

Partner: Microsoft

6.1.7 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: Efforts in 2022 have been focused on higher-order logic, and better proof production. Achievements in 2022 are essentially around proof production, which makes veriT particularly suitable for integration within skeptical proof assistants. Code refactoring is envisioned for the future, to better accommodate the role of the solver as a platform for testing new ideas. Even if the veriT solver participated in the SMT competition SMT-COMP 2022 like previous years, the

improvement on this incarnation of the solver have been minimal, and the performances have not improved compared to 2021.

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*.

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

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

Contact: Pascal Fontaine

Participants: Pascal Fontaine, Hans-Jörg Schurr, 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: ODEbase has emerged from an internal repository of the SYMBIONT Project (9.1.1). In 2022, it has been been launched at its own domain and publicly announced in a journal publication [18]. ODEbase comprises 662 models at the time of writing.

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: Jasmin Christian Blanchette, Martin Bromberger, Rosalie Defourné, Martin Desharnais, Ioannis Filippidis, Pascal Fontaine, Fajar Haifani, Hendrik Leidinger, Lorenz Leutgeb, Stephan Merz, Sibylle Möhle, Hans-Jörg Schurr, Simon Schwarz, Sorin Stratulat, 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. This work is still ongoing. We are working on an encoding of the CCFV higher-order problem into a set of SAT constraints. In 2020, we concentrated our efforts mainly on the theory, to prove the soundness and completeness of our approach. In 2021, as a first step towards an implementation, we designed precise pseudo-code for all elements of CCFV computation. In 2022, these algorithms were implemented in a C++ library, and they were tested on benchmarks from the SMT-lib collection. Future works involve thorough testing of the library, improvements, and release under an open-source permissive license.

Proofs for SMT. We previously developed a framework for processing formulas in automatic theorem provers, with generation of detailed proofs that can be checked by external tools, including skeptical proof assistants. The main components are a generic contextual recursion algorithm and an extensible set of inference rules. Clausification, Skolemization, theory-specific simplifications, and expansion of 'let' expressions are instances of this framework. With suitable data structures, proof generation adds only a linear-time overhead, and proofs can be checked in linear time. We implemented the approach in the SMT solver veriT. This allowed us to dramatically simplify the code base while increasing the number of problems for which detailed proofs can be produced. Our publication at CADE 2021 [88] demonstrates the excellent results of our approach, building on our previous work on proof formats for SMT and on proof reconstruction within the proof assistant Isabelle/HOL (e.g., [76]). Our proof format was moreover the basis for the standard Alethe format [87], which is now getting adopted by the community. In 2022, the format was further polished, also to accommodate the needs of other SMT solvers adopting the format.

This is one of the main contributions, besides techniques for quantifier instantiation and strategy learning, of Hans-Jörg Schurr's PhD thesis [52], defended in October 2022. During his post-doctoral research in the cvc5 team (Iowa and Stanford), Hans-Jörg Schurr will continue to work notably on proofs for SMT.

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 issue 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, is working on an instantiation of this framework for the superposition calculus as formalized by Nicolas Peltier in the Archive of Formal Proof, a repository of Isabelle/HOL mechanizations [85]. We also made progress on the Isabelle/HOL mechanization of the framework with clause splitting. Florent Krasnopol, a student of ENS Saclay, contributed to this endeavor for his first-year internship. The two last pieces of work are still ongoing, while the first one has been completed in 2022 and a conference submission is being prepared on it.

Efficient implementation of superposition for higher-order logic. In previous work, we designed a superposition calculus for higher-order logic and implemented it in a prototype called Zipperposition. Now, we designed and developed an extension of the high-performance first-order prover E [86] with this higher-order calculus. When extending E, we used the extensive experience with Zipperposition to choose a set of effective rules that could easily be retrofitted into an originally first-order prover. Challenges included accommodating λ-terms in E's term representation, extending E by a higher-order unification procedure that can return multiple unifiers instead of a single one, and adapting the superposition rule to higher-order logic. A guiding principle for the design of our extension was *gracefulness*: we made sure that our changes do not impact the strong first-order performance of E.

E helped us win the first-place trophy at the CADE ATP System Competition (CASC), ahead of Zipperposition, in the Sledgehammer division of the 2022 edition of the competition. This confirms that an optimized implementation inside a competitive prover such as E, SPASS, or Vampire can outperform existing higher-order automated provers. A publication on this research has just been accepted and will be presented at the TACAS 2023 conference.

Relevance of clauses for resolution. A clause is relevant for a refutation with respect to an unsatisfiable clause set if it occurs in all refutation proofs. It is semi-relevant if it occurs in at least one refutation proof. We have shown that the question whether a clause C is semi-relevant can be reduced to the question whether there exists a set-of-support (SOS) refutation whose set of support is the singleton $\{C\}$ [77]. To this end we generalized and finalized the well-known completeness result on SOS resolution [80]: SOS resolution is complete if and only if there exists a resolution refutation with one of the clauses out of the SOS. The notion of semi-relevance is in particular useful to explain the contribution of a clause or formula to a specific consequence. For independent clause sets this syntactic notion of semi-relevance has a semantic counterpart. A clause is semantically semi-relevant if it contributes to the overall set of conflict literals [44]. A conflict literal for some clause set is a literal where the literal and its negation are both consequences of satisfiable subsets of the clause set.

SCL for first-order logic with and without equality. We previously showed that the SCL (Clause Learning from Simple Models) calculus can be successfully applied through a Datalog hammer to first-order logic modulo theories [34]. It was on open question whether SCL can be effectively extended to first-order logic with equality. It requires an interpretation of the simple ground model assumption with respect to equality. We could show that all syntactic operations on the simple ground model assumption can be lifted modulo equality and eventually be incorporated in a sound, complete, and effective SCL calculus for full first-order logic with equality [47]. In parallel, a mechanization effort for SCL without equality in Isabelle/HOL has been started by Martin Desharnais, as part of his PhD.

Certification of FOL_{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 (FOL_{ID}), cyclic proofs can be built automatically by the CYCLIST prover, but their implementations are error-prone and the human validation may be tedious.

We developed techniques for checking, using Coq, the cyclic proofs produced by E-CYCLIST (cf. section 6.1), an extension of the CYCLIST theorem prover, relying on the general Noetherian induction principle that is available in Coq. In this way, cyclic FOL_{ID} proofs can be certified mechanically and automatically. Moreover, Coq users can directly perform cyclic induction reasoning. We successfully certified the E-CYCLIST proofs of the FOL_{ID} examples included in the official distribution of the release CSL-LICS14 of CYCLIST, as well as the 2-Hydra problem. The datasets generated during and/or analyzed during the current study, in particular, the full Coq specifications and proof scripts, are archived and made available on the Web.

Abduction for Description Logics. Abduction is the process of explaining new observations using background knowledge. It is central to knowledge discovery and knowledge processing and has been intensely studied in various domains such as artificial intelligence, philosophy and logic. In the description logic literature, abduction has received little attention, despite being recognised as important for ontology repair, query update and matchmaking.

As part of his PhD, Fajar Haifani finalized in 2022 a technique for abduction in the lightweight description logic $\mathscr{EL}$, that specializes in representing subset inclusions and membership. His approach consists in translating the problem to first-order logic to harness the power of the automated deduction tool SPASS to produce prime implicates, i.e., most general consequences, from which the solutions of the abduction problem can be reconstructed. The complete method, including an experimental evaluation, has been presented at IJCAR 2022 [42], and as an extended abstract at the DL workshop 2022 [43], where it received the best student paper award.

In a joint work with P. Koopmann (TU Dresden), W. Del-Pinto and R. Schmidt (Univ. Manchester), we are also working on an extended version of an earlier work on abduction in the expressive description logic $\mathscr{ALC}$ [79].

Proofs for TLA$^+$. In her PhD work, Rosalie Defourné investigates 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 current SMT backend of the TLA$^+$ Proof System TLAPS heavily relies on simplification techniques such as rewriting and term abstraction in order to transform TLA$^+$ proof obligations into the input language of SMT solvers. Although these techniques are quite powerful, they destroy the structure of the original formulas, and the inherent danger of unsoundness or loss of termination makes it hard to maintain them. The approach proposed in this work consists in axiomatizing TLA$^+$ set theory in the SMT language and in adding suitable triggers that guide the instantiation of quantifiers. It has been validated over an extensive corpus of existing TLA$^+$ proofs, demonstrating that its effectiveness is at least comparable with that of the existing backend, and sometimes outperforms it significantly. Moreover, the addition of trigger annotations cannot invalidate the soundness of the axiomatization, which should help in longer-term maintenance of the proof backend. A paper describing this work has been submitted for publication, and the code is intended to replace the existing SMT backend of TLAPS.

7.2 Formal Methods for Developing and Analyzing Algorithms and Systems

Participants: Étienne André, Johan Arcile, Thomas Bagrel, Martin Bromberger, Horatiu Cirstea, Marie Duflot-Kremer, Engel Escaffre-Lefaucheux, Serguei Lenglet, Pierre Lermusiaux, Benjamin Loillier, Dylan Marinho, Dominique Méry, Stephan Merz, Pierre-Etienne Moreau, Victor Roussanaly, Vincent Trélat, Christoph Weidenbach.

7.2.1 Contributions to Formal Methods of System Design

A complete system of proof rules for auxiliary variables. A key problem in proving refinement between two specifications of a system at different levels of abstraction is to assign suitable values to internal values of the high-level specification, given an execution of the low-level specification. It is well known that the standard technique of refinement mappings is incomplete; in their classic paper *The existence of refinement mappings* [63], Abadi and Lamport introduced the concept of auxiliary variables and proved the technique to be complete under certain conditions. In joint work with Leslie Lamport (Microsoft Research), we introduce simpler rules for prophecy variables that predict a certain event, as well as rules for combining auxiliary variables. We demonstrate how our rules can be used in a linearizability proof, and prove unrestricted (relative) completeness of our technique. Our paper was published in ACM Transactions on Programming Languages and Systems [17].

Integration of knowledge in formal development. System engineering advocates a thorough understanding of the engineering domain or certification standards (aeronautics, railway, medical, etc.) associated with the system under design. In this context, engineering domain knowledge plays a predominant role in system design and/or certification, and this knowledge should be made explicit in the corresponding formal design models. Using external theories, we introduce a formal method based on Event-B for describing and setting up domain-specific behavioral analyses [49]. This method allows us to describe and set up domain-specific behavioral analyses [19]. We obtain a formal verification technique for dynamic properties entailed by engineering domain knowledge where Event-B formal models are annotated and analyzed in a non-intrusive way, i.e. without destructive alteration. This method is based on the formalization of analyses of behavioral properties relying on domain knowledge as an ontology on the one hand and a meta-theory for Event-B on the other hand. The proposed method is illustrated using a critical interactive system.

F3FLUID: A formal framework for developing safety-critical interactive systems in FLUID F3FLUID is developed in the ANR project Formedicis and used for developing safety critical systems. F3FLUID (Formal Framework For FLUID) [19] is a unified formal framework for the development of safety-critical interactive systems. This framework is based on the FLUID (Formal Language of User Interface Design) pivot modeling language defined in the FORMEDICIS project and targeted at expressing high-level system requirements for interactive systems. This modeling language 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. We use the Event-B modeling language 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. We illustrate the operational semantics of the FLUID language, and the transformation strategy of FLUID models into Event-B models, including the tool development. We use the ProB model checker to analyse the temporal properties and to animate the formal specification. In addition, an Interactive Cooperative Objects (ICO) model is derived from the Event-B model for animation, visualization and validation of dynamic behaviors, visual properties and task analysis. 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.

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 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 now propose a strategy 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 references.

Certified 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. We propose a transformation from big-step to small-step semantics and certify its correctness using Coq certificates: for a given input language in big-step, we generate the small-step semantics and a Coq proof script that shows the correspondence between the two semantics [27]. We also develop a certified transformation from big-step to abstract machines [26]. Finally, we generate abstract machines in a generic and complete way for non-deterministic languages such as process calculi [33], for which only ad hoc and partial implementations existed so far.

Static analysis of pattern-free properties. From compilation to many approaches of code analysis and formal verification, program transformations are both ubiquitous and critical to properly functioning programs and information systems. In the context of formal verification, it is often necessary to characterize the shape of the result of these transformations. In a typed context, the underlying type system provides syntactic guarantees on the form of these terms by exhibiting, among others, the constructor symbols that they can contain. On the other hand, when performing (program) transformations we often want to eliminate some symbols and, more generally, to ensure that some patterns are absent from the result of the transformation. We have proposed [51] a formalism, based on the notions of pattern-matching and rewriting, to express such properties. The proposed approach relies on annotations on function symbols to express set of specifications describing the expected behavior of the associated functions. Using a rewrite system to encode the considered transformation, we introduced a static analysis method to verify that the rewrite system is indeed consistent with the respective annotations, in order to conclude that the transformation actually verifies the given specifications.

Formal verification of an algorithm for computing strongly connected components. Computing strongly connected components (SCCs) in a graph is a fundamental algorithmic problem that also underlies algorithms for automated system verification by model checking. State-of-the-art algorithms are based on depth-first search, and they are therefore difficult to parallelize. In his PhD thesis [71], Bloemen introduces a novel variant of Dijkstra's SCC algorithm, together with clever data structures suitable for implementations on multi-core architectures. We contributed to ongoing efforts for the formal verification of this algorithm.

During a research project at École des Mines, we formulated a sequential version of the algorithm as two mutually recursive functions in the proof assistant Isabelle/HOL, and we stated suitable pre- and postconditions that allowed us to prove the partial correctness of the algorithm. We also proved that the algorithm terminates for finite graphs. The formal proofs have been published in the Archive of Formal Proofs.

In joint work with Jaco van de Pol (University of Aarhus), we also worked on the verification of parallel versions of the algorithm. We represented the algorithm in TLA$^+$ at two different abstraction levels, corresponding to different grains of atomicity, and verified a number of invariants using model checking and the TLAPS proof assistant.

7.2.2 Automated Reasoning Techniques for Verification

Complementary strengths of verification tools for TLA$^+$. The specification language TLA$^+$ is supported by three main verification tools: the explicit-state model checker TLC, the symbolic SMT-based model checker Apalache, and the interactive proof assistant TLAPS. In joint work with Igor Konnov (Informal Systems) and Markus Kuppe (Microsoft Research), we explore the complementary strengths and weaknesses of these tools, using two specifications of an algorithm for distributed termination

detection due to Safra, expressed at two levels of abstraction. Whereas TLC is very easy to use and can be used for verifying properties for small instances of specifications, it suffers from state space explosion. Apalache performs bounded model checking and becomes quite slow when exploring prefixes of executions beyond a handful of transitions due to the growth of the size of SMT constraints. However, it is very useful for checking candidates for inductive invariants where only one transition needs to be considered, and can in that case handle instances with up to 100 active nodes. Finally, TLAPS is not limited by the size of the considered instances, but requires the user to write a proof whose correctness is then checked by the system. The properties of interest include both safety and liveness properties, making good use of the recently added functionality in TLAPS for handling fairness and liveness conditions. Our experience allows us to suggest a methodology for applying the different tools to different aspects of system verification. A paper describing our findings was published at ISOLA [46], and our specifications and proofs are available on GitHub.

Mechanization and Application of SUPERLOG. In joint work with the groups of Markus Kroetzsch and Christof Fetzer (Technical University of Dresden), we continued the automated reasoning on the so called *SUPERLOG* (Supervisor Logic) fragment that is meant to provide a basis for formalizing abstract control algorithms found in ECUs (Electronical Control Units). The language comes with support for fully automated verification and also for execution [74]. Technically, SUPERLOG is an extension of the (Horn) first-order Bernays-Schoenfinkel fragment with arithmetic constraints. It extends the well known SMT fragment by universally quantified variables. In addition to the already developed sound and complete calculus for the SUPERLOG language and a previously developed Datalog hammer [62, 72] reducing universally as well as existentially quantified queries to plain Datalog, we now refined our hammer by a soft typing discipline. It still outperforms any available state-of-the art technique and improves our previous results by several orders of magnitude. We successfully applied the hammer not only to the ECU scenario but also to a lane change assistant of a car [34]. The SUPERLOG language can also be executed by a hierarchic superposition based interpreter. We improved the performance of the interpreter by several order of magnitude as well by providing fast redundancy tests [35].

7.2.3 Timed model checking

Theoretical questions. In [15], we considered timed games with cost on both transitions and states. Existing work focused mostly on systems where the costs were restricted to non-negative values, thus failing to represent energy consumption and creation for instance. We showed how to solve these games under some usual restrictions over clocks and resets.

In [12], we considered theoretical problems related to reachability and liveness in (subclasses of) parametric timed automata, exhibiting the frontier between decidability and undecidability.

Heuristics and efficient synthesis. In [29, 31], we proposed new algorithms using convex merging and zone extrapolations to synthesize parameter valuations in parametric timed automata much more efficiently than in the past. Our algorithms were implemented and demonstrated on a library of benchmarks.

Application to real-time systems. We developed a tool [28] based on IMITATOR that detects when one can ensure through a simple form of control that a given timed automata is timed opaque (a security property) and which, in the positive case, builds this control.

Monitoring cyber-physical systems. In [41], we proposed new monitoring algorithms to monitor the (offline or online) behavior of cyber-physical systems used as a black box (no inner information known). We do however add the very rough knowledge of a "bounding model" to eliminate spurious violation detections. Several approaches were proposed with different domains of applications.

In [30], we exemplify a specification by exhibiting concrete examples of sample runs, from a specification given in the form of a quantitative extension of automata, involving real-valued signals.

7.2.4 Model Checking Linear Dynamical Systems

Evolution of simple 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. Modeling loops as simple linear dynamical systems, we studied in [45] which kind of properties formulated in Monadic Second Order Logic could be checked over the evolution of a loop. In [32], we parameterized our model, allowing for instance a more accurate representation of the unknown variables that could affect the behavior of a loop. This work aimed at synthesizing for which values of the parameters a loop satisfies some ω-regular properties.

Specialization to termination. Proving the termination of a loop is one of the main properties that one may want to check. In [37] we considered loops (once again modeled as linear dynamical systems) that were already guaranteed to terminate. Despite this guarantee, one could remain within these loops for a very long time. Our result provides a tight bound on the number of iterations within the loop before which the system escapes. We required this bound to be independent of the initial values of the variables (at least when such a bound exists).

Extension to non-deterministic models. Abstraction of a complex behavior of a system is one of the causes leading to non-deterministic modeling. In [36] we consider the evolution of non-deterministic linear dynamical systems in the form of weighted automata. We establish when and how can we detect whether such systems can diverge, or converge to 0.

7.3 Verification and Analysis of Dynamic Properties of Biological Systems

Participants: Hamid Rahkooy, Thomas Sturm.

7.3.1 Generation and Public Provision of Formal Specifications of Biological Models

In the course of our research line on Biological Systems we are applying automated reasoning to problems such as sustained oscillations and Hopf bifurcations, multi-stationarity, multi-scale model reduction, dynamical invariants, and structural properties of steady state varieties. Compared to numerical analysis and simulation, our approach provides not only quantitative but also qualitative results about network dynamics, to some extent in parametric settings. Our focus has been on reaction networks in the sense of chemical reaction network theory [75]. Such networks are usually stored and exchanged in the Systems Biology Markup Language (SBML), a free, open and standardized XML-based format [78]. However, on the one hand, formal methods do not utilize the full information contained in SBML models. For instance, SBML was designed with a focus on network simulation and supports corresponding concepts like events and initial assignments, which are not natural from our point of view. On the other hand, our automated reasoning is based on symbolic computation, which operates on formal objects like polynomials, exact rational numbers, and ordinary differential equations (ODEs), which are not readily available in SBML. For instance, ODEs describing differential network kinetics can be modeled as pieces of code to be used with numerical solvers instead of mathematical expressions accessible for formal methods.

The rigorous construction of of suitable formal models requires joint competence and combined efforts from computer science, mathematics, and biology. Not only resolving but even recognizing issues is a challenge with every single model considered. Software tool chains must be integrated with human interaction. Emerging from combined efforts within the SYMBIONT project (9.1.1), we have picked up the challenge and launched a web service ODEbase (6.2.1) that provides reliable input data for formal methods to the research community. We identify the following benefits:

1. Interdisciplinary competence: Researchers get access to biologically adequate translations of existing relevant biomodels instead of ad hoc solutions.

2. Availability: ODEbase models used and cited in the literature can be conveniently reviewed on the basis of the original data and re-used in follow-up publications.

3. Canonical reference: ODEbase provides an unambiguous mapping of the, in general, too liberal SBML names for species concentrations and parameters to common mathematical notation. This facilitates comparability of results between publications.

4. Benchmarking: Beyond its primary purpose, ODEbase is perfectly suited to generate benchmark sets for novel algorithms and software in the field.

At the time of writing, ODEbase comprises 660 models, mostly from the renowned biomodels.net online database [82]. For further details on the project see [18].

7.3.2 Approximate Conservation Laws

In Section 3.3 we have seen an example for the reduction of the kinetics of a chemical reaction network to multiple timescales [61]. 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 in progress has led to two preprints so far [54, 55].

8 Bilateral contracts and grants with industry

8.1 Bilateral contracts with industry

Participants: Martin Bromberger, Christoph Weidenbach.

The Max Planck Institute for Informatics (MPI-INF) and Logic 4 Business GmbH (L4B) have signed a cooperation contract. Its subject is the application of automated reasoning methods to product complexity management, in particular in the car industry. MPI-INF is providing software and know-how, L4B is providing real-world challenges. The agreement involves Martin Bromberger and Christoph Weidenbach. The company L4B was successfully sold in 2021 to an industrial partner.

9 Partnerships and cooperations

9.1 International initiatives

9.1.1 Participation in other International Programs

ANR-NRF ProMiS

Participants: Étienne André, Johan Arcile, Dylan Marinho.

Title: Provable Mitigation of Side Channel through Parametric Verification

Duration: 2020–2024

Coordinators: Étienne André, Jun Sun

Partner Institutions:

- University of Lorraine, France (coordinator)
- École Centrale Nantes, France
- Singapore Management University (coordinator)
- Singapore University of Technology and Design

Keywords: security, formal methods, model checking, timed automata

Summary: The Spectre vulnerability illustrates the fact that attackers can extract information about private data using a timing attack. It is an example of side channel attacks, where secure information flows through side channels unintentionally. We propose techniques for automatically synthesizing mitigations of side channel attacks using formal verification techniques, by reducing this problem to the parameter synthesis problem of a given formalism. We plan to deliver a fully automated toolkit which can be automatically applied to real-world systems.

More information: ProMiS Web site

ANR-DFG SYMBIONT

Participants: Hamid Rahkooy, Thomas Sturm.

Title: Symbolic Methods for Biological Networks

Duration: July 2018–April 2022

Coordinators: Thomas Sturm and Andreas Weber/Reinhard Klein

Partner Institutions:

- CNRS / LORIA (coordinator)
- University of Lille 1, France
- University of Montpellier, France
- Inria Saclay Île de France (Lifeware), France
- University of Bonn, Germany (coordinator)
- RWTH Aachen (Department of Mathematics and Joint Research Center for Computational Biomedecine), Germany
- University of Kassel, Germany

Keywords: molecular interaction networks, computational models, symbolic methods, tropical geometry, real algebraic geometry

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

More information: SYMBIONT Web site

PHC Polonium

Participant: Sergueï Lenglet.

Title: Abstract Machines for Programming Languages: Investigations in Formal Interderivations

Duration: 2020–2022

Coordinators: Sergueï Lenglet and Dariusz Biernacki

Partner Institutions:

- University of Lorraine / LORIA, France
- University of Wrocław, Poland

Summary: The project funds travel of researchers working on the derivation of abstract machines from the formal semantics of programming languages.

9.2 International research visitors

9.2.1 Visits of international scientists

Jaco van de Pol

Status: professor

Institution of origin: University of Aarhus

Country: Denmark

Dates: 1–15 October 2022

Context of the visit: Collaboration with Étienne André, Dylan Marinho, Stephan Merz

Mobility program/type of mobility: Invited professor (Université de Lorraine)

9.3 European initiatives

9.3.1 H2020 projects

Matryoshka

Participants: Jasmin Blanchette, Rosalie Defourné, Pascal Fontaine, Stephan Merz, Hans-Jörg Schurr, Sophie Tourret, Uwe Waldmann.

Matryoshka project on cordis.europa.eu

Program: ERC

Title: Fast Interactive Verification through Strong Higher-Order Automation

Duration: March 2017 – February 2022

Partners:

- Free University of Amsterdam, The Netherlands (coordinator)
- Inria
- University of Lorraine, France

Inria contact: Stephan Merz

Coordinator: Jasmin Blanchette

Summary: Proof assistants are increasingly used to verify hardware and software and to formalize mathematics. However, despite some success stories, they remain very laborious to use. The situation has improved with the integration of first-order automatic theorem provers—superposition provers and SMT (satisfiability modulo theories) solvers—but only so much can be done when viewing automatic provers as black boxes. The purpose of Matryoshka is to deliver much higher levels of automation to users of proof assistants by fusing and extending two lines of research: automatic and interactive theorem proving. Our approach is to enrich superposition and SMT with higher-order reasoning in a careful manner, in order to preserve their desirable properties. With higher-order superposition and higher-order SMT in place, we will develop highly automatic provers building on modern superposition provers and SMT solvers, following a novel stratified architecture, and integrate them in proof assistants. Users stand to experience substantial productivity gains: From 2010 to 2016, the success rate of automatic provers on interactive proof obligations from a representative benchmark suite called Judgment Day has risen from 47% to 77%; with this project, we aim at 90%–95% proof automation.

More information: Matryoshka Web site

9.3.2 Other European programs

Erasmus+ ARC.

Program: Erasmus+

Title: Automated reasoning in the class

Duration: October 2019 – August 2022

Coordinator: West University of Timisoara (Romania)

Partners:

- Johannes Kepler University Linz, Austria
- RWTH Aachen, Germany
- Eszterházy Károly Catholic University, Eger, Hungary
- University of Lorraine, France

Inria contact: Sorin Stratulat

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

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), Stephan Merz, Hans-Jörg Schurr, 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 about 300 researchers from 40 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.

9.4 National initiatives

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, 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: Zheng Cheng, 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: Zheng Cheng, Dominique Méry, Guillaume Verdier

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 Formedicis

Title: Formal methods for the development and the engineering of critical interactive systems

Duration: January 2017 – July 2022

Coordinator: David Chemouil

Partner Institutions:

- ONERA, Toulouse (coordinator)
- ENSEEIHT & IRIT, Toulouse
- ENAC, Toulouse
- University of Lorraine

Team participants: Horatiu Cirstea, Dominique Méry

Summary: During the last 30 years, the aerospace domain has successfully devised rigorous methods and tools for the development of safe functionally-correct software. During this process, interactive software has received a relatively lower amount of attention. However, Human-System Interactions (HSI) are important for critical systems and especially in aeronautics: for example, the investigation into the crash of the Rio-Paris flight AF 447 in 2009 pointed out a design issue in the Flight Director interface as one of the original causes of the crash. Formedicis aims at designing a formal hub language, in which designers can express their requirements concerning the interactive behavior that must be embedded inside applications, and at developing a framework for validating, verifying, and implementing critical interactive applications expressed in that language.

Keywords: critical systems, aeronautics, human-system interaction, system requirements

ANR Project ICSPA

Title: Interoperable and Confident Set-based Proof Assistants

Duration: January 2022 – December 2025

Coordinator: Catherine Dubois

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: 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

DFG Transregional Research Center 248 CPEC

Title: Foundations of Perspicuous Software Systems.

Duration: January 2019 – December 2022.

Coordinators: Holger Hermanns and Raimund Dachselt

Partner Institutions:

- Saarland University (coordinator)
- University of Dresden (coordinator)
- Max Planck Institute for Software Systems, Saarbrücken

Team participants: Fajar Haifani, Sophie Tourret, Christoph Weidenbach.

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

Keywords: cyber-physical system, explainability, causal analysis

More information: Perspicuous Computing Web site

9.5 Regional initiatives

The PhD thesis of Rosalie Defourné is partly funded by Région Grand Est.

10 Dissemination

10.1 Promoting scientific activities

10.1.1 Scientific events: organisation

General chair, scientific chair. Dominique Méry was co-chair of the conferences MEMOCODE and ABZ.

Member of organizing committees. Sophie Tourret was workshop co-chair of the conference IJCAR (International Joint Conference on Automated Reasoning).

10.1.2 Scientific events: selection

Chair of conference program committees. Jasmin Blanchette was co-chair of the program committee of the conference IJCAR (International Joint Conference on Automated Reasoning).

Member of conference program committees.

- Étienne André: FASE (25th International Conference on Fundamental Approaches to Software Engineering), FORTE (42nd International Conference on Formal Techniques for Distributed Objects, Components, and Systems), FTSCS (8th International Workshop on Formal Techniques for Safety-Critical Systems), ICFEM (23rd International Conference on Formal Engineering Methods), PRDC (27th IEEE Pacific Rim International Symposium on Dependable Computing), TASE (16th Theoretical Aspects of Software Engineering Conference).

- Jasmin Blanchette: ITP (Interactive Theorem Proving), TACAS (Tools and Algorithms for the Construction and Analysis of Systems).

- Horatiu Cirstea: FSCD (7th International Conference on Formal Structures for Computation and Deduction, RuleML+RR (6th International Joint Conference on Rules and Reasoning), TASE (International Symposium on Theoretical Aspects of Software Engineering).

- Pascal Fontaine: IJCAI (31st International Joint Conference on Artificial Intelligence), IJCAR (International Joint Conference on Automated Reasoning).

- Sergueï Lenglet: ICE (15th Interaction and Concurrency Experience).

- Dominique Méry: F-IDE (7th Workshop on Formal Integrated Development Environment), SETTA (Symposium on Dependable Software Engineering: Theories, Tools and Applications), MEDI (11th International Conference on Model and Data Engineering), FMAS (Fourth Workshop on Formal Methods for Autonomous Systems), ICFEM (23rd International Conference on Formal Engineering Methods), ICI2ST (3rd International Conference on Information Systems and Software Technologies), TASE (International Symposium on Theoretical Aspects of Software Engineering).

- Stephan Merz: ICFEM (23rd International Conference on Formal Engineering Methods), iFM (17th International Conference on integrated Formal Methods), ARNQL (4th International Workshop on Automated Reasoning in Quantified Non-Classical Logics), F-IDE (7th Workshop on Formal Integrated Development Environments).

- Sorin Stratulat: SYNASC (24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing), ICEUTE (13th International Conference on EUropean Transnational Educational), CISIS (15th International Conference on Computational Intelligence in Security for Information Systems).

- Thomas Sturm: CASC (24th International Workshop on Computer Algebra in Scientific Computing).

- Sophie Tourret: IJCAI (31st International Joint Conference on Artificial Intelligence) and IJCAR (11th International Joint Conference on Automated Reasoning).

- Uwe Waldmann: IJCAR (11th International Joint Conference on Automated Reasoning).

- Christoph Weidenbach: IJCAR 2022 (11th International Joint Conference on Automated Reasoning).

10.1.3 Journal

Member of editorial boards.

- Jasmin Blanchette is editor-in-chief of the *Journal of Automated Reasoning*.
- 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* [50].

10.1.4 Invited talks

- Stephan Merz gave an invited tutorial on the TLA$^+$ language and its support tools at the University of Lancaster and an invited presentation on auxiliary variables at the CISPA-LORIA workshop in Nancy.
- Thomas Sturm gave an invited talk on *Real Quantifier Elimination by Virtual Substitution* at a workshop on Trends in Arithmetic Theories at ICALP 2022 (49th EATCS International Colloquium on Automata, Languages, and Programming).
- Christoph Weidenbach gave an invited talk on *Algorithm Design for Hard Problems* at the university of New Mexico.

10.1.5 Leadership within the scientific community

- Pascal Fontaine is an elected CADE Trustee. In the COST action EuroProofNet, he was workgroup leader until October 2022, he is now workgroup vice-leader, and he is in the management committee as one Belgian representative. He was member of the committees for the William McCune PhD Award 2022, and for the Herbrand Award 2022.
- 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*.
- Sophie Tourret has become an elected CADE Trustee in 2022.
- Christoph Weidenbach was an elected CADE Trustee and president of CADE until September 2022.

10.1.6 Scientific expertise

- Stephan Merz reviewed a Wittgenstein project for the Austrian Science Fund (FWF) and two ERC project proposals.
- Thomas Sturm is a project partner in the Engineering and Physical Sciences Research Council (EPSRC) Projects EP/T015748/1 and EP/T015713/1 *Pushing Back the Doubly-Exponential Wall of Cylindrical Algebraic Decomposition*, Universities of Coventry and Bath, UK.

10.1.7 Research administration

- Étienne André has been involved in three initiatives around open research data:
  1. He has been a member since 2021 of the "EOSC (European Open Science Cloud) Future User Group", that aims to prepare the opening of the EOSC platform, aiming (among many other goals) at opening the European scientists research data.
  2. He has been a member since 2022 of the national board of the disciplinary DoRANum, aiming at helping France-based scientists in opening their research data. This participation notably leads him to prepare pedagogical material specifically dedicated to researchers in computer science, to help them to open their research data.
  3. He has been a member of the network of data ambassadors at University of Lorraine since its creation in 2020. The goal of this network is to help researchers within University of Lorraine to open their data when possible, and keep them closed when necessary.
- Étienne André and Marie Duflot-Kremer were members of a hiring committee for an assistant professor position at IBISC, Évry.
- Stephan Merz is a member of the executive committee of the project on citizens' trust in the digital world (DigiTrust) funded by *Lorraine Université d'Excellence*.
- Sophie Tourret was a jury member for the CRCN researchers recruitment campaign 2022 at Inria Nancy - Grand Est.

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- BUT 1: Étienne André, Data structures, 42 HETD, Université de Lorraine – IUT Charlemagne, France.
- BUT 1: Étienne André, Human-machine interfaces, 57 HETD, Université de Lorraine – IUT Charlemagne, France.
- BUT 1: Étienne André, Architecture des réseaux, 32 HETD, Université de Lorraine – IUT Charlemagne, France.
- BUT 1: Étienne André, Object-oriented software design, 38 HETD, Université de Lorraine – IUT Charlemagne, France.
- BUT 2: Étienne André, Supervised projects, 56 HETD, Université de Lorraine – IUT Charlemagne, France.
- 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, 60 HETD, L1, Université de Lorraine, France.
- Diplôme inter universitaire: Marie Duflot-Kremer, formation d'enseignants du secondaire à la spécialité NSI, 18 HETD, Université de Lorraine, France.

- Licence: Marie Duflot-Kremer, Individual support for algorithms and programming, 60 HETD, L1, Université de Lorraine, France.
- Master: Marie Duflot-Kremer, Using unplugged activities for future teachers.
- 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.
- Licence: Engel Escaffre-Lefaucheux, Objected-oriented programming, 14 HETD, L2, Université de Lorraine.
- Classe préparatoire universitaire: Engel Escaffre-Lefaucheux, Algorithms and programming (2 and 3), 15 HETD, Université de Lorraine.
- Licence: Engel Escaffre-Lefaucheux, Algorithms and programming 2, 20 HETD, L2, Université de Lorraine.
- Master: Engel Escaffre-Lefaucheux, supervision of 3 students in an intial research experience, M1, Université de Lorraine.
- BUT 1: Sergueï Lenglet, Introduction to data bases, 110 HETD, Université de Lorraine – IUT Charlemagne, France.
- BUT 1: Sergueï Lenglet, Use of data bases, 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: Sophie Tourret, Decision Procedures for Program Verification, 32 HETD, M2 Informatique and Master Erasmus Mundus DESEM (academic year 2021-2022), Université de Lorraine, France.
- Master: Sophie Tourret, Decision Procedures for Program Verification, M2 Informatique (academic year 2022-2023), Université de Lorraine, France.
- Master: Sophie Tourret and Stephan Merz, Secure Coding, M1 Mines Nancy, 26 HETD, Université de Lorraine, France
- Master: Uwe Waldmann, Automated Reasoning I, 60 HETD, Universität des Saarlandes, Germany.
- Master: Uwe Waldmann and Christoph Weidenbach, Automated Reasoning II, 40 HETD, Universität des Saarlandes, Germany.
- Licence: Markus Bläser, Karl Bringmann, Martin Bromberger, and Christoph Weidenbach, Competitive Programming, 40 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.

10.2.2 Supervision

- PhD: Guillaume Ambal, *Skeletal Semantics Transformations*, Université de Rennes 1. Supervised by Alan Schmitt and Sergueï Lenglet, 18 October 2022.
- PhD: Pierre Lermusiaux, *Static analysis of pattern eliminating transformations*, Université de Lorraine. Supervised by Horatiu Cirstea and Pierre-Étienne Moreau, 8 September 2022.
- PhD: Christoph Lüders, *Algorithmic Reduction of Biochemical Reaction Networks*, University of Kassel, Germany. Supervised by Werner Seiler, Thomas Sturm, Sebastian Walcher, Andreas Weber, 2 June 2022.
- PhD: Hans-Jörg Schurr, *Stronger SMT Solvers for Proof Assistants: Proofs, Quantifier Simplification, Strategy Schedules*, Université de Lorraine. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, 7 October 2022.
- 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: Rosalie Defourné, SMT for TLAPS, Université de Lorraine. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, since March 2019.
- PhD: 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: 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, since November 2019.
- 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: Dylan Marinho, *Detecting timing attacks using formal methods*, Université de Lorraine. Supervised by Étienne André, since October 2020.
- PhD in progress: Simon Schwarz, *Automatic Reasoning for Security*, Universität des Saarlandes. Supervised by Christoph Weidenbach, since October 2022.
- L3: Florent Krasnopol, *Verifying in Isabelle/HOL a notion of consequence relation*, ENS Paris Saclay, summer internship. Supervised by Sophie Tourret, June-July 2022.
- M2: Shapagat Bolat, *Enforcing time-opacity of timed automata*, Master internship for Université de Lorraine, March-June 2022, supervised by Étienne André and Engel Lefaucheux

- M2: Nurgul Osmonova, *Optimising Diagnosability in probabilistic systems*, Master internship for Université de Lorraine, March-June 2022, supervised by Engel Lefaucheux.
- M1: Vincent Trélat, *A mechanized proof of an algorithm for computing strongly connected components in a graph*, research project at École des Mines de Nancy, September 2021-July 2022, supervised by Stephan Merz.
- M2: Zunaira Zaman, *Distributed PlusCal*, Master internship for Université de Lorraine, March-June 2022, supervised by Horatiu Cirstea and Stephan Merz.

10.2.3 Juries

- Étienne André reviewed the PhD thesis of Bastien Serée (Nantes) and was a PhD jury member of the defense of Mathieu Hilaire (Saclay).
- Stephan Merz was a reviewer of the PhD theses of Benjamin Binder (Saclay) and of Pierre Civit (Paris). He was the president of the PhD committees of Aurèle Barrière (Rennes), Abir Laraba (Nancy), and Matthieu Nicolas (Nancy).

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 jury of the Cyber Agora 41 award organized by ANSSI (the French agency for the security of information systems) for rewarding a novel related to computer science or cyber-security.
- 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.
- 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 Berlin in September 2022. Stephan Merz was a member of that jury.

10.3.2 Articles and contents

- Within a SIF working group, Marie Duflot-Kremer supervised the production of four videos on the four basic blocks of computer science: machines, languages, data, and algorithms.

10.3.3 Interventions

- Marie Duflot-Kremer organized a workshop on databases and gave a short talk at the Journée NSI-SNT in April in Nancy.
- Marie Duflot-Kremer was an invited speaker at the 6th Journée académique sur l'enseignement de l'informatique in Marseille in May 2022.
- Marie Duflot-Kremer gave a presentation "Rencontre avec une informagicienne" in Strasbourg in June 2022.
- Marie Duflot-Kremer organized a workshop on unplugged computer science activities in Strasbourg in October 2022.

- Marie Duflot-Kremer and Sophie Tourret organized a stand at the "Fête de la science" event (October 2022), to present unplugged computer science activities to the general public with the help of bachelor and master students.
- Marie Duflot-Kremer and Sophie Tourret organized an outreach activity around the game "Turing tumble" (a mechanical, marble-powered programmable computer) at the Ludothèque Saint-Nicolas in Nancy in November 2022.
- Marie Duflot-Kremer and Sophie Tourret took part in several meetings with primary and 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, D. Lime, D. Marinho and J. Sun. 'Guaranteeing Timed Opacity using Parametric Timed Model Checking'. In: *ACM Transactions on Software Engineering and Methodology* 31.4 (31st Oct. 2022), pp. 1–36. DOI: 10.1145/3502851. URL: https://hal.science/hal-03798157.

[12] É. André, D. Lime and O. H. Roux. 'Reachability and liveness in parametric timed automata'. In: *Logical Methods in Computer Science* 18.1 (9th Feb. 2022). DOI: 10.46298/lmcs-18(1:31)2022. URL: https://hal.archives-ouvertes.fr/hal-03574379.

[13] J. Arcile and É. André. 'Timed automata as a formalism for expressing security: A survey on theory and practice'. In: *ACM Computing Surveys* (7th June 2022). DOI: 10.1145/3534967. URL: https://hal.archives-ouvertes.fr/hal-03690234.

[14] L. Bellatreche, C. Ordonez, D. Méry, M. Golfarelli and E. H. Abdelwahed. 'The central role of data repositories and data models in Data Science and Advanced Analytics'. In: *Future Generation Computer Systems* 129 (Apr. 2022), pp. 13–17. DOI: 10.1016/j.future.2021.11.027. URL: https://hal.inria.fr/hal-03904787.

[15] T. Brihaye, G. Geeraerts, A. Haddad, E. Lefaucheux and B. Monmege. 'One-Clock Priced Timed Games with Arbitrary Weights'. In: *Logical Methods in Computer Science* 18.3 (9th Aug. 2022), p. 51. URL: https://hal.archives-ouvertes.fr/hal-02424743.

[16] I. Drămnesc, E. Ábrahám, T. Jebelean, G. Kusper and S. Stratulat. 'Automated Reasoning in the Class'. In: *Computer-Algebra-Rundbrief* 71 (Oct. 2022), pp. 21–26. URL: https://hal.inria.fr/hal-03886685.

[17] L. Lamport and S. Merz. 'Prophecy Made Simple'. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 44.2 (30th June 2022), pp. 1–27. DOI: 10.1145/3492545. URL: https://hal.inria.fr/hal-03636686.

[18] C. Lüders, T. Sturm and O. Radulescu. 'ODEbase: A Repository of ODE Systems for Systems Biology'. In: *Bioinformatics Advances* 2.1 (26th Apr. 2022). DOI: 10.1093/bioadv/vbac027. URL: https://hal.archives-ouvertes.fr/hal-03651751.

[19] 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://hal.inria.fr/hal-03904803.

[20] D. Méry and S. Qin. 'Selected papers from The 13th International Symposium on Theoretical Aspects of Software Engineering 29 July – 1 August 2019, Guilin, China'. In: *Science of Computer Programming* 218 (June 2022), p. 102804. DOI: 10.1016/j.scico.2022.102804. URL: https://hal.inria.fr/hal-03904791.

[21] D. Méry and A. Raschke. 'Selected papers from the Rigorous State-Based Methods 7th International Conference, ABZ 2020, Ulm, Germany, May 27–29, 2020'. In: *Science of Computer Programming* 216 (Apr. 2022), p. 102780. DOI: 10.1016/j.scico.2022.102780. URL: https://hal.inria.fr/hal-03904790.

[22] Y. Sheng, Y. Zohar, C. Ringeissen, J. Lange, P. Fontaine and C. Barrett. 'Polite Combination of Algebraic Datatypes'. In: *Journal of Automated Reasoning* 66.3 (Aug. 2022), pp. 331–355. DOI: 10.1007/s10817-022-09625-3. URL: https://hal.inria.fr/hal-03853159.

[23] P. Vukmirović, A. Bentkamp, J. Blanchette, S. Cruanes, V. Nummelin and S. Tourret. 'Making Higher-Order Superposition Work'. In: *Journal of Automated Reasoning* 66.4 (Nov. 2022), pp. 541–564. DOI: 10.1007/s10817-021-09613-z. URL: https://hal.inria.fr/hal-03909997.

[24] P. Vukmirović, J. Blanchette, S. Cruanes and S. Schulz. 'Extending a brainiac prover to lambda-free higher-order logic'. In: *International Journal on Software Tools for Technology Transfer* 24.1 (Feb. 2022), pp. 67–87. DOI: 10.1007/s10009-021-00639-7. URL: https://hal.inria.fr/hal-03814641.

[25] U. Waldmann, S. Tourret, S. Robillard and J. Blanchette. 'A Comprehensive Framework for Saturation Theorem Proving'. In: *Journal of Automated Reasoning* 66.4 (Nov. 2022), pp. 499–539. DOI: 10.1007/s10817-022-09621-7. URL: https://hal.inria.fr/hal-03909983.

International peer-reviewed conferences

[26] G. Ambal, S. Lenglet and A. Schmitt. 'Certified Abstract Machines for Skeletal Semantics'. In: CPP 2022 - 11th ACM SIGPLAN International Conference on Certified Programs and Proofs. Philadelphia, United States, 17th Jan. 2022, pp. 1–13. DOI: 10.1145/3497775.3503676. URL: https://hal.inria.fr/hal-03466807.

[27] G. Ambal, S. Lenglet, A. Schmitt and C. Noûs. 'Certified Derivation of Small-Step From Big-Step Skeletal Semantics'. In: PPDP 2022 - 24th International Symposium on Principles and Practice of Declarative Programming. Tbilisi, Georgia, 20th Sept. 2022, pp. 1–48. DOI: 10.1145/3551357.3551384. URL: https://hal.inria.fr/hal-03768820.

[28] É. André, S. Bolat, E. Lefaucheux and D. Marinho. 'strategFTO: Untimed control for timed opacity'. In: *Proceedings of the 8th International Workshop on Formal Techniques for Safety-Critical Systems (FTSCS 2022)*. 8th International Workshop on Formal Techniques for Safety-Critical Systems (FTSCS 2022). Proceedings of the 8th International Workshop on Formal Techniques for Safety-Critical Systems (FTSCS 2022). Auckland, New Zealand: ACM, 7th Dec. 2022, pp. 27–33. DOI: 10.1145/3563822.3568013. URL: https://hal.archives-ouvertes.fr/hal-03874435.

[29] É. André, D. Marinho, L. Petrucci and J. van de Pol. 'Efficient Convex Zone Merging in Parametric Timed Automata'. In: 20th International Conference on Formal Modeling and Analysis of Timed Systems. Vol. 13465. Lecture Notes in Computer Science. Warsaw, Poland: Springer International Publishing, 29th Aug. 2022, pp. 200–218. DOI: 10.1007/978-3-031-15839-1_12. URL: https://hal.archives-ouvertes.fr/hal-03772708.

[30] É. André, M. Waga, N. Urabe and I. Hasuo. 'Exemplifying Parametric Timed Specifications over Signals with Bounded Behavior'. In: *Proceedings of the 14th NASA Formal Methods Symposium (NFM 2022)*. 14th NASA Formal Methods Symposium (NFM 2022). Vol. 13260. Lecture Notes in Computer Science. Pasadena, United States: Springer International Publishing, 20th May 2022, pp. 470–488. DOI: 10.1007/978-3-031-06773-0_25. URL: https://hal.science/hal-03690071.

[31] J. Arcile and É. André. 'Zone Extrapolations in Parametric Timed Automata'. In: *Proceedings of the 14th NASA Formal Methods Symposium (NFM 2022)*. 14th NASA Formal Methods Symposium (NFM 2022). Vol. 13260. Lecture Notes in Computer Science. Caltech, Pasadena, United States: Springer International Publishing, 20th May 2022, pp. 451–469. DOI: 10.1007/978-3-031-06773-0_24. URL: https://hal.archives-ouvertes.fr/hal-03690070.

[32] C. Baier, F. Funke, S. Jantsch, T. Karimov, E. Lefaucheux, J. Ouaknine, D. Purser, M. A. Whiteland and J. Worrell. 'Parameter Synthesis for Parametric Probabilistic Dynamical Systems and Prefix-Independent Specifications'. In: 33rd International Conference on Concurrency Theory (CONCUR 2022). Varsovie, Poland, 12th Sept. 2022. DOI: 10.4230/LIPIcs.CONCUR.2022.10. URL: https://hal.inria.fr/hal-03789856.

[33] M. Biernacka, D. Biernacki, S. Lenglet and A. Schmitt. 'Non-Deterministic Abstract Machines'. In: CONCUR 2022 - 33rd International Conference on Concurrency Theory. Varsovie, Poland, 13th Sept. 2022, pp. 1–24. DOI: 10.4230/LIPIcs.CONCUR.2022.7. URL: https://hal.inria.fr/hal-03772712.

[34] M. Bromberger, I. Dragoste, R. Faqeh, C. Fetzer, L. González, M. Krötzsch, M. Marx, H. Murali and C. Weidenbach. 'A Sorted Datalog Hammer for Supervisor Verification Conditions Modulo Simple Linear Arithmetic'. In: Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022. Vol. 13243. Lecture Notes in Computer Science. Munich (Germany), Germany: Springer International Publishing, 30th Mar. 2022, pp. 480–501. DOI: 10.1007/978-3-030-99524-9_27. URL: https://hal.inria.fr/hal-03881921.

[35] M. Bromberger, L. Leutgeb and C. Weidenbach. 'An Efficient Subsumption Test Pipeline for BS(LRA) Clauses'. In: IJCAR 2022 - International Joint Conference in Automated Reasoning. Vol. 13385. Lecture Notes in Computer Science. Haifa, Israel: Springer International Publishing, 1st Aug. 2022, pp. 147–168. DOI: 10.1007/978-3-031-10769-6_10. URL: https://hal.inria.fr/hal-03881893.

[36] W. Czerwiński, E. Lefaucheux, F. Mazowiecki, D. Purser and M. A. Whiteland. 'The boundedness and zero isolation problems for weighted automata over nonnegative rationals'. In: LICS 2022 - 37th Annual ACM/IEEE Symposium on Logic in Computer Science. Haifa, Israel, 2nd Aug. 2022. DOI: 10.1145/3531130. URL: https://hal.inria.fr/hal-03708876.

[37] J. d'Costa, E. Lefaucheux, E. Neumann, J. Ouaknine and J. Worrell. 'Bounding the Escape Time of a Linear Dynamical System over a Compact Semialgebraic Set'. In: 47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022). Vol. 241. 47th International Symposium on Mathematical Foundations of Computer Science. Vienna, Austria, 2022, p. 14. DOI: 10.4230/LIPIcs. URL: https://hal.inria.fr/hal-03712991.

[38] M. Desharnais, P. Vukmirović, J. Blanchette and M. Wenzel. 'Seventeen Provers under the Hammer'. In: 13th International Conference on Interactive Theorem Proving - ITP 2022. Tel Aviv, Israel, 31st July 2022. DOI: 10.5281/zenodo.5940084. URL: https://hal.inria.fr/hal-03814635.

[39] I. Drămnesc, E. Ábrahám, T. Jebelean, G. Kusper and S. Stratulat. 'Experiments with Automated Reasoning in the Class'. In: CICM 2022 - 15th Conference on Intelligent Computer Mathematics. Vol. 13467. Lecture Notes in Computer Science. Tbilisi / Hybrid, Georgia: Springer International Publishing, 17th Sept. 2022, pp. 287–304. DOI: 10.1007/978-3-031-16681-5_20. URL: https://hal.inria.fr/hal-03781994.

[40] I. Drămnesc, T. Jebelean, E. Ábrahám, G. Kusper and S. Stratulat. 'ARC: An Educational Project on Automated Reasoning in the Class'. In: EdMedia + Innovate Learning 2022 - AACE Conferences. New York, United States, 22nd June 2022. URL: https://hal.inria.fr/hal-03900003.

[41] B. Ghosh and É. André. 'Offline and Online Monitoring of Scattered Uncertain Logs Using Uncertain Linear Dynamical Systems'. In: 42nd International Conference on Formal Techniques for Distributed Objects, Components, and Systems. Vol. 13273. Lecture Notes in Computer Science. Lucca, Italy: Springer International Publishing, 12th June 2022, pp. 67–87. DOI: 10.1007/978-3-031-08679-3_5. URL: https://hal.archives-ouvertes.fr/hal-03801020.

[42] F. Haifani, P. Koopmann, S. Tourret and C. Weidenbach. 'Connection-Minimal Abduction in EL via Translation to FOL'. In: *LNCS, LNAI, IJCAR*. Automated Reasoning. Vol. 13385. Lecture Notes in Computer Science. Haifa, Israel: Springer International Publishing; Springer International Publishing, 1st Aug. 2022, pp. 188–207. DOI: 10.1007/978-3-031-10769-6_12. URL: https://hal.inria.fr/hal-03826613.

[43] F. Haifani, P. Koopmann, S. Tourret and C. Weidenbach. 'Connection-Minimal Abduction in EL via Translation to FOL: Extended Abstract'. In: *CEUR Workshops proceedings*. 35th International Workshop on Description Logics (DL 2022). Haifa, Israel, 7th Aug. 2022. URL: https://hal.inria.fr/hal-03937189.

[44] F. Haifani and C. Weidenbach. 'Semantic Relevance'. In: IJCAR, International Joint Conference in Automated Reasoning. Vol. 13385. Lecture Notes in Computer Science. Haifa, Israel: Springer International Publishing, 1st Aug. 2022, pp. 208–227. DOI: 10.1007/978-3-031-10769-6_13. URL: https://hal.inria.fr/hal-03881904.

[45] T. Karimov, E. Lefaucheux, J. Ouaknine, D. Purser, A. Varonka, M. A. Whiteland and J. Worrell. 'What's decidable about linear loops?' In: *Proceedings of the ACM on Programming Languages*. 49th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2022). Vol. 6. POPL 2022. Philadelphia, United States, 16th Jan. 2022, pp. 1–25. DOI: 10.1145/3498727. URL: https://hal.inria.fr/hal-03789796.

[46] I. Konnov, M. Kuppe and S. Merz. 'Specification and Verification with the TLA+ Trifecta: TLC, Apalache, and TLAPS'. In: Leveraging Applications of Formal Methods, Verification and Validation. 11th International Symposium, ISoLA 2022. Vol. 13701. Lecture Notes in Computer Science. Rhodes, Greece: Springer International Publishing, 17th Oct. 2022, pp. 88–105. DOI: 10.1007/978-3-031-19849-6_6. URL: https://hal.inria.fr/hal-03844516.

[47] H. Leidinger and C. Weidenbach. 'SCL(EQ): SCL for First-Order Logic with Equality'. In: IJCAR, International Joint Conference in Automated Reasoning. Vol. 13385. Lecture Notes in Computer Science. Haifa, Israel: Springer International Publishing, 1st Aug. 2022, pp. 228–247. DOI: 10.1007/978-3-031-10769-6_14. URL: https://hal.inria.fr/hal-03881912.

Conferences without proceedings

[48] A. Demin, H. Rahkooy and T. Sturm. 'F5: A REDUCE Package for Signature-based Gröbner Basis Computation'. In: CASC 2022 - Computer Algebra in Scientific Computing. Gezbe, Turkey, 2022. URL: https://hal.archives-ouvertes.fr/hal-03781962.

Scientific book chapters

[49] Y. Aït-Ameur, G. Dupont, I. Mendil, D. Méry, M. Pantel, P. Rivière and N. Singh. 'Empowering the Event-B Method Using External Theories'. In: *Integrated Formal Methods*. Vol. 13274. Lecture Notes in Computer Science. Springer International Publishing, 1st June 2022, pp. 18–35. DOI: 10.1007/978-3-031-07727-2_2. URL: https://hal.inria.fr/hal-03904799.

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

[50] *Special Issue on Computer Algebra in Scientific Computing (CASC 2021)* 16 (Sept. 2022). URL: https://hal.archives-ouvertes.fr/hal-03833048.

Doctoral dissertations and habilitation theses

[51] P. Lermusiaux. 'Static analysis of pattern eliminating transformations'. Université de Lorraine (UL), 8th Sept. 2022. URL: https://hal.inria.fr/tel-03936006.

[52] H.-J. Schurr. 'Stronger SMT Solvers for Proof Assistants : Proofs, Quantifier Simplification, Strategy Schedules'. Université de Lorraine, 7th Oct. 2022. URL: https://hal.univ-lorraine.fr/tel-03845527.

Reports & preprints

[53] M. Biernacka, D. Biernacki, S. Lenglet and A. Schmitt. *Non-Deterministic Abstract Machines*. RR-9475. Inria, July 2022, pp. 1–33. URL: https://hal.inria.fr/hal-03545768.

[54] A. Desoeuvres, A. Iosif, C. Lüders, O. Radulescu, H. Rahkooy, M. Seiß and T. Sturm. *A Computational Approach to Complete Exact and Approximate Conservation Laws of Chemical Reaction Networks*. 30th Dec. 2022. URL: https://hal.science/hal-03934350.

[55] A. Desoeuvres, A. Iosif, C. Lüders, O. Radulescu, H. Rahkooy, M. Seiß and T. Sturm. *Reduction of Chemical Reaction Networks with Approximate Conservation Laws*. 27th Dec. 2022. URL: https://hal.science/hal-03934337.

[56] E. Lefaucheux, J. Ouaknine, D. Purser and M. Sharifi. *Model Checking Linear Dynamical Systems under Floating-point Rounding*. 8th Nov. 2022. URL: https://hal.inria.fr/hal-03843471.

[57] M. Romero, T. Viéville and M. Duflot-Kremer. *Activity for learning computational thinking in plugged and unplugged mode*. 006. UCA - INSPE Académie de Nice, 1st Oct. 2022. URL: https://hal.archives-ouvertes.fr/hal-03793719.

Other scientific publications

[58] M. England, F. Boulier, T. Sadykov and T. Sturm. 'Foreword'. In: *Mathematics in Computer Science* 16.2-3 (Sept. 2022), p. 16. DOI: 10.1007/s11786-022-00533-8. URL: https://hal.archives-ouvertes.fr/hal-03832996.

11.3 Other

Scientific popularization

[59] S. Stratulat. 'Récurrence noethérienne pour le raisonnement de premier ordre'. In: *1024 : Bulletin de la Société Informatique de France* 19 (Apr. 2022), pp. 157–169. DOI: 10.48556/SIF.1024.19.157. URL: https://hal.inria.fr/hal-03688845.

[60] S. Tourret and C. Weidenbach. 'A Posthumous Contribution by Larry Wos: Excerpts from an Unpublished Column'. In: *Journal of Automated Reasoning* 66.4 (Nov. 2022), pp. 575–584. DOI: 10.1007/s10817-022-09617-3. URL: https://hal.inria.fr/hal-03935941.

11.4 Cited publications

[61] 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.

[62] M. Bromberger, A. Fiori and C. Weidenbach. *SCL with Theory Constraints*. 23rd Oct. 2020. URL: https://hal.inria.fr/hal-02975868.

[63] M. Abadi and L. Lamport. 'The Existence of Refinement Mappings'. In: *Theoretical Computer Science* 81.2 (May 1991), pp. 253–284.

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

[65] 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.

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

[67] 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.

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

[69] 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.

[70] A. Bentkamp. 'Superposition for Higher-Order Logic'. PhD thesis. Vrije Universiteit Amsterdam, 2021.

[71] V. Bloemen. 'Strong Connectivity and Shortest Paths for Checking Models'. PhD thesis. Enschede, The Netherlands: University of Twente, 2019.

[72] 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.

[73] 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.

[74] R. Faqeh, C. Fetzer, H. Herrmanns, J. Hoffmann, M. Klauck, M. Koehl, M. Steinmetz and C. Weidenbach. 'Towards Dynamic Dependable Systems through Evidence-Based Continuous Certification'. In: *9th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISOLA 2021)*. 2021.

[75] M. Feinberg. *Foundations of Chemical Reaction Network Theory*. Vol. 202. Applied Mathematical Sciences. Springer, 2019.

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

[77] 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.

[78] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. L. Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner and J. Wang. 'The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models'. In: *Bioinformatics* 19.4 (2003), pp. 524–531. DOI: 10.1093/bioinformatics/btg015.

[79] P. Koopmann, W. Del-Pinto, S. Tourret and R. A. Schmidt. 'Signature-Based Abduction for Expressive Description Logics'. In: *KR*. 2020, pp. 592–602.

[80] L. Wos, G.A. Robinson and D.F. Carson. 'Efficiency and completeness of the set of support strategy in theorem proving'. In: *Journal of the ACM* 12.4 (1965), pp. 536–541.

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

[82] 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.

[83] 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.

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

[85] N. Peltier. 'A Variant of the Superposition Calculus'. In: *Arch. Formal Proofs* 2016 (2016).

[86] S. Schulz, S. Cruanes and P. Vukmirović. 'Faster, Higher, Stronger: E 2.3'. In: *CADE*. Ed. by P. Fontaine. Vol. 11716. LNCS. Springer, 2019, pp. 495–507.

[87] H.-J. Schurr, M. Fleury, H. Barbosa and P. Fontaine. 'Alethe: Towards a Generic SMT Proof Format (extended abstract)'. In: *Seventh Workshop on Proof eXchange for Theorem Proving (PxTP 2021)*. Vol. 336. EPTCS. 2021, pp. 49–54.

[88] H.-J. Schurr, M. Fleury and M. Desharnais. 'Reliable Reconstruction of Fine-Grained Proofs in a Proof Assistant'. In: *28th Intl. Conf. Automated Deduction (CADE 28)*. 2021.