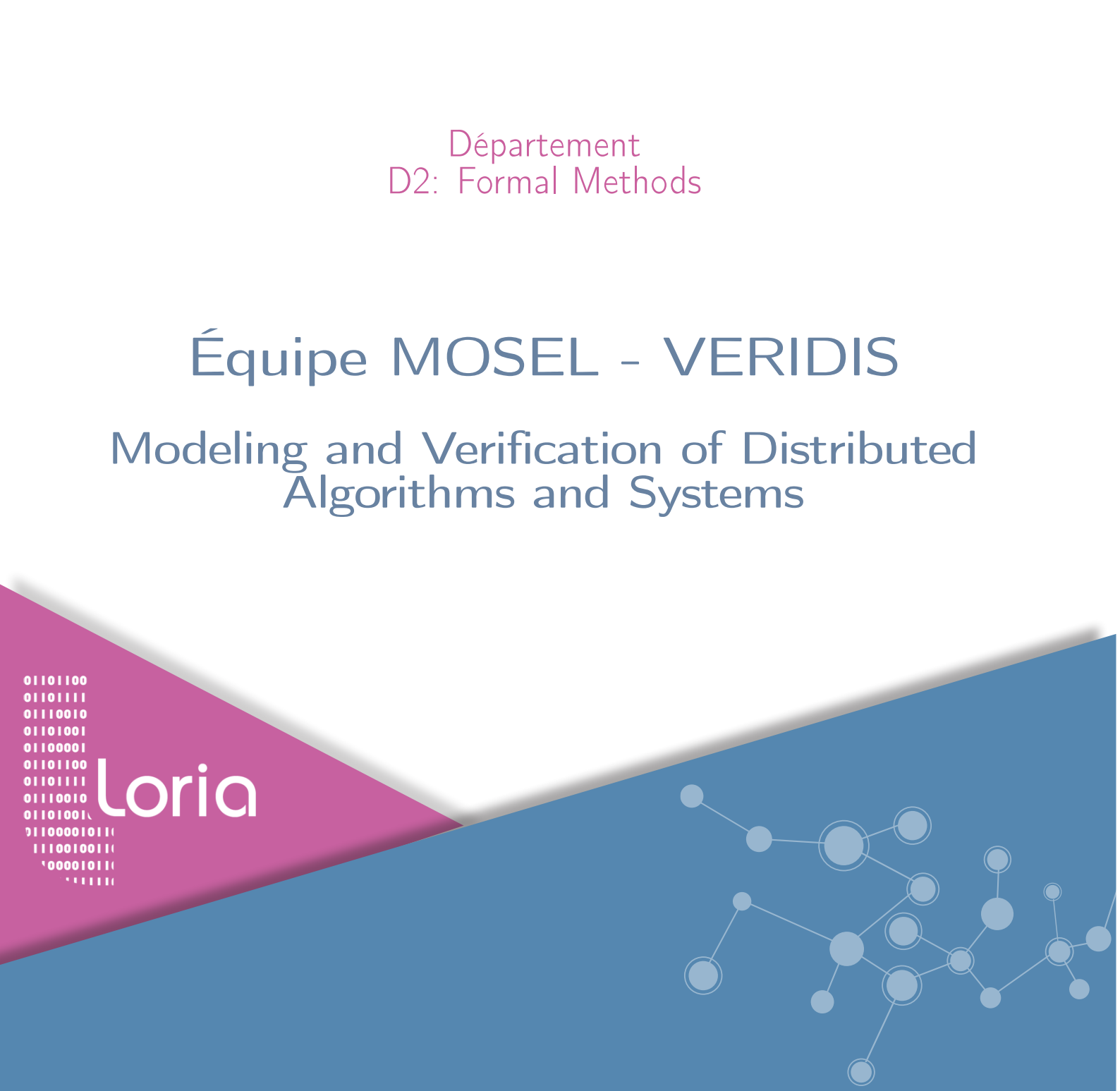
Département
D2: Formal Methods

Équipe MOSEL - VERIDIS

Modeling and Verification of Distributed Algorithms and Systems



Laboratoire lorrain de recherche en informatique et ses applications

Rapport d'activité 2025

cnrs UNIVERSITÉ DE LORRAINE

En partenariat avec

Inria CentraleSupélec

Team VERIDIS

Creation of the Team: 2025 January 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

A2.5. – Software engineering

A4.5. – Formal method for verification, reliability, certification

A4.5.1. – Static analysis

A4.5.2. – Model-checking

A4.5.3. – Program proof

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

Contents

1 Team members, visitors, external collaborators

Research Scientists

- Stephan Merz [Team leader, INRIA, Senior Researcher, HDR]
- Engel Lefaucheux [INRIA, ISFP]
- Thomas Sturm [CNRS, Senior Researcher, HDR]
- Sophie Tourret [INRIA, Researcher]

Faculty Members

- Julie Cailler [UL, Associate Professor]
- Horatiu Cirstea [UL, Professor, HDR]
- Marie Duflot-Kremer [UL, Associate Professor]
- Pierre-Etienne Moreau [UL, Professor, HDR]
- Dominique Méry [Team leader, UL, Professor, HDR]
- Sorin Stratulat [UL, Associate Professor, HDR]
- Martin Vassor [UL, Associate Professor]

Post-Doctoral Fellow

- Qi Qiu [UL, from Oct 2025, ATER]

PhD Students

- Thomas Bagrel [UL, CIFRE, until Nov 2025, Tweag]
- Ghilain Bergeron [INRIA]
- Alessio Coltellacci [INRIA]
- Sarah Depernet [INRIA]
- Florent Krasnopol [UL, from Sep 2025]
- Mohamed Amine Snoussi [UL, CIFRE, until Aug 2025, Westinghouse]
- Vincent Trélat [INRIA]

Interns and Apprentices

- Volkan Burakcin [INRIA, Intern, from Jun 2025 until Jul 2025]
- Tiago Campos Ferreira [UL, Intern, from Oct 2025]
- Baptiste Diedler [INRIA, Intern, from Jun 2025 until Jul 2025]
- Titouan Le Pen [UL, Intern, from May 2025 until Jun 2025]
- Bastien Pichet [INRIA, Intern, from May 2025 until Jun 2025]
- Achille Razafimaharo [UL, Intern, from May 2025 until Jun 2025]
- Lucas Villaume [UL, Intern, from Apr 2025 until Aug 2025]

Administrative Assistants

- Emmanuelle Deschamps [INRIA]
- Elsa Maroko [CNRS]
- Cecilia Olivier [INRIA]

External Collaborator

- Pascal Fontaine [ULIEGE, HDR]

2 Overall objectives

The objectives of Mosel-VeriDis are to contribute to advances in verification techniques, including automated and interactive theorem proving, and to make them available for the development and analysis of concurrent and distributed algorithms and systems, based on mathematically precise and practically applicable development methods. The techniques that we develop are intended to assist designers of algorithms and systems in carrying out formally verified developments, where proofs of relevant properties, as well as bugs, can be found with a high degree of automation.

Within this context, we work on techniques for *automated theorem proving* for expressive languages based on first-order logic, with support for theories (including fragments of arithmetic or of set theory) that are relevant for specifying algorithms and systems. Ideally, systems and their properties would be specified using high-level, expressive languages, errors in specifications would be discovered automatically, and finally, full verification could also be performed completely automatically. Due to the fundamental undecidability of the problem, this cannot be achieved in general. Nevertheless, we have observed important advances in automated deduction in recent years, to which we have contributed. These advances suggest that a substantially higher degree of automation can be achieved over what is available in today's tools supporting deductive verification. Our techniques are developed within trail-based solving and saturation-based reasoning, the two main frameworks of contemporary automated reasoning, of which respectively SMT (Satisfiability Modulo Reasoning) and superposition are the current most prominent examples in first- and higher-order logic. These two frameworks have complementary strengths and weaknesses. Figuring out how and when to make them converge is part of our interests. Techniques developed within the symbolic computation domain, such as algorithms for quantifier elimination for appropriate theories, are also relevant, and are part of 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 an established and highly successful technique for verifying systems and for finding errors. Our contributions in this area more specifically target quantitative aspects, in particular the verification of timed or probabilistic systems. A specificity of Mosel-VeriDis is 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 or have been integrated in verification platforms developed by other groups. 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 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. When correctness properties have been formally verified for a high-level specification of an algorithm, the correctness of an implementation of an algorithm still remains to be checked, using techniques such as refinement proofs, code generation, testing or trace validation.

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

Members of our group design effective quantifier elimination methods and decision procedures for algebraic theories, supported by their efficient implementation in the Redlog [5] and Logic1 systems.

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 Mosel-VeriDis have ample experience in the specification and subsequent machine-assisted, interactive verification of algorithms. In particular, we contribute to the development of methods and tools for verifying properties of specifications written in the TLA$^+$ [44] language, partly supported by the TLA$^+$ Foundation. Our group in

[1] Susceptible – Exposed – Infectious – Removed

particular develops the TLA$^+$ Proof System where proofs are expressed in a declarative language and that 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.

At the intersection of automated and interactive theorem proving, members of Mosel-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. Moreover, members of the group actively collaborate with members of the LORIA teams Mocqua and Pesto, mobilizing theorem proving techniques and tools to address verification and certification challenges from quantum computing and cryptographic protocol verification.

3.2 Formal Methods for Developing and Analyzing Algorithms and Systems

Theorem provers are not used in isolation. We are most interested in their support of sound methodologies for modeling and verifying systems. In this respect, members of Mosel-VeriDis have gained expertise and recognition in making contributions to formal methods for concurrent and distributed algorithms and systems [3, 9], and in applying them to concrete use cases. In particular, the concept of *refinement* [32, 34, 47] in state-based modeling formalisms is central to our approach because it allows us to present a rational (re)construction of system development. An important goal in designing such methods is to establish precise proof obligations, many of which can be discharged by automatic tools. This requires taking into account specific characteristics of certain classes of systems and tailoring the model to concrete computational models. Our research in this area is supported by carrying out case studies for academic and industrial developments. This activity benefits from and influences the development of our proof tools.

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

Today, the formal verification of a new algorithm is typically the subject of a PhD thesis, if it is addressed at all. This situation is not sustainable: 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 Mosel-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 Mosel-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 only has access to the execution time; several recent works address this direction.

Correctness by construction Verification methods are used for a wide variety of algorithm classes. An alternative to these verification methods involves design methods aimed at producing algorithms

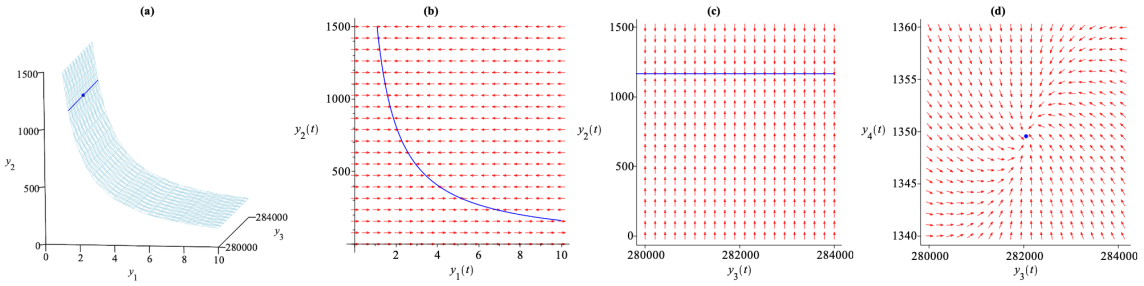


Figure 1: Illustration of the analysis of an epidemic model of avian Influenza A.

or programs that are correct by construction. This approach is based on the notion of refinement, and we aim at identifying fairly general patterns that guide the designer's steps. In [30], we contrast three techniques for constructing correct algorithms by construction.

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 [45] 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, researchers in our group 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 [46]. 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 $\mathscr{M}_3$. The constant factors appearing in the corresponding differential equations indicate that the system T_2 is 125 times slower than T_1, and that T_3 is another 125 times slower. This multiple time scale reduction emphasizes a cascade of successive relaxations of model variables. Figure 1(a) shows the surface of $\mathscr{M}_1$ projected into 3D space, with the line and the dot representing the submanifolds $\mathscr{M}_2$ and $\mathscr{M}_3$. Figure 1(b) illustrates the direction field of T_1 projected into 2D space. The curve corresponds to $\mathscr{M}_1$, indicating that the population of susceptible birds relaxes and that these variables reach quasi-steady state values. Figure 1(c) represents the direction field of T_2 on $\mathscr{M}_1$ projected into 2D space. The line corresponds to $\mathscr{M}_2$, 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 [31]. 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 peer-to-peer networks 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 [33]) as an underlying formalism to solve practical questions. Our work on parametric timed automata is partly motivated by applications in cybersecurity. Foundational decidability results and novel notions of non-interference and opacity for this class of automata allow us, for example, to determine the maximal frequency of attacker actions for the attack to succeed (i.e., so that these actions remain invisible to the external observer). Our contributions give rise to implementations in the Imitator model checker.

5 Highlights of the year

Vincent Trélat received the best paper award at ABZ, the International Conference on Rigorous State-Based Methods, for his paper on safely encoding B proof obligations in SMT-LIB [23] .

6 Latest software developments, platforms, open data

6.1 Latest software developments

6.1.1 Goeland

Name: Goeland

Keywords: First-order logic, Automated deduction, Automated Reasoning, Proof, Certification

Functional Description: Goeland is an automated theorem prover for first-order logic. It relies on a concurrent tableaux-based proof-search procedure that allows it to conduct a fair branch exploration. The prover can perform deskolemization and produce machine-checkable proofs in Rocq, LambdaPi and SC-TPTP. It supports TPTP FOF and TFF files.

News of the Year: In 2025, the main new developments were as follows:

- redesign of type system,
- support for Rocq output,
- support for SC-TPTP output.

Johann Rosain (ENS Lyon) contributes to the development and maintenance of Goeland.

URL: https://github.com/GoelandProver/Goeland

Publication: tel-04526940

Contact: Julie Cailler

Participants: Julie Cailler, Johann Rosain

6.1.2 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

Participant: Sorin Stratulat

6.1.3 TLAPS

Name: TLA+ proof system

Keyword: Proof assistant

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

News of the Year: In 2025, the main new developments were:

- The integration of TLAPS into the TLA$^+$ Virtual Studio Code Extension was consolidated.
- A solution for importing the abstract syntax tree from the standard SANY parser for TLA$^+$ was explored, in view of replacing the bespoke parser underlying TLAPS.
- The standard library, as well as the collection of examples, were consolidated and extended.
- Various bug fixes.

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

Contact: Stephan Merz

Participants: Damien Doligez, Stephan Merz

Partner: Microsoft

6.1.4 Logic1

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

Scientific Description: Logic1 offers a robust framework for working with first-order formulas. Its implementations are designed to be generic and parameterized by theories in the sense of first-order logic, currently supporting the theory of Sets and the theory of Real Closed Fields. Included algorithms cover a range of tasks, such as computing normal forms (CNF, DNF, NNF, PNF), simplification, and quantifier elimination.

Comprehensive documentation is available at docs.logic1.eu and on GitHub.

Functional Description: First-order logic recursively builds terms from variables and a specified set of function symbols with specified arities, which includes constant symbols with arity zero. Next, atomic formulas are built from terms and a specified set of relation symbols with specified arities. Finally, first-order formulas are recursively built from atomic formulas and a fixed set of logical operators.

Logic1 focuses on interpreted first-order logic, where the above-mentioned function and relation symbols have implicit semantics, which is not explicitly expressed via axioms within the logical framework. Typical applications include algebraic decision procedures and, more generally, quantifier elimination procedures, e.g., over the real numbers.

News of the Year: Version 0.2.0 of Logic1 was released on February 12, 2025, via Conda-Forge. This release introduces a parallel implementation of real quantifier elimination by virtual substitution up to degree two, based on the framework of Kosta (2016). The framework permits instantiation for higher degrees through appropriate virtual substitution tables.

The simplifier extends the standard Redlog simplifier (Dolzmann–Sturm, 1995) by allowing global substitution of implicit identities of the form x = q and x = q * y, where x and y are variables and q is a rational number. In addition, Logic1 now supports rational polynomial coefficients rather than being restricted to integer coefficients.

Finally, a programmatic interface has been added to provide convenient access to essential Redlog functionality via process communication. This interface includes several variants of real quantifier elimination and simplification. In 2025, Logic1 exceeded 1,000 downloads.

Besides members of VeriDis, Nicolas Faroß (Chalmers University) actively contributes to the development of Logic1.

URL: https://github.com/logic1-eu/logic1

Contact: Thomas Sturm

Participants: Thomas Sturm, Nicolas Faroß, Lorenz Leutgeb

6.1.5 Redlog

Name: Reduce Logic System

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

Functional Description: Redlog is an integral part of the interactive computer algebra system Reduce. It supplements Reduce's comprehensive collection of powerful symbolic computation methods by supplying more than 100 functions on first-order formulas.

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

News of the Year: While development efforts have primarily focused on the successor system Logic1, Redlog remains available and is committed to continued full support in the long term. Historically, many authors from different institutions contributed to the development of Redlog, the main maintainers have been Thomas Sturm and Andreas Dolzmann (Schloss Dagstuhl).

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

6.2 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 whose key expertise resides in natural sciences. This creates a situation where SBML features may be used in unexpected ways. 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 gained further recognition and citations during 2025.

URL: 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 Deduction Techniques

Graph Superposition for Quantum Circuits *Julie Cailler, Florent Krasnopol, Sophie Tourret, joint work with Noé Delormes and Simon Perdrix (project team Mocqua).*

Quantum circuits are studied by members of the Mocqua team, as a means to represent quantum computing algorithms. In this context, a series of equations, called the coherence equations, define an equivalence relation between circuits that represent the same program. Being able to detect the equivalence of two circuits by this relation is a central problem in this research domain.

The circuits can be naturally represented using hypergraphs. We are adapting a technique for automated reasoning on graphs that uses the superposition calculus to work on circuits represented as a particular class of graphs.

Challenging Benchmarks for Circuit Equivalence Checks in SMT-LIB and TPTP *Julie Cailler, Sophie Tourret, joint work with Noé Delormes (project team Mocqua).*

Circuits, quantum or otherwise, can be represented by first-order terms satifying a given predicate ensuring that some constraints that are natural on the circuits are also true on corresponding terms. By adding constraints to the coherence equations, it is also possible to represent them as clauses in first-order logic with arithmetic. We have produced several encodings of this problem and evaluated the performances of the state-of-the-art automated theorem provers (ATPs) Vampire and cvc5 on them. The analysis of the results suggests interesting directions of further research for ATPs.

GNN for Word Equations Solving *Julie Cailler joint work with Parosh Aziz Abdulla, Mohamed Faouzi Atig, Chencheng Liang (Univ. of Uppsala, Sweden), Philipp Rümmer (Univ. of Uppsala, Sweden & Univ. of Regensburg, Germany).*

Reasoning within theories such as arithmetic, arrays, and algebraic data structures has become a key challenge in automated reasoning. To address this, Satisfiability Modulo Theories (SMT) solvers have been developed, offering efficient decision procedures for a wide range of theories, including string theory. Strings, as a fundamental data type in programming, are crucial for many domains like text processing, database management, and web applications.

We developed a new algorithm that leverages a Graph Neural Network (GNN)-guided approach for solving word equations, building upon the well-known Nielsen transformation for equation splitting. We extend this algorithm to deal with conjunction of word equations. To handle the variable number of conjuncts, three approaches to adapt a multi-classification task to the problem of ranking equations are proposed. The training of the GNN is done with the help of minimum unsatisfiable subsets (MUSes) of word equations.

The algorithm is implemented in a solver named DragonLi. Experimental results on both artificial and real-world benchmarks demonstrate the efficiency of DragonLi in solving satisfiable problems. A paper presenting this work was published at FroCoS 2025 [13].

A Framework for Certifying Tableaux Proofs in Rocq *Julie Cailler joint work with Johann Rosain (ENS de Lyon).*

The free-variable tableau method has been widely used in order to automate proofs in multiple kinds of logics. Many Automated Theorem Provers (ATPs) still use this method, either because it is the only one available (e.g., in modal logics) or in order to produce proofs. However, as far as the authors know, its results have never been formalized. Together with Johann Rosain, we develop TableauxRocq, a deep-embedding of free-variable first-order tableaux in the Rocq prover that uses the Skolemization framework due to Cantone and Nicolosi-Asmundo. This tool can be used as a certifier of ATPs by adapting the Goéland prover to output proofs in the TableauxRocq format, thus seeing a significant speed-up in the checking time.

Decision procedures for fragments with arithmetic. *Pascal Fontaine, joint work with Bernard Boigelot, Thomas Braipson, Tom Clara, and Baptiste Vergain (Univ. of Liège).*

Arithmetic reasoning is an important aspect of automated deduction applied to verification. Fragments mixing arithmetic and uninterpreted symbols are often undecidable. We study the decidability frontier of such fragments. In particular, we are examining monadic first-order logic with order interpreted over the real numbers. This fragment is decidable, whereas adding the successor (or more precisely the "+1") function is undecidable. Although this result has been known for some time, no effective decision procedure exists. We want to develop an effective procedure, first, because a fundamental understanding of this fragment is scientifically interesting, and second, because this deep understanding can be inspirational for new SMT quantifier instantiation techniques in the presence of arithmetic. We previously designed and published one key element of this decision procedure, namely the emptiness test of automata on linear orderings [37]. In 2025, we have made progress towards the development of a quantifier elimination method for automata on linear orderings.

The SMT-LIB standard. *Pascal Fontaine, joint work with Clark Barrett (Stanford University), Cesare Tinelli (University of Iowa).*

We are involved in the standardization of the SMT-LIB language, a widely adopted format for the input of SMT solvers. In 2025, we released version 2.7 of the language, a transitional version introducing maps (lambdas) and polymorphism. We are currently working on version 3.0 which will be based on higher-order logic, and will introduce a flexible concept of modules, to replace the 2.x concepts of theories and logic.

7.2 Interactive Theorem Proving

An Extension of the TPTP Derivation Format for Sequent-Based Calculi. *Julie Cailler, joint work with Simon Guilloud, Sankalp Gambhir, Auguste Poiroux, Yann Herklotz, Thomas Bourgeat, and Viktor Kunčak (EPFL, Switzerland).*

We introduce SC-TPTP, an extension of the TPTP derivation format for supporting proofs expressed in the sequent formalism, enabling seamless proof exchange between interactive theorem provers and first-order automated theorem provers. We provide a way to represent non-deductive steps—Skolemization, clausification, and Tseitin normal form—as deductive steps within the format. Building upon the existing support in the Lisa proof assistant and the Goéland theorem prover, the SC-TPTP ecosystem is further enhanced with proof output interfaces for Egg and Prover9, as well as proof reconstruction support for HOL Light, Lean, and Rocq. A publication of this work was accepted at CADE [22].

Integration of Abduction in Isabelle/HOL. *Tiago Campos Ferreira, Sophie Tourret, joint work with Haniel Barbosa (Universidade Federal de Minas Gerais, Brazil).*

In the context of the associate team Carma (cf. Section 9), Sophie Tourret and Haniel Barbosa have started to work on a new collaborative project. Tiago Campos Ferreira, a master student under Haniel Barbosa's supervision in Brazil, joined VeriDis in October 2025 for an internship of six months. His objective is the integration of cvc5's abduction mechanism into the proof assistant Isabelle/HOL. Abduction is a logical operation that provides tentative explanations to statements in a given theory. For a proof assistant, it could be turned into a tool that suggests missing hypotheses or intermediary steps in proofs. The SMT solver cvc5 is currently the best tool for abduction modulo theories, and Isabelle/HOL is the proof assistant with the best automation. By combining both into a prototype tool, we want to identify robustly the bottlenecks of abduction and address them until abduction becomes useful in practice to help proof engineers.

Formalization of Modal Model Theory in Isabelle/HOL *Sophie Tourret, joint work with Jasmin Blanchette and Yiming Xu (LMU Munich, Germany).*

As a well-established framework for talking about relational structures, propositional modal logic has

been used and investigated for decades [36]. We formalized the finite tree-like model property of multi-modal logic in Isabelle/HOL, which considers families of modal operators. This property is valuable because a modal logic is decidable if it has the finite model property and is finitely axiomatizable. To obtain a general statement independent of the type of worlds constituting the universe of the semantics, we used Isabelle/HOL's HOLZF extension to prove the theorem.

Lean to DHOL Translation *Sophie Tourret, joint work with Jasmin Blanchette, Alexander Bentkamp and Luca Maio (LMU Munich, Germany).*

A new dependently-typed higher-order logic called DHOL has recently been introduced [49] as a bridge between existing dependently-typed logics, on which several major proof assistants are based (e.g., Rocq, Lean), and classical logics that benefit from strong automation. Whereas the translation from DHOL to classical HOL was introduced together with DHOL, the translation between DHOL and other type theories remained to be established. We define this translation and establish the appropriate theoretical guarantees of soundness and completeness. A publication of this work is in preparation.

Certification of Rewriting on Circuits *Julie Cailler, Sophie Tourret, joint work with Noé Delormes and Simon Perdrix (project team Mocqua).*

Circuits as studied in the Mocqua team are instances of signal flow graphs that belong to the symmetric monoidal category known as PROP. Circuits can also be interpreted as directed hypergraphs with fixed inputs and outputs, but this interpretation is not injective and thus builds equivalence classes of circuits that correspond to the same graph. This equivalence between circuits can be defined by a set of equations called the coherence equations. Thus, a straightforward proof that two circuits are equivalent consists in a sequence of atomic transformations, each justified by one application of a coherence equation. We want to produce certificates for these transformations that can be machine checked. An obvious approach to handle this kind of problems is to rely on terminating and confluent term rewrite systems (TRSs), that can rewrite equivalent terms to a unique normal form. However, there is no such TRS that can handle the coherence equations. We devised a technique to produce certificates by decomposing the problem into two subproblems, that of state-and-effect-free PRO (the fragment of PROP where wires cannot cross each other, corresponding to isoplanar graphs), and that of trivial PROPs (corresponding to permutations of wires) that we have proved to both possess terminating and confluent TRSs. We are working at producing these certificates in Isabelle/HOL, using its simplifier as the term-rewriting engine to normalize terms. We have achieved a proof-of-concept implementation for state-and-effect-free PROs, and are now working on the same for trivial PROPs, before we finally consider full PROPs.

Reconstruction of SMT Proofs in Lambdapi. *Alessio Coltellacci, Stephan Merz, joint work with Bruno Andreotti and Haniel Barbosa (Universidade Federal de Minas Gerais, Brazil) and with Frédéric Blanqui (project team Deducteam).*

SMT solvers are widely considered as the tools of choice for the automatic verification of programs and systems. Their integration with skeptical proof assistants requires proofs found by SMT solvers to be certified by the trusted kernel of proof assistants. The Alethe format [50] represents a trace of an SMT proof, explaining why a set of SMT constraints is unsatisfiable. In this work, we aim at interpreting Alethe proof traces and generating corresponding proofs that are accepted by the Lambdapi proof checker, a foundational proof assistant based on dependent type theory and rewriting rules that is intended to serve as a pivot for exchanging proofs between interactive proof assistants. Whereas elementary Alethe rules can directly be mirrored by corresponding proof rules derived in Lambdapi, some rules describe an algorithm for checking an assertion. This is true in particular for rules about n-ary connectives, as well as for checking theorems in linear (integer or real) arithmetic. In these cases, we leverage the technique of proof by reflection by setting up a semi-decision procedure on a suitable data type.

A first paper covering the pure logic fragment of SMT-LIB was accepted for publication in the journal Acta Informatica (to appear in 2026), a second paper focusing on the reconstruction of proofs in linear arithmetic was presented at FroCoS 2025 [18]. A comprehensive description of the approach appears in the PhD thesis of Alessio Coltellacci, to be defended in January 2026.

An Isabelle/HOL framework to Add Splitting on Top of Saturation Calculi *Ghilain Bergeron, Florent Krasnopol, Sophie Tourret.*

Saturation-based calculi, such as resolution and superposition, are implemented in theorem provers to establish the satisfiability of sets of formulas in various logics. In some circumstances, such calculi introduce needless dependencies in their steps. Splitting is a technique that removes such dependencies by introducing branching between independent pieces of formulas. It is very successful and has been adopted in various forms by most state-of-the-art provers. We have formalized in Isabelle/HOL the splitting framework [42] that models several forms of splitting in a modular way. The formalization focuses on a first model of splitting and an instance of resolution extended with splitting called Lightweight AVATAR. The results of this work were presented at ITP 2025 [16].

Towards the Verification of ProVerif in Isabelle/HOL *Qi Qiu, Sophie Tourret, joint work with Steve Kremer (project team Pesto).*

ProVerif is a cryptographic protocol verifier developed by project team Prosecco with real-world applications (including TLS or the Signal messaging protocol). The Inria team Pesto uses ProVerif and made several contributions to the tool. At its heart, ProVerif is built upon the well-known resolution calculus applied to Horn clauses, and the tailoring to security protocols lies in the encoding of protocols as logic formulas rather than in the reasoning. We have started a direct formalization in Isabelle/HOL of the core technique in ProVerif, leveraging our previous work on the formalization of saturation-based calculi including resolution [51].

Analysing the Divergence of Induction-based Proofs. *Sorin Stratulat.*

Primal grammars have been used recently to detect divergence in induction-based and automatically-generated proofs [39]. A divergence is detected if, from a fixed number n (> 2) of generated conjectures, one can build a Presburger system that can subsume further conjectures which are no longer processed. We have shown that the Presburger systems built in that way can also subsume false conjectures, for any n, which limits drastically the use of primal grammars in sound induction-based inference systems.

A Verified Encoding of Proof Obligations. *Vincent Trélat, Ghilain Bergeron, Stephan Merz, Sophie Tourret.*

The B and Event-B methods for formal system development are widely used for developing certified embedded systems, for example in the railway domain. These methods identify a significant number of proof obligations that must be discharged in order for a system development to be acceptable for certification authorities. Since a high degree of proof automation is essential in an industrial context, proofs are routinely delegated to automatic theorem provers, including SMT solvers. In a recent experiment conducted by the Clearsy company, among the roughly 77,000 proof obligations of a representative development project, 64% were proved automatically by the Atelier B tool suite, leaving 28,000 obligations to be proved by human interaction. The existing encoding of B proof obligations for SMT solvers [40] systematically reduces formulas to first-order logic, eliminating all constructs of set theory. A drawback of this approach is that it destroys the structure of formulas, in particular for constructs such as set comprehension that involve binders. In this work, we develop an alternative encoding that takes advantage of recent capabilities of SMT solvers to handle fragments of higher-order logic.

An encoding in SMT-LIB, the input language of SMT solvers, of proof obligations based on these ideas has been implemented in the Lean proof assistant, and an evaluation on the fragment of the benchmark mentioned above that is covered by the encoding indicates that this translation is a useful complement to the existing first-order encoding: the solvers succeed in proving a high number of obligations that they failed to prove for the previous encoding. A paper describing this work [23] was presented at ABZ 2025 and received the best paper award.

By developing the encoding in Lean, it becomes possible to establish its correctness within that proof assistant. To this end, the semantics of B and SMT formulas have been formalized in Lean, and work is in progress in order to formally prove the correctness of the translation.

Moreover, a tool has been developed for representing proof obligations corresponding to formal developments in B directly in Lean, with a particular emphasis on identifying well-definedness conditions arising from possibly undefined expressions, such as the minimum value of a set or the application of a partial function. A paper describing this work is in preparation for submission to an international conference.

7.3 Formal Methods for Developing and Analyzing Algorithms and Systems

Formalization and Implementation of Safe Destination Passing in Pure Functional Programming Settings. *Thomas Bagrel, Horatiu Cirstea, joint work with Arnaud Spiwack (Tweag & Modus Create).*

Destination-passing style programming introduces destinations, which represent the address of a write-once memory cell. These destinations can be passed as function parameters, allowing the caller to control memory management: the callee simply fills the cell instead of allocating space for a return value. While typically used in systems programming, destination passing also has applications in pure functional programming, where it enables programs that were previously not expressible using ordinary immutable data structures.

We developed a core λ-calculus with destinations [14]. This new calculus is more expressive than similar existing systems, with destination passing designed to be as flexible as possible. This is achieved through a modal type system combining linear types with a system of ages to manage scopes, in order to make destination-passing safe. Type safety of our core calculus was proved formally with the Rocq proof assistant.

Then, we see how this core calculus can be adapted to an existing pure functional language, Haskell, whose type system is less powerful than our custom theoretical one [27]. Retaining safety comes at the cost of removing some flexibility in the handling of destinations. We later refine the implementation to recover much of this flexibility, at the cost of increased user complexity. The prototype implementation in Haskell shows encouraging results for adopting destination-passing style programming when traversing or mapping over large data structures such as lists or data trees. A comprehensive presentation of these results appears in Thomas Bagrel's PhD thesis [27].

Verified Code Generation from PlusCal Programs. *Ghilain Bergeron, Horatiu Cirstea, Stephan Merz.*

Specifications written in high-level languages such as TLA$^+$ are useful for verifying correctness properties. They often result in state spaces that remain manageable for model checking, and they allow users to design inductive invariants that are at the heart of deductive system verification. However, there is a substantial gap between high-level specifications of algorithms and implementations of those algorithms in actual programs. One way to avoid this gap is to translate specifications into code in a programming language that can be compiled and executed. In this work, we investigate the feasibility of such an approach for algorithms written in Distributed PlusCal, an algorithmic language that can be translated to TLA$^+$ for verification. We define a series of PlusCal fragments together with semantics-preserving translations from one fragment to the next one, and then aim at implementing a code generator for the most restrictive fragment.

A translator from a suitable fragment of Distributed PlusCal to the Go language has been implemented in Lean based on this approach, and the correctness of the first two phases of the translation has been proved. This work was presented at WPTE 2025 [24], and we have been invited to prepare an extended paper for an international journal. Work on formal proofs of the correctness of the translation in the Lean proof assistant is ongoing.

Formalization and verification of a train scheduler in TLA$^+$. *Martin Vassor, Lucas Villaume, joint work with Guillaume Bonfante (Carbone Team of LORIA).*

In the context of Cyber Humanum Est [38], Guillaume Bonfante created a small scale train model. Trains run on routes automatically, following traffic lights. A scheduler is in charge of controlling railroad switches and traffic lights, thus indirectly controlling trains. The model is a simplified version of rules used in real train networks.

In order to prevent crashes (two trains colliding) or deadlocks (two trains being locked, e.g. facing each other), one has to ensure that the scheduler never allows such situations to occur. Therefore, we formalized the traffic rules in TLA$^+$, allowing to model check the correctness of a schedule. In a second step, we show how schedules can be composed while remaining correct, allowing one to manage large train networks.

Reversibility for Fault Tolerance in Typed Sessions. *Martin Vassor, joint work with Adam Barwell (University of St. Andrews, Scotland), Ping Hou (University of Oxford, England), and Nobuko Yoshida (University of Oxford, England).*

Multiparty Session Types (MPST) are a family of type-based techniques developed to prevent various classes of bugs in message-passing concurrent programs, such as the absence of deadlock. MPST are also used as a form of protocol specification, where the well-typedness of a program ensures its conformance to the specification (session fidelity). However, MPST often assume a reliable model of communication, i.e. without message loss or process faults, and existing approaches to relax this assumption either augment MPST with additional elements to deal with faults (e.g. placing explicit checkpoints in the protocol specification), or lower the guarantees provided (e.g. Affine MPST, which safely stops the whole system in case of a process failure, losing session fidelity).

In this project, we aim to implement MPST on top of a *reversible* variant of the higher-order π-calculus. In such a setting, upon the failure of a process, we expect the system to transparently revert to a previous state, thus recovering from the failure. Contrary to existing approaches based on reversibility, we do not modify the syntax or the semantics of MPST, allowing users to transparently switch from standard MPST to fault-tolerant MPST.

A paper describing this work was published at the International Conference on Reversible Computing [15].

Bounded Reversibility in HOπ *Martin Vassor, joint work with Ivan Lanese (University of Bologna and Inria - Université Côte d'Azur, Italy), and Claudio A. Mezzina (University of Urbino, Italy).*

Reversible process calculi are variants of process calculi such as CCS or π-calculus, which are equipped with forward and backward semantics. Forward semantics correspond to the usual ones of those process calculi, while backward semantics allows for actions to be *undone* (e.g., undoing a message reception). When designing such calculi, the community often aims at *complete* and *causally-consistent* backward semantics, where causal consistency means that states reached using backward reduction rules ought to be reachable from the initial state using only forward rules (i.e., reverting does not allow new states to be reached), and completeness means that any such state ought to be reachable.

While those two key properties are desirable, as they lead to nice theoretical frameworks, they are not suitable for practical use. Indeed, completeness entails that enough information has to be kept during execution to reach the initial state. Causal consistency entails that reverting part of the system possibly spreads to all components. Those two consequences are problematic in practice: the first one due to the amount of information that has to be kept forever, even if it is not used in the end. In the second one, the spread of rollback disallows standard software architectures such as client-server, in which one would not allow the rollback of a client to spread to the server, and then to other clients.

[26] contains preliminary work addressing those issues. We drafted a reversible higher-order π calculus with additional primitives for committing messages, preventing them to be rolled back in the rest of the computation (thus breaking completeness, but allowing the garbage-collection of rollback information). We also introduce non-causally-consistent uses of communication channels, in which case the rollback spread is not propagated through such uses (thus breaking causal consistency, but allowing some spatial control of rollbacks).

An inductive invariant for a high-level Raft specification *Volkan Burakcin, Stephan Merz.*

Raft [48] is a protocol for achieving iterated consensus that is used in the implementation of many distributed systems. Although the original author of Raft accompanied the design by a TLA$^+$ specification, it is written at a very low level of abstraction and is therefore hard to verify using model checking or

theorem proving. We have been working on a more high-level specification of the protocol that makes it easier to understand the main mechanisms underlying Raft. During the internship of Volkan Burakcin, an inductive invariant for Raft was designed and proved correct using TLAPS, the TLA$^+$ Proof System, cf. Section 6.1.3. The invariant implies the main correctness properties of Raft, in particular the existence of a single leader for any given term, the agreement of two entries at the same index provided the terms of the entries are identical, the fact that any leader contains all committed entries, and the property that committed entries are stable.

7.4 Algorithmic Verification

Degradation of stochastic systems *Baptiste Diedler, Marie Duflot-Kremer, Engel Lefaucheux, Bastien Pichet.*

Diagnosis aims at providing information about specific unobservable behaviors of a system, such as failures. The study of diagnosis in stochastic systems, such as those based on Markov chains, is now well-developed. For instance, in [35], the authors demonstrate, for various forms of diagnosis, how to determine whether a system is diagnosable and, if so, how to construct a diagnoser, i.e., a tool that translates a sequence of observations into a verdict.

During the internships of Baptiste Diedler and Bastien Pichet, we extended this classical framework by introducing the notion of degradation. Degradation generalizes diagnosis by aiming not only to detect whether a fault has occurred, but also to quantify the amount of damage suffered by the system. In this setting, failures may accumulate over time, and the objective becomes to detect when the accumulated degradation exceeds a given threshold. This work is ongoing, with the goal of submitting a paper next year.

Multi-agent transfer systems *Engel Lefaucheux, joint work with Nathalie Bertrand, Loïc Helouet and Luca Paparazzo (project team Devine).*

In [28], we introduced multi-agent transfer systems, a cooperative model in which agents move on individual weighted arenas while managing non-negative energy levels and sharing energy with peers. Each agent has a target vertex, and the goal is to reach a global configuration where all targets are achieved simultaneously. We considered different execution semantics (namely asynchronous, strongly synchronous, and weakly synchronous), depending on whether the agents need to act simultaneously or not. Such a game under either the asynchronous and strongly synchronous semantics can be turned into a Petri net. However, the weakly synchronous semantic produces a model that is strictly more expressive than Petri nets.

We investigated the computational complexity of the resulting global reachability problem. For asynchronous and strongly synchronous semantics, we established decidability and derived complexity bounds ranging from NP to EXPSPACE, relying on monotonicity and bounded witness arguments. In contrast, we showed that reachability becomes undecidable under weakly synchronous semantics due to the loss of monotonicity.

7.5 Foundational Research in Arithmetic Reasoning

On the number of real types of univariate polynomials. *Thomas Sturm, joint work with Thomas Faroß (Chalmers University).*

We consider univariate polynomials with real coefficients. The *real type* of a family $f_1, \ldots, f_n$ of such polynomials is their combined sign behavior from $-\infty$ to ∞, presented as column vectors of length n. For instance, all real roots of $f_1 = x+1$, $f_2 = 2x+1$, $f_3 = x^2-1$ are located at -1, $-\frac{1}{2}$, 1. Evaluation of the signs of $[f_1, f_2, f_3]^T$ at and between those points yields the real type

$$\begin{bmatrix} -1 & 0 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & -1 & 0 & 1 \end{bmatrix}.$$

We are specifically interested in the number of possible real types of families of n polynomials subject to a degree bound d. Košta [43, pp.11–13] has given a formula for the number R_d of possible real types of a single polynomial of degree d in terms of summations of certain binomial coefficients, and conjectured that this number could not be expressed in a closed form. We are not aware of any former results for $n > 1$.

Considering finite families $f_1, \ldots, f_n$ of polynomials of degrees $d_1, \ldots, d_n$, and denoting by m the finite cardinality of the union of their real roots, we have derived the following results:

1. In the special case $n = 1$, we simplify Košta's result mentioned above to a simple case distinction based on Fibonacci numbers, which can be expressed in closed form, in contrast to sums of binomial coefficients.

2. On these grounds, we derived another closed form for the number $\widehat{R}_d$ of all real types realized by a single polynomial *up to* degree d. It follows that both $R_d, \widehat{R}_d \in \Theta(\varphi^d)$, where φ is the golden ratio, and Θ is the Bachmann–Landau symbol for asymptotic growth of the same order.

3. For the general case $n \geq 1$, we derive a formula for the number $R^{(m)}_{d_1,\ldots,d_n}$ of all real types realized by polynomials with degrees d_i and m distinct real roots. Summation over m yields an explicit form for the number $R_{d_1,\ldots,d_n}$ of all real types realized by polynomials with degrees d_i and any number of roots, which generalizes Košta's original result.

4. For the number of all real types realized by polynomials with any choice of $d_1, \ldots, d_n$ and m distinct real roots, we can reduce or formula for $R_{d_1,\ldots,d_n}$ to the closed form $S^{(m)}_n = 2^n \cdot (3^n - 1)^m$, which generalizes the obvious $S^{(m)}_1 = 2^{m+1}$. In particular, this imposes an upper bound on $R^{(m)}_{d_1,\ldots,d_n}$.

Real types for $n > 1$ play a key role in a number of decision and quantifier elimination procedures for real closed fields. Our results were published at ISSAC [21].

8 Bilateral contracts and grants with industry

8.1 Bilateral contracts with industry

Participants: Thomas Bagrel, Horatiu Cirstea, Marie Duflot-Kremer, Engel Lefaucheux, Dominique Méry, Stephan Merz, Mohamed Amine Snoussi.

Type systems for the memory safety of functional programs

Duration: April 2022 – March 2025

Industrial Partner: Tweag

Team participants: Thomas Bagrel, Horatiu Cirstea

Summary: In his PhD thesis [27] supported by a CIFRE contract, Thomas Bagrel studies type systems and corresponding constructions in a pure functional calculus with destinations at its core for guaranteeing programs that are memory safe and can be compiled to efficient machine code.

Reengineering protocols for industrial controllers

Duration: May 2023 – April 2026

Industrial Partner: Westinghouse France

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

Summary: In his PhD work supported by a CIFRE contract, Amine Snoussi aims at constructing formal models and simulations of protocols that are used for industrial controllers, in particular for the diagnosis and control of electronic components in nuclear power plants. He has modeled the main protocol for the interaction between a controller and a component as a system of timed automata and verified a number of properties using the UppAal model checker. He also developed a simulator of the protocol that can interact with an actual controller.

Event-B modeling for Human-Machine Interaction

Duration: June 2024 – June 2025

Industrial Partner: SAS H-AUGENPLUS

Team participants: Dominique Méry

Summary: The objective of this work is to assist in the development of Event-B models for HMI systems.

9 Partnerships and cooperations

9.1 International initiatives

9.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program

Title: CAtalyzing progRess in smt solving and proof assistants via Modularity, proof trAnslation, and proof reconstruction (CARMA)

Duration: 2024 - 2026

Coordinators: Sophie Tourret and Haniel Barbosa (UFMG)

Partners: Universidad Federal de Minas Gerais, Belo Horizonte (Brazil)

Inria contact: Sophie Tourret

Team participants: Tiago Campos Ferreira, Alessio Coltellacci, Stephan Merz, Sophie Tourret, Vincent Trélat

Summary: The Carma associate team aims at improving the state of the art in SMT solving on three fronts. Our first objective is to design a new SMT solver as a research vessel that emphasizes modularity over utmost performance. Our goal is that all components can simply be plugged in and out to make it easy to upgrade and serve as a platform for comparison between different techniques. The tentative name of this new solver is ModulariT. Our second objective concerns higher-order SMT and the missing components to make it competitive. We plan to provide an efficient conflict-based instantiation technique for higher-order logic for ModulariT. Our third objective concerns the coexistence of various proof formats for SMT solving. We aim at more interoperability so that the various formats do not create a divide in the community. Specifically, our aim will be to provide a translation of Alethe proofs generated by SMT solvers to Dedukti, a logical framework based on the $\lambda\Pi$-calculus modulo, that has become the lingua franca of proof translation. In 2024, we added a fourth objective to the team, that of studying the integration of SMT-based abduction in a proof assistant.

In 2025, there were again several exchanges between Belo Horizonte and Nancy, despite cancellations of trips in the spring related to the late vote of the budget in France. A master student from Belo Horizonte, Tiago Campos Ferreira is visiting us for a 4 months master internship funded by Orion, a program within the excellence initiative of Lorraine University. He is working on our fourth objective with Sophie Tourret and Haniel Barbosa. A PhD student from our team (Alessio Coltellacci) and a postdoc from Deducteam (part of the associate team) visited Belo Horizonte for 3 weeks to work on our first objective. Some more trips have been anticipated for early 2026 to counteract the previously mentioned delays.

9.1.2 Participation in other International Programs

Title: Artificial Intelligence based Robotics (AiRobo)

Partner Institution(s):

- University of Macedonia, Greece
- RWTH, Germany
- Eszterhazy Karoly Catholic University, Hungary
- West University of Timisoara, Romania

Date/Duration: December 2023 - November 2026

Team participant: Sorin Stratulat

Summary: AiRobo [41] is an ERASMUS+ project that aims to increase the quality and the attractivity of related departments at the partner universities, by a significant raise of the level of competence and skills of the relevant academic staff in the field of Artificial Intelligence based Robotics. The correct functioning of such safety-critical systems is ensured by formal verifications using theorem provers such as Theorema and Coq, as well as SAT and SMT solvers such as SMT-RAT.

9.2 National initiatives

ANR Project BiSoUS

Title: Better Synthesis for Underspecified Quantitative Systems

Duration: March 2023 – February 2027

Coordinator: Didier Lime, École Centrale de Nantes & LS2N

Partner Institutions:

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

Team participants: Marie Duflot-Kremer, Engel Lefaucheux

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

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

More information: BiSoUS Web site

ANR Project BLaSST

Title: Enhancing B Language Reasoners Using SAT and SMT Techniques

Duration: March 2022 – February 2027

Coordinator: Stephan Merz

Partner Institutions:

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

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

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

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

More information: BLaSST Web site

ANR Project EBRP

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

Duration: January 2020 – December 2026

Coordinator: Dominique Méry

Partner Institutions:

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

Team participants: Dominique Méry

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

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

More information: EBRP Web site

ANR Project ICSPA

Title: Interoperable and Confident Set-based Proof Assistants

Duration: January 2022 – December 2026

Coordinator: Catherine Dubois, ENSIIE & Samovar

Partner Institutions:

- ENSIIE & Samovar, Évry
- Inria (Nancy and Saclay research centers)
- University Paul Sabatier & IRIT, Toulouse
- University of Montpellier & LIRMM, Montpellier
- CLEARSY, Aix-en-Provence

Team participants: Alessio Coltellacci, Dominique Méry, Stephan Merz

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

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

More information: ICSPA Web site

10 Dissemination

10.1 Promoting scientific activities

- Julie Cailler:
  - Editor of the Newsletter of the Association of Automated Reasoning (AAR).

10.1.1 Scientific events: organization

General chair, scientific chair

- Julie Cailler: Artifact Evaluation co-chair of SPIN 2025 (Hamilton, Canada)
- Stephan Merz:

  - PC co-chair and co-organizer of the TLA$^+$ Community Meeting 2025 (Hamilton, Canada)
  - PC co-chair and co-organizer of the 12th Workshop on Formal Reasoning on Distributed Algorithms (Berlin, Germany)

- Sophie Tourret:
  - PC co-chair of the SMT workshop 2025 (Glasgow, UK)
  - PC co-chair and co-organizer of the workshop Weidenbach60 (Stuttgart, Germany).

Member of organizing committees

- Julie Cailler:
  - LVP days 2025, Strasbourg, France.
- Marie Duflot-Kremer:
  - Journées sciences et médias (together with journalists and other scientific associations) 2025, Paris, France.
- Pascal Fontaine:
  - Organizing committee of the summer school VTSA 2025 in Liège, Belgium.
- Stephan Merz:
  - Organizing committee of the summer school VTSA 2025 in Liège, Belgium.
- Sophie Tourret:
  - workshop chair of CADE 2025 (Stuttgart, Germany),
  - co-organizer of the “reflection” seminar, a joint project with Institut Élie Cartan de Lorraine and the Poincaré archives, that will take place as a series in Nancy in 2026,
  - co-organizer of the “deduction” seminar series for its next two iterations at Dagstuhl, Germany. One is already accepted and scheduled for early 2026.

10.1.2 Scientific events: selection

Member of conference program committees

- Julie Cailler: TASE 2025, FMCAD Student Forum 2025, FSTTCS 2025 (artifact PC committees: VMCAI 2025).
- Horatiu Cirstea: RuleML 2025.
- Engel Lefaucheux: QEST-FORMATS 2025.
- Dominique Méry: iFM2025, ABZ2025, TASE2025, ICFEM2025.
- Stephan Merz: ABZ 2025, CADE 2025, FMICS 2025, VECOS 2025.
- Thomas Sturm: CASC 2025, SYMCOMP 2025 (and workshop: SC-Square 2025).
- Sophie Tourret: CADE 2025, ITP 2025 (and workshops: SMT 2025, Weidenbach60).

10.1.3 Journals

Editor in Chief

- Thomas Sturm: Mathematics in Computer Science, Springer.

Member of the editorial boards

- Thomas Sturm: Journal of Symbolic Computation, Elsevier.

Reviewer - reviewing activities

- Julie Cailler: Journal of Applied Logic, Journal of Logical and Algebraic Methods in Programming, Science of Computer Programming.
- Horatiu Cirstea: book chapter for ISTE Press.
- Stephan Merz: Formal Methods in System Design, Software Tools for Technology Transfer (2 articles).
- Sophie Tourret: Journal of Artificial Intelligence Research, Transactions on Computational Logic, Automated Reasoning, Logical Methods in Computer Science.

10.1.4 Invited talks

- Julie Cailler:
  - Deskolemization: From Tableaux to Proof Certificates, CHoCoLa Meeting, Lyon, France
  - Goéland: A Concurrent Tableau-Based ATP that Produces Machine-Checkable Proofs, Euro-ProofNet School on Natural Formal Mathematics, University of Bonn, Germany
- Stephan Merz: The TLA$^+$ Framework – From High-Level Specifications to Distributed Programs, 20th Intl. Conf. Integrated Formal Methods, Paris, France.

10.1.5 Leadership within the scientific community

- Julie Cailler:
  - Co-chairperson of the LVP (*Langages et Vérification de Programmes*) group of GDR GPL.
- Stephan Merz:
  - chair of the TLA$^+$ Specification Language Committee,
  - member of the Governing Board of the TLA$^+$ Foundation,
  - member of IFIP WG 2.2 (Formal Specification of Programming Concepts).
- Thomas Sturm:
  - advisory positions in the EPRSC Projects EP/T015748/1 and EP/T015713/1, UK,
  - SC-Square steering committee member, invited in 2023.
- Sophie Tourret:
  - CADE trustee, elected in 2022, reelected in 2025,
  - PAAR workshop steering committee member, invited in 2020,
  - Ex-officio SMT trustee for 2025,
  - SAT/SMT/AR coordination committee member, invited in 2024.

10.1.6 Scientific expertise

- Stephan Merz: assessment of a promotion request at the University of Delaware, U.S.A.
- Sophie Tourret: reviewer for a project submitted to the Israel Science Foundation.

10.1.7 Research administration

- Horatiu Cirstea:
  - member of the hiring committee for an associate professor position at Toulouse INP - ENSEEIHT / IRIT.
- Stephan Merz:
  - coach for ERC project applications at Inria,
  - scientific referent for European affairs (project REIL) at University of Lorraine,
  - member of the *bureau du comité des projets* at Inria Nancy,
  - chairman of the hiring committee for an associate professor position at IDMC, University of Lorraine.
- Sophie Tourret:
  - EuroProofNet European COST action core member, deputy leader of working group 2 on automated theorem provers,
  - member of the doctoral commission of Inria centre at University of Lorraine,
  - member of the hiring committee for an associate professor position at Polytech Paris-Saclay,
  - member of the Bill McCune PhD Award committee.

10.2 Teaching - Supervision - Juries - Educational and pedagogical outreach

10.2.1 Teaching

The university employees of VeriDis have a statutory teaching obligation of 192 hours per year. We only list the teaching activities at the master level.

- Licence: Julie Cailler, Algorithms and imperative programming, 60 HETD, L1, Université de Lorraine, France.
- Licence: Julie Cailler, Referring teacher , 10 HETD, L1, Université de Lorraine, France.
- Licence: Julie Cailler, Agent-oriented programming, 50 HETD, L2, Université de Lorraine, France.
- Licence: Julie Cailler, System: processes, memory and files, 12 HETD, L2, Université de Lorraine, France.
- Licence: Julie Cailler, Student's projects supervision, 20 HETD, L2, Université de Lorraine, France.
- Licence: Julie Cailler, Functional programming, 30 HETD, L3, Université de Lorraine, France.
- Master: Julie Cailler, Software engineering, 20 HETD, 2A (M1), ENSEM, France.
- Master: Horatiu Cirstea, Programming paradigms, 32 HETD, M2 Informatique, Université de Lorraine, France.
- Master: Horatiu Cirstea, Software analysis and design, 80 HETD, M1 informatique, Université de Lorraine, France.
- Master: Horatiu Cirstea, Software engineering, 50 HETD, 2A (M1), ENSEM, Université de Lorraine, France.
- Licence: Marie Duflot-Kremer, Algorithms and programming 1, 90 HETD, L1, Université de Lorraine, France.
- Licence : Marie Duflot-Kremer, data bases, 12 HETD, L3, Université de Lorraine, France

- Licence : Marie Duflot-Kremer, Scientific Outreach, 30h ETD, L1, Université de Lorraine, France.
- Licence : Marie Duflot-Kremer online optional course on computer science, 30HETD, first year medicine students (PASS), Université de Lorraine,.
- Master: Marie Duflot-Kremer, Using unplugged activities to pass on computer science concepts to students, 35 HETD, master for future teachers, Université de Lorraine, France
- Master: Marie Duflot-Kremer, Elements of model checking, 16 HETD, M2 Informatique, Université de Lorraine, France.
- Master: Marie Duflot-Kremer, Distributed algorithms, 15 HETD, M1 Informatique, Université de Lorraine, France.
- Classe préparatoire universitaire: Engel Lefaucheux, Colles Langages et automates, 3 HETD, Université de Lorraine.
- Classe préparatoire universitaire: Engel Lefaucheux, remplacements cours Langages et automates, 7 HETD, Université de Lorraine.
- Licence: Engel Lefaucheux, Algorithms and programming 2, 20 HETD, L2, Université de Lorraine.
- Master: Marie Duflot-Kremer and Engel Lefaucheux, supervision of 3 students in a short *initiation à la recherche* internship, M1, Université de Lorraine
- Classe préparatoire des INP: Engel Lefaucheux, Langages et Automates, 34.5 HETD, Université de Lorraine
- Master: Dominique Méry, Formal Modeling for Software-based Systems 40 HETD, M2 Informatique, Université de Lorraine, France.
- Master: Dominique Méry, Models and algorithms, M1 Telecom Nancy, 48 HETD, Université de Lorraine, France.
- Master: Dominique Méry, Formal Modeling for Software-based Systems, M2 Telecom Nancy, 24 HETD, Université de Lorraine, France.
- Master: Stephan Merz, Elements of model checking, 16 HETD, M2 Informatique, Université de Lorraine, France.
- Master: Stephan Merz, Distributed algorithms, 15 HETD, M1 Informatique, Université de Lorraine, France.
- Master: Stephan Merz, Secure Coding, M1 Mines Nancy, 13 HETD, Université de Lorraine, France.
- Licence: Sorin Stratulat, Algorithms and Programming, 128 HETD, L1 Informatique, Université de Lorraine, France.
- Licence: Sorin Stratulat, Logic for Computer Science, 28 HETD, L1 Informatique, Université de Lorraine, France.
- Licence: Sorin Stratulat, Logic, 30 HETD, L2 Informatique, Université de Lorraine, France.
- Licence: Sorin Stratulat, Logic, 60 HETD, L3 Informatique, Université de Lorraine, France.
- Licence: Sorin Stratulat, Introduction to Relational Databases, 32 HETD, L1 Informatique, Université de Lorraine, France.
- Master: Thomas Sturm, Decision Procedures for Specific Theories (Seminar), Universität des Saarlandes, Germany.
- Master: Sophie Tourret, Secure Coding, M1 Mines Nancy, 13 HETD, Université de Lorraine, France.

- Licence: Martin Vassor, Introduction to the CS department, 12HETD, 4th years, Polytech Nancy, Université de Lorraine, France.
- Master: Martin Vassor, Distributed Algorithms, 16HETD, 4th years, Polytech Nancy, Université de Lorraine, France.
- Master: Martin Vassor, Foundation of Cybersecurity, 24.5HETD, 2A, Mines Nancy, Université de Lorraine, France.
- Master: Martin Vassor, Python for engineers, 40.5 HETD, 2A and 3A, Mines Nancy, Université de Lorraine, France.

10.2.2 Supervision

- PhD completed: Thomas Bagrel, Formalization and Implementation of Safe Destination Passing in Functional Programming Languages [27]. University of Lorraine. Supervised by Horatiu Cirstea, since April 2022, defended in November 2025.
- PhD in progress: Ghilain Bergeron, Generating distributed programs from formal specifications. University of Lorraine. Supervised by Horatiu Cirstea and Stephan Merz, since October 2023.
- PhD in progress: Alessio Coltellacci, Reconstructing SMT Proofs in Lambdapi. Supervised by Stephan Merz, since January 2023.
- PhD in progress: Sarah Dépernet, Model-checking security properties on Timed automata. Supervised by Engel Lefaucheux and Stephan Merz, since September 2024.
- PhD in progress: Florent Krasnopol, Automated Theorem Proving over the Reals for Reasoning on Quantum Circuits. Supervised by Julie Cailler, Sophie Tourret, and Stephan Merz, since September 2025.
- PhD in progress: Mohamed Amine Snoussi, Reengineering of an Industrial Communication Protocol. Supervised by Marie Duflot-Kremer, Engel Lefaucheux, and Stephan Merz, since May 2023.
- PhD in progress: Vincent Trélat, Higher-Order SMT Solving for Proof Obligations in Set Theory. University of Lorraine. Supervised by Stephan Merz and Sophie Tourret, since October 2023.
- Master internship in progress: Tiago Campos Ferreira, Integrating SMT-based abduction in the proof assistant Isabelle/HOL, since October 2025. Supervised by Sophie Tourret.
- Master internship: Lucas Villaume, Formalisation and verification of a train scheduler in TLA$^+$, Spring semester 2025. Supervised by Martin Vassor.
- Bachelor internship: Volkan Burakcin, Correctness proof of Raft, Spring semester 2025. Supervised by Stephan Merz.
- Bachelor internship: Baptiste Diedler, Degradation of stochastic systems, Spring semester 2025. Supervised by Marie Duflot-Kremer and Engel Lefaucheux.
- Bachelor internship: Bastien Pichet, Degradation of stochastic systems, Spring semester 2025. Supervised by Marie Duflot-Kremer and Engel Lefaucheux.

10.2.3 Juries

- Julie Cailler was a member of the jury for the French national high school teacher competitive exam in computer science (CAPES NSI).
- Marie Duflot-Kremer was a member of the board (secrétaire générale) of the jury for the French national high school teacher competitive exam in computer science (CAPES NSI).
- Stephan Merz was a reviewer and member of the PhD committees of theses at IPP Paris, University of Saclay, and University of Strasbourg.

10.2.4 Educational and pedagogical outreach

- Julie Cailler, Marie Duflot-Kremer, and Sophie Tourret gave a talk and a lab session on formal verification for a training program for teachers in France (PNF), Nancy.
- Marie Duflot-Kremer gave
  - two talks and activities on unplugged computer science for a training program for teachers in France (PNF), online,
  - one workshop at a *séminaire de pédagogie universitaire* on including unplugged activities in a computer science curriculum, Université de Lorraine, Nancy.
  - a talk about popularization activities at the annual day of LIP6 lab, Paris.
  - a talk about popularization activities at the new staff Inria seminar, Strasbourg.
- Stephan Merz gave a lecture (6 hours) on Specifying and Verifying Algorithms in TLA$^+$ at the VTSA summer school in Liège, Belgium.
- Sophie Tourret gave
  - a lecture at the SAT/SMT/AR summer school 2025 (St Andrews, Scotland), and
  - a talk at the CP/SAT Doctoral Program 2025 (Glasgow, Scotland).

10.3 Popularization

10.3.1 Specific official responsibilities in science outreach structures

- Julie Cailler and Marie Duflot-Kremer co-organized *Le stage des cigognes*, a one week experience of research in maths and computer science for high school girls.
- Marie Duflot-Kremer is the deputy vice-president for outreach activities in the supervisory council of SIF (*Société Informatique de France*).
- Thomas Sturm was involved in the organization of the scientific training of the German team for the International Olympiad in Informatics (IOI).

10.3.2 Participation in Live events

- Marie Duflot-Kremer:
  - *Journée décodeuses du numérique*, unplugged activities workshop, organized by CNRS, Paris,
  - *Journées filles et sciences*, unplugged activities workshop for secondary school girls, lycée Loritz, Nancy,
  - Mathematics week, presentation of outreach unplugged activities for teachers, rectorat de Nancy,
  - one day of training secondary school students and two live shows of *informagic* for 300 and 400 students, lycée Vauban, Luxembourg,
  - *Journées du matrimoine*, explaining research in formal verification, Jarville,
  - a *popularization tour* with 12 workshops ranging from kindergarden to secondary school audience over 4 days, Montereau-Fault-Yonne,
  - a talk for general audience on unplugged activities, Montereau-Fault-Yonne,
  - *Journée d'initiative citoyenne en faveur de l'égalité femmes-hommes*, a short talk on women in science, Vandœuvre-lès-Nancy,
  - two days of *fête de la science*, training and supervising first year students realizing unplugged computer science workshops for a general audience, Université de Lorraine, Vandœuvre-lès-Nancy,

10.3.3 Other activities related to science outreach

- Julie Cailler:
  - Chiche: 1 scientifique, 1 classe
  - MATh.en.JEANS
- Marie Duflot-Kremer:
  - *Chiche: 1 scientifique, 1 classe*, 4 talks to secondary school students, Fameck,
  - referent researcher of a group of secondary school kids doing research for the Math.en.JEANS 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] F. Kröger and S. Merz. *Temporal Logic and State Systems*. Texts in Theoretical Computer Science. Springer, 2008, p. 436. URL: http://hal.inria.fr/inria-00274806/en/.

[8] 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.

[9] 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.

[10] 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.

[11] 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

[12] R. Defourné. 'Encoding TLA+ proof obligations safely for SMT'. In: *Science of Computer Programming*. Selected Papers From the Rigorous State-Based Methods 9th International Conference(ABZ 2023) 239.103178 (Jan. 2025). DOI: 10.1016/J.SCICO.2024.103178. URL: https://inria.hal.science/hal-04701040.

International peer-reviewed conferences

[13] P. A. Abdulla, M. F. Atig, J. Cailler, C. Liang and P. Rümmer. 'When GNNs Met a Word Equations Solver: Learning to Rank Equations'. In: FroCoS 2025. Vol. 15979. Lecture Notes in Computer Science. Reykavik, Iceland: Springer Nature Switzerland, 15th Sept. 2025, pp. 327–345. DOI: 10.1007/978-3-032-04167-8_18. URL: https://hal.science/hal-05329165.

[14] T. Bagrel and A. Spiwack. 'Destination Calculus: A Linear λ-Calculus for Purely Functional Memory Writes'. In: *Proceedings of the ACM on Programming Languages, Volume 9, Issue OOPSLA1Article No.: 89, Pages 253 - 279*. OOPSLA 2025 - ACM Conference on Object Oriented Programming Systems Languages and Applications. Vol. 9. OOPSLA1. Singapore, Singapore: ACM, 9th Apr. 2025, pp. 253–279. DOI: 10.1145/3720423. URL: https://hal.science/hal-05455806.

[15] A. Barwell, P. Hou, M. Vassor and N. Yoshida. 'Encoding Choice and Replication in roll-π'. In: *International Conference on Reversible Computation*. International Conference on Reversible Computation. Vol. 15716. Lecture Notes in Computer Science. Odense (Danemark), Denmark, 22nd June 2025, pp. 27–36. DOI: 10.1007/978-3-031-97063-4_3. URL: https://hal.science/hal-05332890.

[16] G. Bergeron, F. Krasnopol and S. Tourret. 'Formalizing Splitting in Isabelle/HOL'. In: ITP 2025 - 16th International Conference on Interactive Theorem Proving. Reykjavik, Iceland: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. DOI: 10.4230/LIPIcs.ITP.2025.22. URL: https://inria.hal.science/hal-05329260.

[17] N. Bertrand, L. Hélouët, E. Lefaucheux and L. Paparazzo. 'Reachability in multi-agent transfer systems'. In: VMCAI 2026 - 27th International Conference on Verification, Model Checking, and Abstract Interpretation. LNCS. Rennes, France, 2026. URL: https://inria.hal.science/hal-05447890.

[18] A. Coltellacci and S. Merz. 'Checking Linear Integer Arithmetic Proofs in Lambdapi'. In: FroCoS 2025 - 15th International Symposium Frontiers of Combining Systems. Vol. 15979. Lecture Notes in Computer Science. Reykjavik, Iceland: Springer Nature Switzerland, 2025, pp. 367–385. DOI: 10.1007/978-3-032-04167-8. URL: https://inria.hal.science/hal-05328609.

[19] I. Dramnesc, E. Abraham, N. Fachantidis, T. Jebelean, G. Kusper and S. Stratulat. 'AIROBO: A EUROPEAN PROJECT FOR SOFTWARE ENGINEERS'. In: EDULEARN 2025 - 17th International Conference on Education and New Learning Technologies. Vol. EDULEARN25 Proceedings. Palma De Majorque, Spain, 30th June 2025, pp. 1221–1228. DOI: 10.21125/edulearn.2025.0394. URL: https://inria.hal.science/hal-05447018.

[20] I. Dramnesc, E. Abraham, N. Fachantidis, T. Jebelean, G. Kusper and S. Stratulat. 'AiRobo: A EU Project on Artificial Intelligence and Robotics for Higher Education'. In: ED-MEDIA 2025 - EdMedia + Innovate Learning. Vol. ED-MEDIA 2025 - EdMedia + Innovate Learning. Barcelona, Spain, 19th May 2025, pp. 1198–1203. URL: https://inria.hal.science/hal-05447056.

[21] N. Faroß and T. Sturm. 'On the Number of Real Types of Univariate Polynomials'. In: *ISSAC '25: Proceedings of the 2025 International Symposium on Symbolic and Algebraic Computation*. ISSAC 2025 - 50th International Symposium on Symbolic and Algebraic Computation. Guanajuato, Mexico: ACM, 10th Nov. 2025, pp. 62–69. DOI: 10.1145/3747199.3747547. URL: https://hal.science/hal-05406461.

[22] S. Guilloud, J. Cailler, A. Poiroux, Y. Herklotz, T. Bourgeat and V. Kunčak. 'Interoperability of Proof Systems with SC-TPTP'. In: CADE 30 - 30th International Conference on Automated Deduction. Vol. 15943. Lecture Notes in Computer Science. Stuttgart, Germany: Springer Nature Switzerland, 30th July 2025, pp. 325–340. DOI: 10.1007/978-3-031-99984-0_18. URL: https://hal.science/hal-05329188.

[23] ***Best Paper***
V. Trélat. 'Safely Encoding B Proof Obligations in SMT-LIB'. In: *Lecture Notes in Computer Science*. Rigorous State-Based Methods – 11th International Conference, ABZ 2025. Vol. 15728. Düsseldorf, Germany: Springer, 2nd June 2025, pp. 52–69. DOI: 10.1007/978-3-031-94533-5_4. URL: https://hal.science/hal-05329157.

Conferences without proceedings

[24] G. Bergeron, H. Cirstea and S. Merz. 'Towards a verified compiler for Distributed PlusCal'. In: 11th International Workshop on Rewriting Techniques for Program Transformations and Evaluation. Birmingham, United Kingdom, July 2025. URL: https://hal.science/hal-05329156.

[25] S. Merz. 'Invariant Synthesis: Decidable Fragments to the Rescue'. In: First-Order Reasoning, Below and Beyond: Workshop in Honor of Christoph Weidenbach's 60th Birthday. Stuttgart, Germany, Aug. 2025. URL: https://hal.science/hal-05329139.

Scientific book chapters

[26] I. Lanese, C. A. Mezzina and M. Vassor. 'Bounded Reversibility in HOπ'. In: *Components Operationally: Reversibility and System Engineering. Essays Dedicated to Jean-Bernard Stefani on the Occasion of His 65th Birthday*. Vol. 16065. Lecture Notes in Computer Science. Springer Nature Switzerland, 19th Oct. 2025, pp. 24–45. DOI: 10.1007/978-3-031-99717-4_2. URL: https://hal.science/hal-05332969.

Doctoral dissertations and habilitation theses

[27] T. Bagrel. 'Formalization and Implementation of Safe Destination Passing in Pure Functional Programming Settings'. LORIA (Université de Lorraine, CNRS, INRIA), 14th Nov. 2025. URL: https://hal.science/tel-05455981.

Reports & preprints

[28] N. Bertrand, L. Hélouët, E. Lefaucheux and L. Paparazzo. *Reachability in multi-agent transfer systems (Extended Version)*. 15th Nov. 2025. URL: https://inria.hal.science/hal-05366409.

[29] N. Faroß and T. Sturm. *On the Number of Real Types of Univariate Polynomials*. 2025. DOI: 10.48550/arXiv.2502.04914. URL: https://hal.science/hal-05207784.

Scientific popularization

[30] D. Méry. *Teaching formal techniques with Event-B and Rodin ⋆*. 27th May 2025. URL: https://inria.hal.science/hal-05436671.

11.3 Cited publications

[31] 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.

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

[33] 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.

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

[35] N. Bertrand, S. Haddad and E. Lefaucheux. 'A tale of two diagnoses in probabilistic systems'. In: *Information and Computation* 269 (2019), p. 104441.

[36] P. Blackburn, M. d. Rijke and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001. DOI: 10.1017/CBO9781107050884.

[37] B. Boigelot, P. Fontaine and B. Vergain. 'Non-emptiness Test for Automata over Words Indexed by the Reals and Rationals'. In: *Lecture Notes in Computer Science*. Vol. 15015. Lecture Notes in Computer Science. Akita, Japan: Springer Nature, Sept. 2024, pp. 94–108. DOI: 10.1007/978-3-031-71112-1_7. URL: https://inria.hal.science/hal-04896026.

[38] G. Bonfante, J.-P. Auzelle, R. Badonnel, S. Duval, S. Gégout, M. Gilson, C. Joliot, É. Koessler, A. Knauf, N. Krommenacker, S. Schmitt and P. Veutin. 'CyberHumanumEst, une guerre cyber autour des Riverchelles'. In: *Rendez-Vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information (RESSI) 2024*. Centre de Recherche en Informatique, Signal et Automatique de Lille (Cristal, Université de Lile). Eppe-Sauvage, France, May 2024. URL: https://hal.science/hal-04988987.

[39] A. Bouhoula and M. Hermann. 'Primal Grammars Driven Automated Induction'. In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. Jeju, South Korea, Aug. 2023, pp. 3259–3269. DOI: 10.24963/ijcai.2024/361. URL: https://hal.science/hal-04790992.

[40] D. Déharbe, P. Fontaine, Y. Guyot and L. Voisin. 'SMT solvers for Rodin'. In: *ABZ - Third International Conference on Abstract State Machines, Alloy, B, VDM, and Z - 2012*. Ed. by J. Derrick, J. A. Fitzgerald, S. Gnesi, S. Khurshid, M. Leuschel, S. Reeves and E. Riccobene. Vol. 7316. Lecture Notes in Computer Science. Pisa, Italy: Springer, 2012, pp. 194–207.

[41] I. Drămnesc, E. Ábrahám, T. Jebelean, N. Fachantidis, G. Kusper and S. Stratulat. 'A European Project on AI-based Robotics'. In: *TALE2024 (International Conference on Teaching, Assessment and Learning for Engineering)*. Bengalore, India, Dec. 2024. URL: https://inria.hal.science/hal-04814646.

[42] G. Ebner, J. Blanchette and S. Tourret. 'Unifying Splitting'. In: *J. Autom. Reason.* 67.2 (2023), p. 16.

[43] M. Košta. 'New Concepts for Real Quantifier Elimination by Virtual Substitution'. Doctoral Dissertation. Germany: Saarland University, 2016. DOI: 10.22028/D291-26679.

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

[45] 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.

[46] 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.

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

[48] D. Ongaro and J. K. Ousterhout. 'In Search of an Understandable Consensus Algorithm'. In: *USENIX Annual Technical Conference 2014*. Ed. by G. Gibson and N. Zeldovich. Philadelphia, PA: Usenix Association, 2014, pp. 305–319.

[49] C. Rothgang, F. Rabe and C. Benzmüller. 'Theorem Proving in Dependently-Typed Higher-Order Logic'. In: *CADE*. Vol. 14132. Lecture Notes in Computer Science. Springer, 2023, pp. 438–455.

[50] 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.

[51] S. Tourret and J. Blanchette. ‘A modular Isabelle framework for verifying saturation provers’. In: *CPP*. ACM, 2021, pp. 224–237.