RESEARCH CENTRE

Nancy - Grand Est

IN PARTNERSHIP WITH:

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

2020 ACTIVITY REPORT

Project-Team VERIDIS

Modeling and Verification of Distributed Algorithms and Systems

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

DOMAIN

Algorithmics, Programming, Software and Architecture

THEME

Proofs and Verification

Contents

Project-Team VERIDIS

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

Keywords

Computer sciences and digital sciences

A2.1.7. – Distributed programming

A2.1.11. – Proof languages

A2.4. – Formal method for verification, reliability, certification

A2.4.1. – Analysis

A2.4.2. – Model-checking

A2.4.3. – Proofs

A2.5. – Software engineering

A7.2. – Logic in Computer Science

A8.4. – Computer Algebra

Other research topics and application domains

B6.1. – Software industry

B6.1.1. – Software engineering

B6.3.2. – Network protocols

B6.6. – Embedded systems

1 Team members, visitors, external collaborators

Research Scientists

- Stephan Merz [Team leader, Inria, Senior Researcher, HDR]
- Thomas Sturm [CNRS, Senior Researcher, HDR]
- Uwe Waldmann [Max Planck Society]
- Christoph Weidenbach [Max Planck Society, HDR]

Faculty Members

- Étienne André [Univ de Lorraine, Professor, HDR]
- Marie Duflot-Kremer [Univ de Lorraine, Associate Professor]
- Dominique Méry [Univ de Lorraine, Professor, HDR]
- Sorin Stratulat [Univ de Lorraine, Associate Professor]

Post-Doctoral Fellows

- Martin Bromberger [Max Planck Society]
- Zheng Cheng [Univ de Lorraine]
- Yann Duplouy [Inria, until Oct 2020]
- Mathieu Montin [Univ de Lorraine, from Nov 2020]
- Hamid Rahkooy [CNRS, until Jul 2020, Max Planck Society from Aug 2020]
- Sophie Tourret [Max Planck Society]
- Marco Voigt [Max Planck Society, until Feb 2020]

PhD Students

- Antoine Defourné [Inria]
- Margaux Duroeulx [Univ de Lorraine, until Apr 2020]
- Daniel El Ouraoui [Inria]
- Mathias Fleury [Max Planck Society, until Jan 2020]
- Alexis Grall [Univ de Lorraine, until Sep 2020]
- Fajar Haifani [Max Planck Society]
- Pierre Lermusiaux [Univ de Lorraine]
- Hans Jorg Schurr [Inria]

Interns and Apprentices

- Heba Al Kayed [Inria, from Mar 2020 until Jul 2020]
- Adele Barbier [Univ de Lorraine, from Apr 2020 until Jun 2020]
- Amelie Ferstler [Univ de Lorraine, from May 2020 until Jun 2020]
- Aleksander Kryukov [Univ de Lorraine, from Mar 2020 until Jul 2020]
- Raphael Le Bihan [Inria, from Jun 2020 until Jul 2020]
- Mathieu Tabary [Univ de Lorraine, from Apr 2020 until Jun 2020]
- Cristian Vargas Montero [Univ de Lorraine, from Mar 2020 until Jul 2020]

Administrative Assistants

- Sophie Drouot [Inria]
- Sylvie Hilbert [CNRS]

Visiting Scientist

- Masaki Waga [Univ de Lorraine, Jan 2020]

External Collaborators

- Jasmin Christian Blanchette [Université libre d'Amsterdam - Pays-Bas]
- Pascal Fontaine [Université de Liège, HDR]
- Igor Konnov [Informal Systems, on leave from Inria]

2 Overall objectives

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

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

Within this context, we work on techniques for *automated theorem proving* for expressive languages based on first-order logic, with support for theories (fragments of arithmetic, set theory etc.) that are relevant for specifying algorithms and systems. Ideally, systems and their properties would be specified in high-level, expressive languages, errors in specifications would be discovered automatically, and finally, full verification could also be performed completely automatically. Due to the fundamental undecidability of the problem, this cannot be achieved in general. Nevertheless, we have observed important advances in automated deduction in recent years, to which we have contributed. These advances suggest that a substantially higher degree of automation can be achieved over what is available in today's tools supporting deductive verification. Our techniques are developed within SMT (satisfiability modulo theories) solving and superposition reasoning, the two main frameworks of contemporary automated reasoning that have complementary strengths and weaknesses, and we are interested in making them

converge when appropriate. Techniques developed within the symbolic computation domain, such as algorithms for quantifier elimination for appropriate theories, are also relevant, and we are working on integrating them into our portfolio of techniques. In order to handle expressive input languages, we are working on techniques that encompass tractable fragments of higher-order logic, for example for specifying inductive or co-inductive data types, for automating proofs by induction, or for handling collections defined through a characteristic predicate.

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

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

Beyond formal verification, we pursue applications of some of the symbolic techniques that we are developing in other domains. We have observed encouraging success in using techniques of symbolic computation for the qualitative analysis of biological and chemical 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 models, which are typically used for Influenza A or Covid-19. This work is being pursued within a large-scale interdisciplinary collaboration. It aims for our work grounded in verification to have an impact on the sciences, beyond engineering, which will feed back into our core formal methods community.

3 Research program

3.1 Automated and Interactive Theorem Proving

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

VeriDis members from Saarbrücken are developing the SPASS [10] workbench. It currently consists of one of the leading automated theorem provers for first-order logic based on the superposition calculus [62] and a theory solver for linear arithmetic [2].

In a complementary approach to automated deduction, VeriDis members from Nancy work on techniques for integrating reasoners for specific theories. They develop veriT [1], an SMT [1] solver that combines decision procedures for different fragments of first-order logic. The veriT solver is designed to produce detailed proofs; this makes it particularly suitable as a component of a robust cooperation of deduction tools.

Finally, VeriDis members design effective quantifier elimination methods and decision procedures for algebraic theories, supported by their efficient implementation in the Redlog system [5].

An important objective of this line of work is the integration of theories in automated deduction. Typical theories of interest, including fragments of arithmetic, are difficult or impossible to express in first-order logic. We therefore explore efficient, modular techniques for integrating semantic and syntactic reasoning methods, develop novel combination results and techniques for quantifier instantiation. These problems are addressed from both sides, i.e. by embedding decision procedures into the superposition framework or by allowing an SMT solver to accept axiomatizations for plug-in theories. We also develop specific decision procedures for theories such as non-linear real arithmetic that are important when reasoning about certain classes of (e.g., real-time) systems but that also have interesting applications beyond verification.

We rely on interactive theorem provers for reasoning about specifications at a high level of abstraction when fully automatic verification is not (yet) feasible. An interactive proof platform should help verification engineers lay out the proof structure at a sufficiently high level of abstraction; powerful automatic plug-ins should then discharge the resulting proof steps. Members of VeriDis have ample experience in the specification and subsequent machine-assisted, interactive verification of algorithms. In particular, we participate in a project at the joint Microsoft Research-Inria Centre on the development of methods and tools for the formal proof of TLA$^+$ [71] specifications. Our prover relies on a declarative proof language, and calls upon several automatic backends [4]. Trust in the correctness of the overall proof can be ensured when the backends provide justifications that can be checked by the trusted kernel of a proof assistant. During the development of a proof, most obligations that are passed to the prover actually fail – for example, because necessary information is not present in the context or because the invariant is too weak, and we are interested in explaining failed proof attempts to the user, in particular through the construction of counter-models.

Members of VeriDis formalize a framework in the proof assistant Isabelle/HOL for representing the correctness and completeness of automated theorem provers. This work encompasses proof calculi such as ordered resolution or superposition, as well as concrete prover architectures such as Otter or DISCOUNT loops.

3.2 Formal Methods for Developing and Analyzing Algorithms and Systems

Theorem provers are not used in isolation, but they support the application of sound methodologies for modeling and verifying systems. In this respect, members of VeriDis have gained expertise and recognition in making contributions to formal methods for concurrent and distributed algorithms and systems [3, 8], and in applying them to concrete use cases. In particular, the concept of *refinement* [60, 63, 76] 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.

[1] Satisfiability Modulo Theories [65]

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

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. Yesterday, simple models of 2 to 5 ordinary differential equations were used to illustrate fundamental ideas such as multi-stationarity or biological rhythms sustained by biological cells. Even such simple models required involved analysis, justifying 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 [74] currently contains thousands of hand-built models of up to several hundreds of variables. Much quicker analysis is also needed. In precision medicine one wants to be able to move from hypotheses to their verification in minutes. Numerical analysis of large models often fails to meet these desiderata because it either requires an exhaustive scan of the parameter space or the identification of the numerical parameters from data. Both tasks are impossible 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. An alternative to numerical methods is the use of symbolic methods, which can be used to compute bifurcation and phase diagrams of biological models, which are very useful tools for decision.

It turns out that, from a computational point of view, many problems in that field combine algebraic questions with combinatorics, for which we have found that SMT solving over various theories is an excellent tool. When generalizing to parametric variants of the considered problems, first-order quantifier elimination methods over complex numbers, real numbers, or integers enter the stage.

Members of VeriDis have carried out considerable research on applications of symbolic techniques in systems biology, especially since the launch of the interdisciplinary SYMBIONT project in 2018 (section 9.3).

4 Application domains

Distributed algorithms and protocols are found at all levels of computing infrastructure, from many-core processors and systems-on-chip to wide-area networks. We are particularly interested in the verification of algorithms that are developed for supporting novel computing paradigms, including ad-hoc networks that underly mobile and low-power computing or overlay networks, peer-to-peer networks that provide services for telecommunication, or cloud computing services. Computing infrastructure must be highly available and is ideally invisible to the end user, therefore correctness is crucial. One should note that standard problems of distributed computing such as consensus, group membership or leader election have to be reformulated for the dynamic context of these modern systems. We are not ourselves experts in the design of distributed algorithms, but we work together with domain experts on designing formal models of these protocols, and on verifying their properties. These cooperations help us focus on concrete algorithms and ensure that our work is relevant to the distributed algorithm community.

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

numerical analysis techniques and are validated against collections of models from computational biology.

We have also targeted applications with respect to cybersecurity, notably within the ANR-NRF ProMiS project. In [23], we proposed a new definition of non-interference for parametric timed automata, an extension of finite-state automata with clocks and timing parameters (unknown constants) [61]. We proposed a method to quantify the minimal frequency of attack actions for their attack to succeed (or, put differently, the maximal frequency of internal actions so that there is no leak to an external observer). We inferred values of the minimal time between two consecutive actions of the attacker, so that (s)he disturbs the discrete behavior (set of reachable locations). We also synthesized valuations for the timing constants of the automaton (seen as parameters) guaranteeing non-interference.

5 Highlights of the year

5.1 Awards

- The paper [33] , prepared in cooperation with colleagues from the University of Iowa and Stanford University, received a best paper award at IJCAR 2020. We proved that theories of algebraic data types such as lists or trees, as present in the SMT-LIB language, exhibit interesting properties that enable to integrate them nicely in a combination of decision procedures in SMT reasoning.
- The prover Zipperposition, developed by Blanchette, Tourret, Waldmann, and their colleagues in the context of the Matryoshka project (cf. section 9.2.1), finished in first place in the higher-order proving division of the 2020 edition of the CADE ATP System Competition[2], over 20 percentage points ahead of the nearest competitor. This breakthrough was possible thanks to the theoretical work done by team members, notably the design of the superposition calculus with lambda-abstractions mentioned in last year's report. Zipperposition was initially developed by Simon Cruanes, who was a senior engineer in the VeriDis team from 2015 to 2017.
- Martin Bromberger won the EAPLS dissertation award[3]. Citing the summary of the jury's findings
  - The problems addressed in this thesis are highly relevant and extremely challenging.
  - The excellent results obtained in this thesis advance the state of the art of SMT solving and theorem proving in an outstanding way.
  - The thesis presents breakthrough-like achievements in a well-established and researched field with an immediate impact and advancement of the field.
  - The presented procedures offer significant improvements over the state of the art that has been adopted in professional SMT provers.

6 New software and platforms

6.1 New software

6.1.1 Redlog

Name: Reduce Logic System

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

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

Redlog generally works with interpreted first-order logic in contrast to free first-order logic. Each first-order formula in Redlog must exclusively contain atoms from one particular Redlog-supported

[2] http://www.tptp.org/CASC/J10/WWWFiles/DivisionSummary1.html
[3] https://eapls.org/items/3622/

theory, which corresponds to a choice of admissible functions and relations with fixed semantics. Redlog-supported theories include Nonlinear Real Arithmetic (Real Closed Fields), Presburger Arithmetic, Parametric QSAT, and many more.

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

During 2020 we implemented and launched an integrated documentation and online help system, which is available interactively in the software itself as well as online on the Redlog website via an interactive GUI. The website also features a RESTful interface to the documentation. In that course, the website has undergone a total redesign, now using state-of-the-art web technologies and frameworks such as Angular and Bootstrap.

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

Contact: Thomas Sturm

Participant: Thomas Sturm

6.1.2 SPASS-SATT

Name: SPASS-SATT

Keywords: Automated deduction, Decision

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

We further investigated the use of redundancy elimination in SAT solving and underlying implementation techniques. Our aim is a new approach to SAT solving that needs fewer conflicts (on average) *and* is faster than the current state-of-the art solvers. Furthermore, we have developed a new calculus and first prototypical implementation of a SAT solver with mixed OR/XOR clauses and are currently adapting our algorithms to support SUPERLOG reasoning.

News of the Year: SPASS-SATT participated in the SMT competitions 2018-2019 in the quantifier free integer and rational linear arithmetic categories and won one category each year, respectively. We have stopped entering the competition, but the competition designers run previous years winners. Our 2019 version scored second in the rational linear arithmetic category in 2020.

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

Contacts: Martin Bromberger, Christoph Weidenbach

Participants: Martin Bromberger, Mathias Fleury, Christoph Weidenbach

6.1.3 veriT

Keywords: Automated deduction, Formula solving, Verification

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

News of the Year: Efforts in 2020 have been focused on quantifier handling, higher logic, and better proof production.

The veriT solver participated in the SMT competition SMT-COMP 2020 with good results. In particular, our fast version (tuned for 24s) was among the fastest (besides portfolio approaches) for several logics, in the 24s category.

We target applications where validation of formulas is crucial, such as the validation of TLA$^+$ and B specifications, and work together with the developers of the respective verification platforms to make veriT even more useful in practice. The solver is available as a plugin for the *Rodin* platform, and it is integrated within *Atelier B*.

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

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

Authors: David Déharbe, Pascal Fontaine, Diego Caminha B. De Oliveira, Haniel Barbosa, Thomas Bouton, Daniel El Ouraoui, Hans-Jörg Schurr

Contact: Pascal Fontaine

Participants: Haniel Barbosa, Daniel El Ouraoui, Pascal Fontaine, Hans-Jörg Schurr

Partner: Université de Lorraine

6.1.4 TLAPS

Name: TLA+ proof system

Keyword: Proof assistant

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

News of the Year: In 2020, we published a minor release, fixing some issues notably for the SMT back-end. Substantial work was devoted to supporting liveness reasoning, in particular proofs about the ENABLED and action composition constructions of TLA+. We also prepared support for current versions of the Isabelle back-end prover.

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

Contacts: Stephan Merz, Damien Doligez

Participants: Damien Doligez, Stephan Merz, Ioannis Filippidis

Partner: Microsoft

6.1.5 Apalache

Name: Abstraction-based Parameterized TLA+ Checker

Keyword: Model Checker

Scientific Description: Apalache is a symbolic model checker that works under the following assumptions:

(1) As in TLC, all specification parameters are fixed and finite, e.g., the system is initialized integers, finite sets, and functions of finite domains and co-domains. (2) As in TLC, all data structures

evaluated during an execution are finite, e.g., a system specification cannot operate on the set of all integers. (3) Only finite executions up to a given bound are analysed.

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

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

Functional Description: Version 0.5.0 implements a symbolic bounded model checker for TLA$^+$ that runs under the same assumptions as the explicit-state model checker TLC. It checks whether a TLA$^+$ specification satisfies an invariant candidate by checking satisfiability of an SMT formula that encodes: (1) an execution of bounded length, and (2) preservation of the invariant candidate in every state of the execution. Our tool is still in the experimental phase, due to a number of challenges posed by the semantics of TLA$^+$ to SMT solvers.

Release Contributions: Version 0.10.0 introduces multiple features in comparison to version 0.5.0:

- the new type checker Snowcat that works independently of the model checker's pipeline - the new assignment finder that delivers an improved user experience - an improved build system and continuous integration development - support for TLC configuration files - an improved SMT encoding for set cardinalities - support for recursive operators and recursive functions - support for advanced language features such as local instances, local operators, lambdas, etc. - support for offline and online SMT solvers

News of the Year: In 2020, the tool was mainly developed at TU Wien and Informal Systems. The verification team of Informal Systems applied Apalache for model checking of several blockchain protocols in TLA+. Importantly, Apalache was the only tool that could cope with those specifications, as these protocols operate under Byzantine faults and time constraints.

We have improved the overall tool stability and accounted for the feedback that was provided to us by the tool users at Informal Systems. Notably, we have found that the type checker and assignment finder were the main bottlenecks from the users' point of view. To this end, we have implemented a new type checker that expects type annotations in a simple format. The new type checker is able to find types of many operators by constraint solving.

Moreover, we have also developed a completely automatic type inference tool. However, this tool is still in the prototype phase, and it needs a good amount of work to become a mature tool.

URL: https://github.com/informalsystems/apalache

Publications: hal-01899719v1, hal-01871131v1, hal-02280888v1

Contact: Igor Konnov

Partners: Technische Universität Wien, Informal Systems

6.1.6 IMITATOR

Name: IMITATOR

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

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

News of the Year: New algorithm for NDFS-based cycle synthesis (by Laure Petrucci and Jaco Van de Pol). Extension of the syntax: if-then-else conditions allowed in updated, #include allowed for submodel inclusion. New applications to cybersecurity.

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

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

Contact: Etienne Andre

Participants: Etienne Andre, Jaime Eduardo Arias Almeida

Partner: Loria

7 New results

7.1 Automated and Interactive Theorem Proving

Participants Jasmin Christian Blanchette, Martin Bromberger, Antoine Defourné, Daniel El Ouraoui, Mathias Fleury, Pascal Fontaine, Fajar Haifani, Raphaël Le Bihan, Stephan Merz, Hans-Jörg Schurr, Sorin Stratulat, Sophie Tourret, Marco Voigt, Uwe Waldmann, Christoph Weidenbach.

7.1.1 Contributions to SMT Techniques

Combination of Satisfiability Procedures. A satisfiability problem is often expressed in a combination of theories, and a natural approach consists in solving the problem by combining the satisfiability procedures available for the component theories. This is the purpose of the combination method introduced by Nelson and Oppen. However, in its initial presentation, the Nelson-Oppen combination method requires the theories to be signature-disjoint and stably infinite. The design of a generic combination method for non-disjoint unions of theories is difficult, but it is worth exploring simple non-disjoint combinations that appear frequently in verification. An example is the case of shared sets, where sets are represented by unary predicates. Another example is the case of bridging functions between data structures and a target theory (e.g., a fragment of arithmetic).

In 2015, we defined a sound and complete combination procedure *à la* Nelson-Oppen for the theory of absolutely free data structures (including lists and trees) connected to another theory via bridging functions [67]. This combination procedure has also been refined for standard interpretations. The resulting theory has a nice politeness property, enabling combinations with arbitrary decidable theories of elements. We also investigated other theories [68] amenable to similar combinations: this class includes the theory of equality, the theory of absolutely free data structures, and all the theories in between.

In 2018 and 2019, we have been improving the framework and unified both results. This was published in the Journal of Automated Reasoning in 2020 [17].

The above works pave the way for combinations involving the theory of algebraic datatypes as found in the SMT-LIB. Together with colleagues in Iowa and Stanford, this was published at IJCAR 2020 [33]. This article received a best paper award.

Quantifier Handling in SMT. SMT solvers generally rely on various instantiation techniques for handling quantifiers. We built a unifying framework encompassing quantified formulas with equality and uninterpreted functions, such that the major instantiation techniques in SMT solving can be cast in that framework. It is based on the problem of E-ground (dis)unification, a variation of the classic Rigid E-unification problem. We introduced a sound and complete calculus to solve this problem in practice: Congruence Closure with Free Variables (CCFV). Experimental evaluations of implementations of CCFV demonstrate notable improvements in the state-of-the-art solver CVC4 and make the solver veriT competitive with state-of-the-art solvers for several benchmark libraries, in particular those originating in verification problems.

In 2019 and 2020, we investigated machine learning techniques for predicting the usefulness of an instance in order to decrease the number of instances passed to the SMT solver. For this, we proposed a meaningful way to characterize the state of an SMT solver, to collect instantiation learning data, and to integrate a predictor in the core of a state-of-the-art SMT solver. This ultimately leads to more efficient SMT solving for quantified problems.

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

This result was published at CADE 2019 [64]. Work in 2020 focused on extending the CCFV algorithm to higher-order logic [41]: the first-order algorithm is not directly usable since it strongly relies on the fact that functions are fully applied, and no variable can appear in a function place. It is also necessary to find a radically different approach. Our approach is to work on an encoding of the CCFV higher-order problem into SAT.

Proofs for SMT. We have previously developed a framework for processing formulas in automatic theorem provers, with generation of detailed proofs that can be checked by external tools, including skeptical proof assistants. The main components are a generic contextual recursion algorithm and an extensible set of inference rules. Clausification, skolemization, theory-specific simplifications, and expansion of 'let' expressions are instances of this framework. With suitable data structures, proof generation adds only a linear-time overhead, and proofs can be checked in linear time. We implemented the approach in the SMT solver veriT. This allowed us to dramatically simplify the code base while increasing the number of problems for which detailed proofs can be produced. In 2019, the format of proof output was further improved, while also improving the reconstruction procedure in the proof assistant Isabelle/HOL. This allowed the tactic using SMT with proofs to be regularly suggested by Sledgehammer as the fastest method to automatically solve proof goals. This was the subject of a workshop publication in 2019 [69]. In 2020, we have made steady progress on this front, and thanks to this progress, the veriT solver has been integrated into Isabelle with full support of reconstruction for veriT proof. This lead to improvements in the Sledgehammer facility to automatically discharge Isabelle proofs.

Theory Solving for Quantifier-Free Linear Integer Arithmetic. In [16] we consolidate our research in effective methods for the existential theory of Presburger Arithmetic over the past years in a journal article. We consider feasibility of linear integer problems in the context of verification systems such as SMT solvers or theorem provers. Although satisfiability of linear integer problems is decidable, many state-of-the-art implementations neglect termination in favor of efficiency. We present the calculus CutSat++ that is sound, terminating, complete, and leaves enough space for model assumptions and simplification rules in order to be efficient in practice. CutSat++ combines model-driven reasoning and quantifier elimination to the feasibility of linear integer problems.

7.1.2 Certification of automated reasoning techniques

A formal framework for automated reasoning. We are part of a group developing a framework for formal refutational completeness proofs of abstract provers that implement automated reasoning calculi, such as CDCL (Conflict Driven Clause Learning), ordered resolution, or superposition.

For CDCL we have been able to derive the state-of-the-art algorithms from a simple, verified core, inheriting its properties via instantiation. Then we could further refine this setting up to the generation of executable code that performs surprisingly well compared to hand-coded CDCL implementations [50].

The framework relies on modular extensions of lifted redundancy criteria that underlie the deletion of subsumed formulas. In presentations of proof calculi, this aspect is usually only discussed informally. Our framework allows us to extend redundancy criteria so that they cover subsumption, and also to model entire prover architectures in such a way that the static refutational completeness of a calculus immediately implies the dynamic refutational completeness of a prover implementing the calculus, for instance within an Otter or DISCOUNT loop. Our framework is mechanized in Isabelle/HOL. This research was presented at IJCAR 2020 [35].

Certifying Cyclic Induction Reasoning for First-order Logic with Admissible Predicates. The Coq proof assistant is powerful enough to reproduce cyclic reasoning for first-order logic with inductive definitions (FOL_{ID}) in terms of cyclic proofs. We identify a class of Coq-certifiable cyclic proofs convertible to a set of Coq proofs relying on normalized well-founded explicit induction. These proofs start with a unique explicit induction step whose induction schema is derived from the definition of new admissible predicates. The admissibility property, as well as the rest of the proofs, can be deduced from the input proof. The conversion procedure does not backtrack and no extra reconstruction proof techniques are needed. In practice, it has been used to certify FOL_{ID} CYCLIST proofs, including a proof for the 2-Hydra problem, and non-trivial cyclic SPIKE proofs of conjectures about conditional specifications.

7.1.3 Automated reasoning for specific logics

Signature-based abduction. Abduction is the process of explaining new observations using background knowledge. It is central to knowledge discovery and knowledge processing and has been intensely studied in various domains such as artificial intelligence, philosophy and logic.

Signature-based abduction aims at building hypotheses over a specified set of names, the signature, that explain an observation relative to some background knowledge. This type of abduction is useful for tasks such as diagnosis, where the vocabulary used for observed symptoms differs from the vocabulary expected to explain those symptoms. In the description logic literature, abduction has received little attention, despite being recognised as important for ontology repair, query update and matchmaking.

S. Tourret, together with P. Koopmann, W. Del-Pinto and R. Schmidt, presented the first complete method solving signature-based abduction for observations expressed in the expressive description logic $\mathcal{ALC}$ [30]. The method is guaranteed to compute a finite and complete set of hypotheses, and is evaluated on a set of realistic knowledge bases.

In joint work with P. Koopmann, we are currently investigating an alternative approach to abduction for description logics based on a translation to first-order logic and back. This work is motivated by the recent development of efficient tools for abductive reasoning in first-order logic.

Relevance of clauses for resolution. In joint work with P. Koopmann [39], we define a notion of relevance of a clause for proving a particular entailment by the resolution calculus. We think that our notion of relevance is useful for explaining why an entailment holds. A clause is relevant if there is no proof of the entailment without it. It is semi-relevant if there is a proof of the entailment using it. It is irrelevant if it is not needed in any proof. By using well-known translations of description logics to first-order clause logic, we show that all three notions of relevance are decidable for a number of description logics, including $\mathcal{EL}$ and $\mathcal{ALC}$. We provide effective tests for (semi-)relevance. The (semi-)relevance of a DL axiom is defined with respect to the (semi-)relevance of the respective clauses resulting from the translation.

This notion of semi-relevance is particularly interesting and can be detected using SOS-resolution derivations. We are currently working on a generalized proof of the SOS strategy for resolution to validate the theory behind the tests for semi-relevance in first-order logic.

Derivation reduction for SLD resolution. Inductive Logic Programming (ILP) is a form of machine learning that induces hypotheses from examples and background knowledge. Many forms of ILP use second-order Horn clauses as templates, also denoted as meta-rules, to learn logic programs, and several

of them rely on SLD resolution to produce new candidate solutions. Determining which meta-rules to use for a given learning task is a major open problem in ILP and most approaches use clauses provided by the designers of the systems without any theoretical justifications.

In joint work with A. Cropper we formalized the derivation reduction problem for SLD resolution, the undecidable problem of finding a finite subset of a set of clauses from which the whole set can be derived using SLD resolution. We studied the reducibility of various fragments of second-order Horn logic that are relevant in ILP and extended our results to standard resolution. We also conducted an empirical study of the effects of using reduced sets of such metarules on the overall learning accuracy and time, which shows a substantial improvement over the state of the art, in addition to the theoretical guarantees offered.

Towards Improved Encodings of TLA$^+$ Proof Obligations. We reconsider the encoding of proof obligations that arise in proofs about TLA$^+$ specifications in multi-sorted first-order logic. In his PhD thesis, Antoine Defourné studies correctness criteria for assigning types to TLA$^+$ expressions, based on embeddings between models for different logics. The objective is to delineate what type assignments are sound when translating from the untyped TLA$^+$ language into the multi-sorted logics underlying typical automated reasoning engines.

He also implemented a new back-end reasoner based on Zipperposition for handling proof obligations that involve features of higher-order logic such as function or predicate variables that may appear in TLA$^+$ proof sequents. The design of this back-end was presented at JFLA 2020 [38], and a working prototype at the TLA$^+$ Community Meeting [37].

During his undergraduate internship, Raphaël Le Bihan revisited the coalescing technique used in TLAPS for separating first-order and modal reasoning.

7.2 Formal Methods for Developing and Analyzing Algorithms and Systems

Participants Heba Al Kayed, Étienne André, Martin Bromberger, Zheng Cheng, Marie Duflot-Kremer, Yann Duplouy, Margaux Duroeulx, Alexis Grall, Igor Konnov, Aleksander Kryukov, Dominique Méry, Stephan Merz, Mathieu Montin, Christoph Weidenbach.

7.2.1 Contributions to Formal Methods of System Design

Simpler Rules for Auxiliary Variables. Refinement of a specification expressed at a high level of abstraction by a lower-level specification is a fundamental concept in formal system development. A key problem in proving refinement is to demonstrate that suitable values of internal variables of the high-level specification can be assigned to every possible execution of the low-level specification. The standard technique for doing so is to exhibit a *refinement mapping* where values for these variables are computed for each state, but it is also well known that this technique is not complete. In joint work with Leslie Lamport (Microsoft Research), we revisit the classic paper [59] that introduced constructions for *auxiliary variables* in order to strengthen the refinement mapping technique. In particular, we introduce simpler rules for defining prophecy variables and demonstrate how they can be used for proving the correctness of an algorithm implementing a linearizable object. We also show that our constructions of auxiliary variables yield a complete proof method. An article based on this work has been submitted for publication to a journal.

Generating Distributed Programs from Event-B Models. In [25] we present an approach for combining correct-by-construction approaches and transformations of formal models expressed in Event-B to executable programs written in DistAlgo, a domain-specific language embedded in Python. Our objective is to address the design of verified distributed programs. We define a subset LB (Local Event-B) of the Event-B modelling language restricted to events modelling typical actions of distributed programs, including internal or local computations, as well as sending and receiving messages. We define transformations of the various elements of the LB language into DistAlgo programs. The general methodology consists

in starting from a statement of the algorithmic problem and then progressively producing an LB model obtained after several refinement steps of the initial LB model. The derivation of the LB model has already been addressed in previous research. The transformation of LB models into DistAlgo programs is illustrated through a simple example. The refinement process and the soundness of the transformation allow one to produce correct-by-construction distributed programs.

An Extension of PlusCal for Distributed Algorithms. PlusCal [72] is a language for describing algorithms. It has the look and feel of pseudo-code, but also has a formal semantics through a translation to TLA$^+$ specifications. During her master internship, Heba Al Kayed extended the PlusCal language and translator so that it is more suitable for modeling distributed algorithms. As a first extension, parallel processes may have several code blocks that represent threads communicating through local variables. Second, communication channels are introduced as first-class entities, together with send, multicast, and receive operations. This work was presented at the TLA$^+$ Community Meeting [36].

Formal Analysis of Critical Interactive Systems. When interactive systems allow users to interact with critical systems, they are qualified as Critical Interactive Systems. Their design requires the support of different activities and tasks to achieve user goals. Examples of such systems are cockpits, nuclear plant control panels, medical devices, etc. Such critical systems are very difficult to model due to the complexity of the offered interaction capabilities. In joint work with Ismaël Mendil, Neeraj Kumar Singh, Yamine Aït-Ameur, and Philippe Palanque (IRIT Toulouse), we present [31] a formal framework, F3FLUID (Formal Framework For FLUID), for designing safety-critical interactive systems. It relies on FLUID as the core modelling language. FLUID enables modelling and using interactive systems domain concepts and supports an incremental design of such systems. Formal verification, validation and animation of the designed models are supported through different transformations of FLUID models into target formal verification techniques: Event-B for formal verification, ProB model checker for animation and Interactive Cooperative Objects for user validation. The Event-B models are generated from FLUID while ICO and ProB models are produced from Event-B. We exemplify the real-life case study TCAS (Traffic alert and Collision Avoidance System) to demonstrate our framework.

7.2.2 Automated Reasoning Techniques for Verification

Towards Mechanization and Application of SUPERLOG. In joint work with Markus Kroetzsch and Christof Fetzer (Technical University of Dresden), we have introduced a logical fragment called *SUPERLOG* (Supervisor Logic) that is meant to provide a basis for formalizing abstract control algorithms found in ECUs (Electronical Control Unit), plus fully automated verification, plus execution [28]. Technically, the language is an extension of the first-order Bernays-Schoenfinkel fragment with arithmetic constraints. It extends the well known SMT fragment by universally quantified variables. We have developed a sound and complete calculus for the SUPERLOG language [51]. The calculus supports non-exhaustive propagation and can therefore be a role model for other calculi where exhaustive propagation cannot be afforded [16]. Based on the decidability results obtained by Marco Voigt [77], we are working on fully automatic verification approaches for fragments of the SUPERLOG language. One line of research is to “hammer” verification conditions in SUPERLOG fragments to DATALOG [66] for efficient solving. The other is to use abstractions guiding the search of the calculus, related to our abstraction refinement approach [22].

Satisfiability Techniques for Reliability Assessment. Margaux Durœulx defended her PhD thesis [49], funded by the excellence program of University of Lorraine and prepared in cooperation with Nicolae Brînzei (Centre de Recherche en Automatique de Nancy). The thesis studies the use of satisfiability techniques for assessing the reliability of complex systems, represented by static or dynamic fault trees that determine which combinations of component failures lead to system failures. Based on encodings of fault trees in propositional logic, a SAT solver can be used to compute minimal tie sets or sequences, and these are instrumental for probabilistic realiability assessment.

7.2.3 Verification of Quantitative Systems or Properties

Statistical Model Checking of Distributed Programs. We completed in 2020 our work on developing a prototype tool for performing statistical model checking within the SimGrid framework. The goal was to give users the opportunity, in one single framework, to take advantage of both verification and simulation possibilities. To do so, we added to SimGrid the possibility to use stochastic profiles, introducing probabilities in the model of the network. The prototype tool can be interfaced with the SimGrid simulator to perform statistical model checking on the actual programs simulated using the SimGrid framework. The prototype was evaluated on examples such as the Bit Torrent protocol in which we added a probabilistic model of node failures. This work resulted in a publication at the SIMULTECH conference [27].

Hybrid System Design with Safety Constraints. Hybrid systems are characterized by the interaction of continuous dynamics and discrete control. As hybrid systems become ubiquitous and more and more complex, analysis and synthesis techniques for designing safe hybrid systems are in high demand. This is however challenging due to the nature of hybrid systems and their designs, and the question of how to formulate and reason about their safety problems. Previous work has demonstrated how to extend the discrete modeling language Event-B with support for continuous operators and how to integrate traditional refinement in hybrid system design. In the same spirit, we [52] extend previous work by proposing a strategy that can coherently refine an abstract hybrid system design with safety constraints down to the concrete one with implementable discrete control that can behave safely. Our proposal is validated on the design of a smart heating system.

7.3 Verification and Analysis of Dynamic Properties of Biological Systems

Participants Cristian Vargas Montero, Hamid Rahkooy, Thomas Sturm.

Geometric Analysis of Steady State Regimes. Our work in toricity of steady state ideals of biomodels [70] from 2019 has been accepted for journal publication and will appear during 2021. The approach there was to automatically recognize relevant geometric structure of steady state varieties in K^n, where K stands for either the complex or the real numbers. For the complex numbers we used quite complicated algebraic techniques based on Gröbner basis theory. For the real numbers, in contrast, our approach was purely based on logic. Technically we employed real quantifier elimination; SMT-solving in `QF_NRA` is a possible alternative, which has not been studied systematically. This year we managed to treat also the case of complex numbers on a purely logical basis [43, 56]. Based on arguments from algebraic model theory, this also gives insights into the interdependencies of the occurrences of relevant geometric structures over the complex numbers versus the real numbers.

Geometric toricity of a variety resembles the algebraic concept of binomiality of the corresponding polynomial ideal. Generalizing that well-studied binomiality concept of chemical reaction networks, in [42, 55] unconditional binomiality has been defined, its properties have been investigated and a linear algebra approach has been given for testing unconditional binomiality in the case of reversible reactions. A graph theoretical version of the linear algebra approach has been presented in [40].

Parametric Analysis of Steady State Regimes. *Joint work with Russell Bradford (Bath, UK), James Harold Davenport (Bath, UK), Matthew England (Coventry, UK), Hassan Errami (Bonn, Germany), Vladimir Gerdt (Dubna, Russia), Dima Grigoriev (Lille), Charles Hoyt (Bonn, Germany), Marek Košta (Bratislava, Slovak Republic), Ovidiu Radulescu (Montpellier), and Andreas Weber (Bonn, Germany)*

In [15] we address, on the one hand, the simpler question whether or not there is are unique steady states, without going into details on the exact geometry. On the other hand, we do so in dependence on *parametric* reaction rates, so that the results are necessary and sufficient formal logical conditions in the Tarski Algebra. Again, the underlying methods are of logical nature, mostly real elimination methods like virtual substitution, cylindrical algebraic decomposition, and real triangular sets.

Reduction of Reaction Network Kinetics to Multiple Timescales. *Joint work with Niclas Kruff (Aachen, Germany), Christoph Lüders (Bonn, Germany), Ovidiu Radulescu (Montpellier), Sebastian Walcher (Aachen, Germany)*

Our interdisciplinary work [54] in computer science, mathematics, and systems biology is concerned with the reduction of a system of ordinary differential equations (in time) into several simpler subsystems, each corresponding to a certain orders of magnitude of velocities, also called time scales, of the corresponding differential variables. To our knowledge this is the first mathematically rigorous approach for reaction networks that allows for multiple time scales. Previous work either did not give any formal guarantees on the obtained results, or was limited to only two different time scales. 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.

As an example consider a model related to the transmission dynamics of subtype H5N6 of the avian Influenza A virus in the Philippines in August 2017 [75]. That model is identified as `BIOMD0000000716` in the BioModels database, a repository of mathematical models of biological processes [74]. The model specifies four species: `S_b` (susceptible bird), `I_b` (infected bird), `S_h` (susceptible human), and `I_h` (infected human), the concentrations of which over time we denote by differential variables $y_1, \dots, y_4$, respectively. The input system S is given by

$$\dot{y}_1 = -\tfrac{9137}{2635182}\,y_1 y_2 - \tfrac{1}{730}\,y_1 + \tfrac{412}{73}, \qquad \dot{y}_2 = \tfrac{9137}{2635182}\,y_1 y_2 - \tfrac{4652377}{961841430}\,y_2,$$
$$\dot{y}_3 = -\tfrac{1}{6159375000}\,y_2 y_3 - \tfrac{1}{25258}\,y_3 + \tfrac{40758549}{3650000}, \qquad \dot{y}_4 = \tfrac{1}{6159375000}\,y_2 y_3 - \tfrac{112500173}{2841525000000}\,y_4.$$

Our approach reduces this to three systems T_1, T_2, T_3 along with corresponding attractive manifolds $\mathscr{M}_1$, $\mathscr{M}_2$, $\mathscr{M}_3$:

$$\begin{aligned}
T_1 &: \quad \dot{y}_1 = 1 \cdot \left(-\tfrac{9137}{2635182}\,y_1 y_2 + \tfrac{412}{73}\right), \quad \dot{y}_2 = \dot{y}_3 = \dot{y}_4 = 0 \\
\mathscr{M}_1 &: \quad y_1 y_2 = \tfrac{1085694984}{667001} \\
T_2 &: \quad \dot{y}_2 = \tfrac{1}{125} \cdot \left(-\tfrac{116309425}{192368286}\,y_2 + \tfrac{51500}{73}\right), \quad \dot{y}_3 = \dot{y}_4 = 0 \\
\mathscr{M}_2 &: \quad y_1 = \tfrac{4652377}{3335005}, \quad y_2 = \tfrac{5428474920}{4652377} \\
T_3 &: \quad \dot{y}_3 = \tfrac{1}{15625} \cdot \left(-\tfrac{15625}{25258}\,y_3 + \tfrac{203792745}{1168}\right), \quad \dot{y}_4 = \tfrac{1}{15625} \cdot \left(\tfrac{15079097}{5094352815}\,y_3 - \tfrac{112500173}{181857600}\,y_4\right) \\
\mathscr{M}_3 &: \quad y_1 = \tfrac{4652377}{3335005}, \quad y_2 = \tfrac{5428474920}{4652377}, \quad y_3 = \tfrac{7051228977}{25000}, \quad y_4 = \tfrac{441466240042010928888}{327120760850763125}.
\end{aligned}$$

Notice the explicit constant factors on the right hand sides of the differential equations. We see that the system T_2 is 125 times slower than T_1, and T_3 is another 125 times slower. The total computation time was about one second. Figure 1 visualizes the direction fields of $T_1, \dots, T_3$ on $\mathscr{M}_1, \dots, \mathscr{M}_3$, respectively.

This multiple time scale reduction of the bird flu model emphasizes a cascade of successive relaxations of different model variables. First, the population of susceptible birds relaxes, meaning that these variables reach quasi-steady state values. This relaxation is illustrated in Fig. 1(b). Then the population of infected birds relaxes as shown in Fig. 1(c). Finally, the populations of susceptible and infected humans relax to a stable steady state as shown in Fig. 1(d), following a reduced dynamics described by T_3.

Real Singularities of Implicit Ordinary Differential Equations. *Joint Work with with Werner Seiler and Matthias Seiß (Kassel, Germany)*

Implicit differential equations, i.e. equations which are not solved for a derivative of highest order, appear in many applications. In particular, the so-called differential algebraic equations may be considered as a special case of implicit equations. Compared with equations in solved form, implicit equations are more complicated to analyze and show a much wider range of phenomena. Already basic questions about the existence and uniqueness of solutions of an initial value problem become much more involved. One reason is the possible appearance of singularities.

In [57] we discuss the effective computation of geometric singularities of implicit ordinary differential equations over the real numbers using methods from logic. Via the Vessiot theory of differential

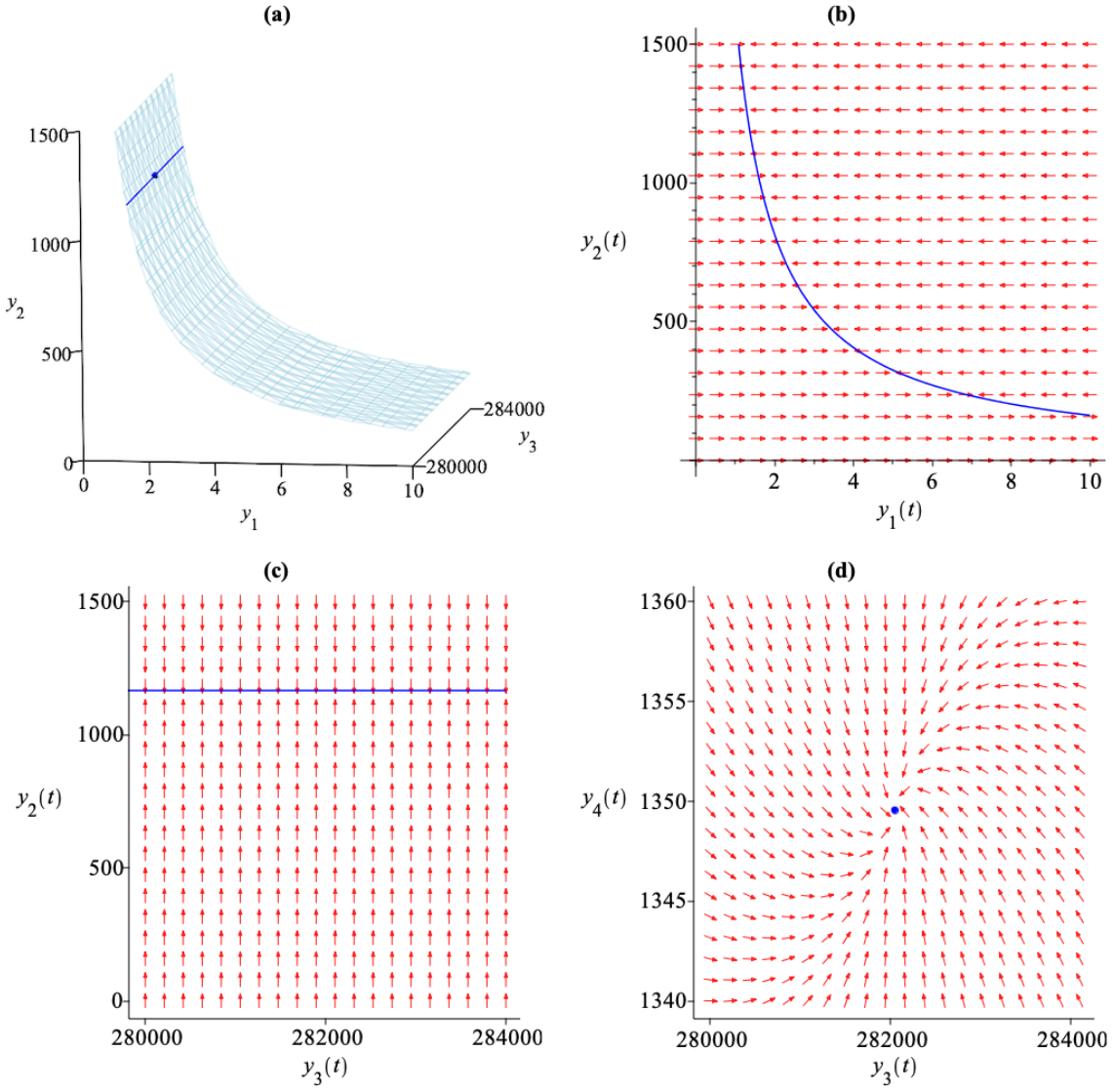

Figure 1: Reduction of an epidemic model of avian Influenza A. **(a)** The surface is the critical manifold $\mathscr{M}_1$ projected from $\mathbb{R}^4$ into real (y_1, y_2, y_3)-space. The line located at $(y_1, y_2) \approx (1.4, 1166.8)$ is the critical submanifold $\mathscr{M}_2 \subseteq \mathscr{M}_1$. The dot located at $(y_1, y_2, y_3) \approx (1.4, 1166.8, 282049.2)$ is the critical submanifold $\mathscr{M}_3 \subseteq \mathscr{M}_2$. Both $\mathscr{M}_1$ and $\mathscr{M}_2$ extend to $\pm\infty$ in both y_3 and y_4 direction, and $\mathscr{M}_3$ is located near $(1.4, 1166.8, 282049.2, 1349.6)$. **(b)** The direction field of T_1 projected from $\mathbb{R}^4$ into real (y_1, y_2)-space. The curve is the critical manifold $\mathscr{M}_1$. **(c)** The direction field of T_2 on $\mathscr{M}_1$ projected from $\mathbb{R}^4$ into real (y_3, y_2)-space. The line is the critical submanifold $\mathscr{M}_2 \subseteq \mathscr{M}_1$. **(d)** The direction field of T_3 on $\mathscr{M}_2$ projected from $\mathbb{R}^4$ into real (y_3, y_4)-space. The dot is the critical submanifold $\mathscr{M}_3 \subseteq \mathscr{M}_2$.

equations, geometric singularities can be characterized as points where the behaviour of a certain linear system of equations changes. These points can be discovered using a specifically adapted parametric generalisation of Gaussian elimination combined with real quantifier elimination methods and other logic-based simplification techniques. We demonstrate the relevance and applicability of our approach with computational experiments using a prototypical implementation based on Reduce and Redlog.

A key novelty of our approach is to consider the decisive linear system determining the Vessiot spaces first, independently of the given differential system. This allows us to make maximal use of the linearity and to apply a wide range of heuristic optimizations. Compared with the more comprehensive approach of [73], this also leads to an increased flexibility and we believe that the new approach will be in general more efficient in the sense that fewer cases will be returned.

8 Bilateral contracts and grants with industry

8.1 Bilateral Contracts with Industry

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

9 Partnerships and cooperations

9.1 International Research Visitors

9.1.1 Visits of International Scientists

Masaki Waga from NII Tokyo (now at Kyoto University) visited the team in January and worked with Étienne André.

9.1.2 Visits to International Teams

Thomas Sturm visited the research group of Werner Seiler at the University of Kassel, Germany, during February 20–21, 2020. The primary topic was joint research on logic-based methods in the area of singularities of implicit ordinary differential equations.

International travel was essentially impossible from March 2020 due to the Covid pandemic.

9.2 European Initiatives

9.2.1 FP7 & H2020 Projects

Matryoshka

Program: ERC

Title: Fast Interactive Verification through Strong Higher-Order Automation

Duration: April 2017 – March 2022

Coordinator: Jasmin Blanchette (VU Amsterdam)

Partners: Université de Lorraine (France)

Inria contact: Stephan Merz

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

More information: https://matryoshka-project.github.io.

9.2.2 Collaborations in European Programs, except FP7 and H2020

PIAF

Program: Erasmus+

Title: Pensée Informatique et Algorithmique au Fondamental / Computational and Algorithmic Thinking in Primary Education

Duration: September 2018 – August 2021

Coordinator: Université de Liège

Partners: Université du Luxembourg, Saarland University, ESPE Nancy

Inria contact: Marie Duflot-Kremer

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

ARC

Program: Erasmus+

Title: Automated reasoning in the class

Duration: October 2019 – August 2022

Coordinator: West University of Timisoara (Romania)

Partners: Johannes Kepler University Linz, RWTH Aachen, Eszterhazy Karoly University, Université de Lorraine.

Inria contact: Sorin Stratulat

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

9.3 National Initiatives

ANR International Project ProMiS

Title: Provable Mitigation of Side Channel through Parametric Verification

Duration: November 2019 – April 2022.

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

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

Participant: Étienne André

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

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

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

More information: https://www.loria.science/ProMiS/

ANR International Project SYMBIONT

Title: Symbolic Methods for Biological Networks

Duration: July 2018–April 2022

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

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

Participants: Thomas Sturm, Hamid Rahkooy

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

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

ANR Project Formedicis

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

Duration: January 2017 – December 2021

Coordinator: David Chemouil (Onera)

Partners: ENSEEIHT/IRIT Toulouse, ENAC, Université de Lorraine

Participants: Dominique Méry

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

ANR Project DISCONT

Title: Correct integration of discrete and continuous models

Duration: March 2018 – February 2023

Coordinator: Dominique Méry

Partners: ENSEEIHT/IRIT Toulouse, LACL, ClearSy, Université de Lorraine

Participants: Dominique Méry, Zheng Cheng

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

More information: https://discont.loria.fr/

ANR Project PARDI

Title: Verification of parameterized distributed systems

Duration: January 2017 – December 2021

Coordinator: Philippe Quéinnec (ENSEEIHT/IRIT Toulouse)

Partners: Université Paris Sud/LRI, Université Nanterre/LIP6, Inria Nancy – Grand Est

Participants: Stephan Merz

Summary: Distributed systems and algorithms are parameterized by the number of participating processes, the communication model, the fault model, and more generally the properties of interaction among the processes. The project aims at providing methodological and tool support for verifying parameterized systems, using combinations of model checking and theorem proving. VeriDis contributes its expertise on TLA$^+$ and its verification tools, and the integration with the Cubicle model checker is a specific goal of the project.

More information: http://pardi.enseeiht.fr/

PIA2 ISITE LUE - Digitrust

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

Duration: 2016 – 2020

Coordinator: Marine Minier

Participants: Margaux Durœulx, Stephan Merz

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

Inria IPL HAC SPECIS

Title: High-performance application and computers: studying performance and correctness in simulation

Duration: June 2016 – June 2020

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

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

Participants: Marie Duflot-Kremer, Yann Duplouy, Stephan Merz

Summary: The goal of HAC SPECIS was to allow the study of real HPC systems with respect to both correctness and performance. To this end, this Inria Project Lab assembled experts from the HPC, formal verification, and performance evaluation communities. VeriDis contributed its expertise in formal verification techniques. In particular, our goal was to extend the functionalities of exhaustive and statistical model checking within the SimGrid platform. The project was successfully evaluated in the fall of 2020.

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

DFG Transregional Research Center 248 CPEC

Title: Foundations of Perspicuous Software Systems.

Duration: January 2019 – December 2022.

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

Partners: Saarland University, University of Dresden, Max Planck Institute for Software Systems, Saarbrücken.

Participants: Fajar Haifani, Sophie Tourret, Christoph Weidenbach.

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

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

10 Dissemination

10.1 Promoting Scientific Activities

10.1.1 Scientific Events: Organization

General Chair, Scientific Chair

- Étienne André was a general chair of the 41st International Conference on Application and Theory of Petri Nets and Concurrency (Petri Nets 2020), that was organized virtually due to the COVID-19 pandemic.
- Stephan Merz, together with Markus Kuppe and Leslie Lamport, chaired the TLA$^+$Community Meeting, organized online as a satellite event of DISC 2020.
- Thomas Sturm was the chair of the steering committee of the International Symposium on Symbolic and Algebraic Computation (ISSAC), an ACM conference series.

Member of Organizing Committees

- Stephan Merz and Christoph Weidenbach are co-organizers of the International Summer School on Verification Techniques, Systems, and Applications (VTSA) that has been organized since 2008 in the Greater Region (Nancy, Saarbrücken, Luxembourg, Liège, and Koblenz). In 2020, VTSA had to be canceled due to the COVID-19 pandemic.
- Christoph Weidenbach is a co-organizer of the final round of the German Computer Science Competition for High School Students (BWINF) that took place online in September 2020.

10.1.2 Scientific Events: Selection

Chair of Conference Program Committees

- Dominique Méry was co-chair of the program committee of the 7th International Conference on Rigorous State Based Methods (ABZ 2020).
- Pascal Fontaine and Sophie Tourret were co-chairs of the 7th Workshop on Practical Aspects of Automated Reasoning.

Member of Conference Program Committees

- Étienne André served on the program committees of the 25th International Conference on Engineering of Complex Computer Systems (ICECCS), the 22nd International Conference on Formal Engineering Methods (ICFEM), the 16th International Conference on integrated Formal Methods (iFM), the 25th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC), and 14th International Symposium on Theoretical Aspects of Software Engineering (TASE).

- Pascal Fontaine served on the program committees of the 10th International Joint Conference on Automated Reasoning (IJCAR 2020), the 23rd International Conference on Theory and Applications of Satisfiability Testing (SAT), the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020), and the 18th International Workshop on Satisfiability Modulo Theories (SMT).

- Dominique Méry served on the program committees of the 14th International Symposium of Theoretical Aspects of Software Engineering (TASE), the 17th International Colloquium on Theoretical Aspects of Computing (ICTAC), the 25th International Conference on Engineering of Complex Computer Systems (ICECCS), and the 22nd International Conference on Formal Engineering Methods (ICFEM).

- Stephan Merz served on the program committees of the International Conference on Rigorous State Based Methods (ABZ), the International Conference on Formal Techniques for Distributed Objects, Components, and Systems (FORTE), the 22nd International Conference on Formal Engineering Methods (ICFEM), the International Conference on Integrated Formal Methods (iFM), the International Conference on Tests and Proofs (TAP), and the International Workshop on Algebraic Development Techniques (WADT).

- Sorin Stratulat served on the program committees of the International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), the International Conference on Information Assurance and Security (IAS), the Working Formal Methods Symposium (FROM), the International Conference on EUropean Transnational Educational (ICEUTE), and the International Conference on Computational Intelligence in Security for Information Systems (CISIS).

- Thomas Sturm was a member of the program committees of the 22nd Conference on Computer Algebra in Scientific Computing (CASC 2020) and the 5th International Workshop on Satisfiability Checking and Symbolic Computation (SC-Square 2020).

- Sophie Tourret was a member of the program committees of the 34th AAAI Conference on Artificial Intelligence (AAAI-20) and the 29th International Joint Conference on Artificial Intelligence (IJCAI 2020).

- Uwe Waldmann was a member of the program committee of the 10th International Joint Conference on Automated Reasoning (IJCAR 2020).

- Christoph Weidenbach was a member of the program committee of the 10th International Joint Conference on Automated Reasoning (IJCAR 2020) and the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020).

10.1.3 Journals

Editorial Board Membership

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

- Christoph Weidenbach is a member of the editorial board of the *Journal of Automated Reasoning* (JAR) (Springer).

Special Issues Edited

- Thomas Sturm has edited a special issue of Mathematics in Computer Science on *Computer Algebra in Scientific Computing* [47] and a special issue of the Journal of Symbolic Computation on *Symbolic Computation and Satisfiability Checking* [46].

- Together with Armin Biere and Cesare Tinelli, Christoph Weidenbach edited the Journal of Automated Reasoning special issue *Automated Reasoning Systems* [14].

10.1.4 Invited Talks

Marie Duflot-Kremer gave a plenary invited talk at the DIDAPRO 8 - DIDASTIC conference in Lille on February 7, 2020, on how to teach computer science without a computer.[4]

10.1.5 Leadership within the Scientific Community

- Stephan Merz is a member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*.
- Christoph Weidenbach is president of CADE and a member of the IJCAR steering committee.

10.1.6 Scientific Expertise

- Étienne André was a scientific expert for the EIG CONCERT-Japan (Japan Science and Technology Agency, 2020).

10.1.7 Research Administration

- Stephan Merz is the delegate for scientific affairs at the Inria Nancy – Grand Est research center and a member of Inria's Evaluation Committee. In 2020, he was a member of the hiring committees of senior researchers at Inria and of junior researchers at Inria Paris. He is also a member of the executive committee of the project on citizens' trust in the digital world (DigiTrust) funded by *Lorraine Université d'Excellence*.
- Thomas Sturm was an external expert on the appointment committee for a professorship "W2 Computeralgebra" at the University of Kassel, Germany.
- Sophie Tourret is the editor of the newsletter of AAR, the Association for Automated Reasoning.
- Uwe Waldmann is ombudsperson of the Max Planck Institute for Informatics.
- Christoph Weidenbach is a member of the selection committee of the Saarbrücken Graduate School in Computer Science. He was an external expert on the appointment committees for professorships in Oldenburg and Regensburg, Germany. He coordinates the scientific affairs at MPI-INF.

10.2 Teaching - Supervision - Juries

10.2.1 Teaching

- DUT 1: Étienne André, Structures de données, 42 HETD, Université de Lorraine – IUT Charlemagne, France.
- DUT 1: Étienne André, Interfaces hommes machines, 57 HETD, Université de Lorraine – IUT Charlemagne, France.
- DUT 1: Étienne André, Architecture des réseaux, 32 HETD, Université de Lorraine – IUT Charlemagne, France.
- DUT 1: Étienne André, Conception orientée objets, 38 HETD, Université de Lorraine – IUT Charlemagne, France.
- DUT 2: Étienne André, Projets tuteurés, 14 HETD, Université de Lorraine – IUT Charlemagne, France.
- DUT 2: Étienne André, Stages, 42 HETD, Université de Lorraine – IUT Charlemagne, France.
- Licence: Marie Duflot-Kremer, Algorithmes et programmation 1, 60 HETD, L1, Université de Lorraine, France.

[4] https://www.didapro.org/8/conferences-invitees/ci-duflot-kremer/

- Diplôme inter universitaire: Marie Duflot-Kremer, formation d'enseignants du secondaire à la spécialité NSI, 40 HETD, Université de Lorraine, France
- Licence: Marie Duflot-Kremer, Introduction au Web, 20 HETD, L1, Université de Lorraine, France
- Licence: Marie Duflot-Kremer, Accompagnement Algorithmique, 60 HETD, L1, Université de Lorraine, France
- Master: Marie Duflot-Kremer and Stephan Merz, Elements of model checking, 40 HETD, M2 Informatique and Master Erasmus Mundus DESEM, Université de Lorraine, France.
- Master: Marie Duflot-Kremer and Stephan Merz, Algorithmes distribués, 24 HETD M1 informatique, Université de Lorraine, France.
- Master: Dominique Méry, Modeling software systems, 30 HETD, M2, Université de Lorraine, France.
- Master: Dominique Méry, Modèles et algorithmes, 30 HETD, Université de Lorraine – Télécom Nancy, France.
- Master: Uwe Waldmann and Christoph Weidenbach, Automated Reasoning I, Universität des Saarlandes, Germany.
- Master: Uwe Waldmann, Automated Reasoning II, Universität des Saarlandes, Germany.
- Master: Markus Blaeser, Karl Bringmann, and Christoph Weidenbach, Competitive Programming, Universität des Saarlandes, Germany.

10.2.2 Supervision

- PhD: Margaux Duroeulx, SAT Techniques for Reliability Assessment, Université de Lorraine. Supervised by Nicolae Brînzei, Marie Duflot-Kremer, and Stephan Merz, 5 March 2020.
- PhD: Mathias Fleury, Formalization of Logical Calculi in Isabelle/HOL, Universität des Saarlandes. Supervised by Jasmin Blanchette and Christoph Weidenbach, 28 January 2020.
- PhD in progress: Antoine Defourné, SMT for TLAPS, Université de Lorraine. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, since March 2019.
- PhD in progress: Daniel El Ouraoui, Higher-Order SMT, Université de Lorraine. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, since November 2017.
- PhD in progress: Alexis Grall, Integration of a modeling language and a language for programming distributed systems, Université de Lorraine. Supervised by Horatiu Cirstea and Dominique Méry, since October 2018. Abandoned in September 2020.
- PhD in progress: Fajar Haifani, Explications in Logic, Universität des Saarlandes. Supervised by Christoph Weidenbach, since November 2019.
- PhD in progress: Dylan Marinho, Detecting timing attacks using formal methods, Université de Lorraine. Supervised by Étienne André, since October 2020.
- PhD in progress: Pierre Lermusiaux, Analysis of properties of interactive critical systems, Université de Lorraine. Supervised by Horatiu Cirstea and Pierre-Etienne Moreau, since October 2017.
- PhD in progress: Hans-Jörg Schurr, Higher-Order SMT, Université de Lorraine. Supervised by Jasmin Blanchette, Pascal Fontaine, and Stephan Merz, since November 2017.

10.2.3 Juries

- Étienne André served as a reviewer in the PhD committees of Clément Bertrand (Université d'Evry, Université Paris-Saclay), Victor Roussanaly (Université de Rennes 1), and Jiao Jiao (Nanyang Technological University, Singapore).
- Stephan Merz served as a reviewer in the PhD committees of Ivana Vukotic (University of Luxembourg) and of Xiaojie Guo (Université Grenoble Alpes).
- Pascal Fontaine served as a reviewer in the PhD committee of Yanis Sellami (Université Grenoble Alpes).

10.3 Popularization

10.3.1 Internal or external Inria responsibilities

- Marie Duflot-Kremer is the deputy vice-president for outreach activities in the supervisory council of SIF (*Société Informatique de France*) and a member of the scientific committee of *Fondation Blaise Pascal*. She is also a member of the CAPES NSI (numérique et sciences informatique) jury, the committee for hiring secondary school teachers.
- Christoph Weidenbach is the head of the steering committee of the German Computer Science Competitions for pupils and high school students (BWINF) https://bwinf.de/.

10.3.2 Articles and contents

- Marie Duflot-Kremer is a member of the ERASMUS+ project PIAF with colleagues from Liège, Luxembourg, and Saarbrücken. This projects aims at studying how computational thinking can be introduced in primary education (with kids ranging from 5 to 12 years old). A catalogue of computational thinking competences has been designed, and test educational scenarios and didactical resources are being developed. This work was presented at the Didapro conference [32].
- As a member of the group *Informatique Sans Ordinateur* (ISO), Marie Duflot-Kremer takes part in creating new popularization activities and publishing online documentation to help people reproduce unplugged computer science activities. She also supervised two internships for 3rd year students to develop, test in classrooms and promote such activities.

10.3.3 Education

Marie Duflot-Kremer intervenes in the training of teachers. In 2020 she participated in two webinars for Educode.be.

10.3.4 Interventions

Marie Duflot-Kremer takes part every year in several local or national popularization events. In 2020 she contributed:

- a webinar for "confine ta science" during the lockdown in spring,
- a conference on computer networks for the event "filles, mathématiques et informatique, une équation lumineuse", meant to promote mathematics and computer science towards high school girls,
- a presentation on the inaugural day of the "Chiche – 1 scientist 1 class" Inria project, intending to help high school students imagine their future in (computer) science.

11 Scientific production

11.1 Major publications

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

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

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

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

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

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

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

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

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

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

11.2 Publications of the year

International journals

[11] É. André, D. Lime and N. Markey. 'Language Preservation Problems in Parametric Timed Automata'. In: *Logical Methods in Computer Science* 16.1 (22nd Jan. 2020). DOI: 10.23638/LMCS-16. URL: https://hal.archives-ouvertes.fr/hal-02498022.

[12] É. André, T. H. Tan, M. Chen, S. Liu, J. Sun, Y. Liu and J. S. Dong. 'Automated synthesis of local time requirement for service composition'. In: *Software and Systems Modeling* (13th Mar. 2020). DOI: 10.1007/s10270-020-00787-5. URL: https://hal.archives-ouvertes.fr/hal-02512449.

[13] H. Barbosa, J. Blanchette, M. Fleury and P. Fontaine. 'Scalable Fine-Grained Proofs for Formula Processing'. In: *Journal of Automated Reasoning* 64.3 (Mar. 2020), pp. 485–510. DOI: 10.1007/s10817-018-09502-y. URL: https://hal.inria.fr/hal-02515103.

[14] A. Biere, C. Tinelli and C. Weidenbach. 'Preface to the Special Issue on Automated Reasoning Systems'. In: *Journal of Automated Reasoning* 64.3 (Mar. 2020), pp. 361–362. DOI: 10.1007/s10817-019-09531-1. URL: https://hal.archives-ouvertes.fr/hal-03145431.

[15] R. Bradford, J. H. Davenport, M. England, H. Errami, V. Gerdt, D. Grigoriev, C. Hoyt, M. Košta, O. Radulescu, T. Sturm and A. Weber. 'Identifying the parametric occurrence of multiple steady states for some biological networks'. In: *Journal of Symbolic Computation* 98 (May 2020), pp. 84–119. DOI: 10.1016/j.jsc.2019.07.008. URL: https://hal.inria.fr/hal-02397154.

[16] M. Bromberger, T. Sturm and C. Weidenbach. 'A complete and terminating approach to linear integer solving'. In: *Journal of Symbolic Computation* 100 (1st Sept. 2020), pp. 102–136. DOI: 10.1016/j.jsc.2019.07.021. URL: https://hal.inria.fr/hal-02397168.

[17] P. Chocron, P. Fontaine and C. Ringeissen. 'Politeness and Combination Methods for Theories with Bridging Functions'. In: *Journal of Automated Reasoning* 64 (2020), pp. 97–134. DOI: 10.1007/s10817-019-09512-4. URL: https://hal.inria.fr/hal-01988452.

[18] A. Cropper and S. Tourret. 'Logical reduction of metarules'. In: *Machine Learning* 109.7 (2020), pp. 1323–1369. DOI: 10.1007/s10994-019-05834-x. URL: https://hal.archives-ouvertes.fr/hal-02988003.

[19] J. H. Davenport, M. England, A. Griggio, T. Sturm and C. Tinelli. 'Symbolic computation and satisfiability checking'. In: *Journal of Symbolic Computation* 100 (Sept. 2020), pp. 1–10. DOI: 10.1016/j.jsc.2019.07.017. URL: https://hal.inria.fr/hal-02397190.

[20] G. Krait, S. Lazard, G. Moroz and M. Pouget. 'Certified numerical algorithm for isolating the singularities of the plane projection of generic smooth space curves'. In: *Journal of Computational and Applied Mathematics* (2021). URL: https://hal.inria.fr/hal-03161393.

[21] A. Schlichtkrull, J. Blanchette, D. Traytel and U. Waldmann. 'Formalizing Bachmair and Ganzinger's Ordered Resolution Prover'. In: *Journal of Automated Reasoning* 64.7 (Oct. 2020), pp. 1169–1195. DOI: 10.1007/s10817-020-09561-0. URL: https://hal.inria.fr/hal-03144467.

[22] A. Teucke and C. Weidenbach. 'SPASS-AR: A First-Order Theorem Prover Based on Approximation-Refinement into the Monadic Shallow Linear Fragment'. In: *Journal of Automated Reasoning* 64.3 (Mar. 2020), pp. 611–640. DOI: 10.1007/s10817-020-09546-z. URL: https://hal.inria.fr/hal-02965030.

International peer-reviewed conferences

[23] É. André and A. Kryukov. 'Parametric non-interference in timed automata'. In: *Proceedings of the 25th International Conference on Engineering of Complex Computer Systems (ICECCS 2020)*. ICECCS 2020 - 25th International Conference on Engineering of Complex Computer Systems. IEEE Conference Proceedings. Singapore, Singapore: https://formal-analysis.com/iceccs/2020/, 4th Mar. 2021. URL: https://hal.archives-ouvertes.fr/hal-02972357.

[24] G. Busana, B. Denis, M. Duflot-Kremer, S. Higuet, L. Kataja, Y. Kreis, C. Laduron, C. Meyers, Y. Parmentier, R. Reuter and A. Weinberger. 'PIAF: promoting computational thinking and algorithmics in fundamental education'. In: Didapro 8 – DidaSTIC – L'informatique, objets d'enseignements – enjeux épistémologiques, didactiques et de formation. Actes de la 8e édition du colloque Didapro - DidaSTIC. Lille, France, 5th Feb. 2020. URL: https://hal.archives-ouvertes.fr/hal-02463940.

[25] H. Cirstea, A. Grall and D. Méry. 'Generating Distributed Programs from Event-B Models'. In: International Workshop on Verification and Program Transformation. Vol. 320. Dublin, Ireland, 7th Aug. 2020, pp. 110–124. DOI: 10.4204/EPTCS.320.8. URL: https://hal.inria.fr/hal-02997277.

[26] H. Cirstea, P. Lermusiaux and P.-E. Moreau. 'Pattern eliminating transformations'. In: LOPSTR 2020 - 30th International Symposium on Logic-Based Program Synthesis and Transformation. Bologna, Italy, 7th Sept. 2020. URL: https://hal.inria.fr/hal-02476012.

[27] M. Duflot and Y. Duplouy. 'Statistical Model Checking of Distributed Programs within SimGrid'. In: *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, SIMULTECH 2020*. SIMULTECH 2020 - 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications. Lieusaint, France, 8th July 2020. URL: https://hal.inria.fr/hal-02978389.

[28] R. Faqeh, C. Fetzer, H. Herrmanns, J. Hoffmann, M. Klauck, M. Koehl, M. Steinmetz and C. Weidenbach. 'Towards Dynamic Dependable Systems through Evidence-Based Continuous Certification'. In: ISoLA 2020 - 9th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Rhodes, Greece, Oct. 2021. URL: https://hal.archives-ouvertes.fr/hal-02965830.

[29] J. Jerray, L. Fribourg and É. André. 'Guaranteed phase synchronization of hybrid oscillators using symbolic Euler's method (verification challenge)'. In: *Proceedings of the 7th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH 2020)*. ARCH20 - 7th International Workshop on Applied Verification of Continuous and Hybrid Systems. Vol. EPiC Series in Computing. Proceedings of the 7th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH 2020) 74. Berlin, Germany: https://cps-vo.org/group/ARCH, 25th Sept. 2020, pp. 197–184. DOI: 10.29007/l3k2. URL: https://hal.archives-ouvertes.fr/hal-02972549.

[30] P. Koopmann, W. Del-Pinto, S. Tourret and R. Schmidt. 'Signature-Based Abduction for Expressive Description Logics'. In: 17th International Conference on Principles of Knowledge Representation and Reasoning {KR-2020}. Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning. Rhodes, France, 12th Sept. 2020, pp. 592–602. DOI: 10.24963/kr.2020/59. URL: https://hal.archives-ouvertes.fr/hal-03141064.

[31] I. Mendil, N. K. Singh, Y. Aït-Ameur, D. Méry and P. Palanque. 'An Integrated Framework for the Formal Analysis of Critical Interactive Systems'. In: The 27th Asia-Pacific Software Engineering Conference. The 27th Asia-Pacific Software Engineering Conference. Singapour, Singapore: https://formal-analysis.com/apsec/2020/, 1st Dec. 2020, p. 10. URL: https://hal.inria.fr/hal-02999148.

[32] Y. Parmentier, R. Reuter, S. Higuet, L. Kataja, Y. Kreis, M. Duflot-Kremer, C. Laduron, C. Meyers, G. Busana, A. Weinberger and B. Denis. 'PIAF: Developing Computational and Algorithmic Thinking in Fundamental Education'. In: AACE 2020 - EdMedia + Innovate Learning. Vol. 1. Proceedings of EdMedia + Innovate Learning 2020 Online. Amsterdam / Virtual, Netherlands: http://www.aace.org/conf/edmedia/, 2020, pp. 315–322. URL: https://hal.archives-ouvertes.fr/hal-02888504.

[33] ***Best Paper***
Y. Sheng, Y. Zohar, C. Ringeissen, J. Lange, P. Fontaine and C. Barrett. 'Politeness for the Theory of Algebraic Datatypes'. In: 10th International Joint Conference on Automated Reasoning, IJCAR. Vol. 12166. Lecture Notes in Computer Science. Paris, France, 2020, pp. 238–255. DOI: 10.1007/978-3-030-51074-9_14. URL: https://hal.inria.fr/hal-02962716.

[34] S. Stratulat. 'SPIKE, an automatic theorem prover – revisited'. In: SYNASC2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. Timisoara, Romania, 1st Sept. 2020, pp. 93–96. URL: https://hal.archives-ouvertes.fr/hal-02965319.

[35] U. Waldmann, S. Tourret, S. Robillard and J. Blanchette. 'A Comprehensive Framework for Saturation Theorem Proving'. In: IJCAR 2020 (Part I) International Joint Conference on Automated Reasoning. Vol. 12166. IJCAR 2020: Automated Reasoning. Paris, France, 24th June 2020, pp. 316–334. DOI: 10.1007/978-3-030-51074-9_18. URL: https://hal.inria.fr/hal-03106208.

Conferences without proceedings

[36] H. Alkayed, H. Cirstea and S. Merz. 'An Extension of PlusCal for Modeling Distributed Algorithms'. In: TLA+ Community Event 2020. Freiburg (online), Germany, 15th Oct. 2020. URL: https://hal.inria.fr/hal-03143502.

[37] A. Defourné. 'Better Automation for TLA+ Proofs'. In: TLA+ Community Event 2020. Virtual, France: https://conf.tlapl.us/home/, 15th Oct. 2020. URL: https://hal.archives-ouvertes.fr/hal-02990598.

[38] A. Defourné and P. Vukmirovic. 'Higher-order Automation in TLAPS'. In: JFLA 2020 - 31emes Journées Francophones des Langages Applicatifs. Gruissan, France: http://jfla.inria.fr/jfla2020.html, 29th Jan. 2020. URL: https://hal.archives-ouvertes.fr/hal-02990614.

[39] F. Haifani, P. Koopmann, S. Tourret and C. Weidenbach. 'On a Notion of Relevance'. In: Proceedings of the 33rd International Workshop on Description Logics (DL 2020). Online, Greece, 12th Sept. 2020. URL: https://hal.archives-ouvertes.fr/hal-03141063.

[40] H. Rahkooy and C. V. Montero. 'A Graph Theoretical Approach for Testing Binomiality of Reversible Chemical Reaction Networks'. In: 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing - SYNASC 2020. Timisoara/Virtual, Romania: https://synasc.ro/2020/, 1st Sept. 2020. URL: https://hal.inria.fr/hal-03140916.

[41] S. Tourret, P. Fontaine, D. El Ouraoui and H. Barbosa. 'Lifting congruence closure with free variables to λ-free higher-order logic via SAT encoding'. In: SMT 2020 - 18th International Workshop on Satisfiability Modulo Theories. Online COVID-19, France, 5th July 2020. URL: https://hal.archives-ouvertes.fr/hal-03049088.

Scientific book chapters

[42] H. Rahkooy, O. Radulescu and T. Sturm. 'A Linear Algebra Approach for Detecting Binomiality of Steady State Ideals of Reversible Chemical Reaction Networks'. In: *Computer Algebra in Scientific Computing: 22nd International Workshop - CASC 2020*. 2nd Oct. 2020, pp. 492–509. DOI: 10.1007/978-3-030-60026-6_29. URL: https://hal.archives-ouvertes.fr/hal-02977486.

[43] H. Rahkooy and T. Sturm. 'First-Order Tests for Toricity'. In: *Computer Algebra in Scientific Computing, CASC 2020*. Vol. 12291. 2nd Oct. 2020, pp. 510–527. DOI: 10.1007/978-3-030-60026-6_30. URL: https://hal.archives-ouvertes.fr/hal-02977488.

[44] L. Viard, L. Ciarletta and P.-E. Moreau. 'A Mission Definition, Verification and Validation Architecture'. In: *Formal Methods. FM 2019 International Workshops*. Vol. 12232. 13th Aug. 2020, pp. 281–287. DOI: 10.1007/978-3-030-54994-7_20. URL: https://hal.archives-ouvertes.fr/hal-02963914.

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

[45] Y. Aït-Ameur, S. Nakajima and D. Méry. *Implicit and Explicit Semantics Integration in Proof-Based Developments of Discrete Systems*. 2021. DOI: 10.1007/978-981-15-5054-6. URL: https://hal.inria.fr/hal-02910199.

[46] J. H. Davenport, M. England, A. Griggio, T. Sturm and C. Tinelli. *Special Issue : Symbolic Computation and Satisfiability Checking*. Vol. 100. Sept. 2020. URL: https://hal.archives-ouvertes.fr/hal-03142461.

[47] M. England, W. Koepf, T. M. Sadykov, W. Seiler and T. Sturm, eds. *Special Issue: Proceedings of the 21st International Workshop on Computer Algebra in Scientific Computing (CASC 2019)*. Moscow, Russia, 2021. URL: https://hal.archives-ouvertes.fr/hal-03142481.

[48] A. Raschke, D. Méry and F. Houdek, eds. *Rigorous State-Based Methods - 7th International Conference, {ABZ} 2020, Ulm, Germany, May 27-29, 2020, Proceedings*. ABZ 2020. Vol. Lecture Notes in Computer Science. Rigorous State-Based Methods - 7th International Conference, {ABZ} 2020, Ulm, Germany, May 27-29, 2020, Proceedings 12071. ULM, Germany, 27th May 2020. URL: https://hal.inria.fr/hal-02999312.

Doctoral dissertations and habilitation theses

[49] M. Duroeulx. 'Reliability assessment of systems modeled by fault trees thanks to satisfiability techniques'. Université de Lorraine, 5th Mar. 2020. URL: https://hal.univ-lorraine.fr/tel-02881242.

[50] M. Fleury. 'Formalization of Logical Calculi in Isabelle/HOL'. Universität des Saarlandes Saarbrücken, 28th Jan. 2020. URL: https://hal.univ-lorraine.fr/tel-02963301.

Reports & preprints

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

[52] Z. Cheng and D. Méry. *A Refinement Strategy for Hybrid System Design with Safety Constraints*. Université de Lorraine; INRIA; CNRS, 9th July 2020. URL: https://hal.inria.fr/hal-02895528.

[53] H. Cirstea, A. Grall and D. Méry. *Generating Distributed Programs from Event-B Models*. LORIA UMR 7503 CNRS, INRIA, Université de LORRAINE, 13th May 2020, p. 36. URL: https://hal.inria.fr/hal-02572971.

[54] N. Kruff, C. Lüders, O. Radulescu, T. Sturm and S. Walcher. *Algorithmic Reduction of Biological Networks With Multiple Time Scales*. 20th Oct. 2020. URL: https://hal.archives-ouvertes.fr/hal-02977490.

[55] H. Rahkooy, O. Radulescu and T. Sturm. *A Linear Algebra Approach for Detecting Binomiality of Steady State Ideals of Reversible Chemical Reaction Networks*. 28th Feb. 2020. DOI: 10.1007/978-3-030-60026-6_29. URL: https://hal.archives-ouvertes.fr/hal-03017913.

[56] H. Rahkooy and T. Sturm. *First-Order Tests for Toricity*. 10th Feb. 2020. DOI: 10.1007/978-3-030-60026-6_30. URL: https://hal.archives-ouvertes.fr/hal-03017907.

[57] W. M. Seiler, M. Seiss and T. Sturm. *A Logic Based Approach to Finding Real Singularities of Implicit Ordinary Differential Equations*. 2nd Mar. 2020. URL: https://hal.archives-ouvertes.fr/hal-02978976.

[58] S. Stratulat. *E-Cyclist: Implementation of an Efficient Validation of FOL ID Cyclic Induction Reasoning (System Description)*. 3rd Feb. 2020. URL: https://hal.archives-ouvertes.fr/hal-02464242.

11.3 Cited publications

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

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

[61] 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.

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

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

[64] H. Barbosa, A. Reynolds, D. El Ouraoui, C. Tinelli and C. Barrett. 'Extending SMT Solvers to Higher-Order Logic'. In: *CADE-27*. Vol. 11716. Lecture Notes in Computer Science. Natal, Brazil: Springer, Aug. 2019, pp. 35–54. DOI: 10.1007/978-3-030-29436-6_3. URL: https://hal.archives-ouvertes.fr/hal-02300986.

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

[66] S. Ceri, G. Gottlob and L. Tanca. 'What you Always Wanted to Know About Datalog (And Never Dared to Ask)'. In: *IEEE Trans. Knowl. Data Eng.* 1.1 (Mar. 1989), pp. 146–166.

[67] P. Chocron, P. Fontaine and C. Ringeissen. 'A Polite Non-Disjoint Combination Method: Theories with Bridging Functions Revisited'. In: *25th International Conference on Automated Deduction, CADE-25*. Ed. by A. P. Felty and A. Middeldorp. Vol. 9195. Lecture Notes in Computer Science. Christoph Benzmueller. Berlin, Germany: Springer, Aug. 2015, pp. 419–433. DOI: 10.1007/978-3-319-21401-6_29. URL: https://hal.inria.fr/hal-01157898.

[68] P. Chocron, P. Fontaine and C. Ringeissen. 'A Rewriting Approach to the Combination of Data Structures with Bridging Theories'. In: *Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015*. Ed. by C. Lutz and S. Ranise. Vol. 9322. Lecture Notes in Computer Science. Wroclaw, Poland: Springer, Sept. 2015, pp. 275–290. DOI: 10.1007/978-3-319-24246-0_17. URL: https://hal.inria.fr/hal-01206187.

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

[70] D. Grigoriev, A. Iosif, H. Rahkooy, T. Sturm and A. Weber. 'Efficiently and Effectively Recognizing Toricity of Steady State Varieties'. working paper or preprint. Oct. 2019. URL: https://hal.inria.fr/hal-02397107.

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

[72] L. Lamport. 'The PlusCal Algorithm Language'. In: *6th Intl. Coll. Theoretical Aspects of Computing (ICTAC 2009)*. Ed. by M. Leucker and C. Morgan. Vol. 5684. Lecture Notes in Computer Science. Kuala Lumpur, Malaysia: Springer, 2009, pp. 36–60.

[73] M. Lange-Hegermann, D. Robertz, W. Seiler and M. Seiß. *Singularities of Algebraic Differential Equations*. Preprint Kassel University (arXiv:2002.11597). 2020.

[74] 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.

[75] 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.

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

[77] M. Voigt. 'Decidable fragments of first-order logic and of first-order linear arithmetic with uninterpreted predicates'. PhD thesis. Saarland University, Saarbrücken, Germany, 2019.