IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2015

Project-Team VERIDIS

Modeling and Verification of Distributed Algorithms and Systems

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

RESEARCH CENTER
Nancy - Grand Est

THEME
Proofs and Verification

Table of contents

Project-Team VERIDIS

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

Keywords:

Computer Science and Digital Science:

2.1.11. - Proof languages
2.1.7. - Distributed programming
2.4.1. - Analysis
2.4.2. - Verification
2.4.3. - Proofs
7.4. - Logic in Computer Science
7.6. - Computer Algebra

Other Research Topics and Application Domains:

2.7. - Medical devices
6.1. - Software industry
6.6. - Embedded systems

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

1. Members

Research Scientists

Jasmin Christian Blanchette [Inria, Starting Research Scientist]
Stephan Merz [Team leader, Inria, Senior Researcher, HdR]
Thomas Sturm [Max-Planck Institut für Informatik, Senior Researcher, HdR]
Uwe Waldmann [Max-Planck Institut für Informatik, Researcher]
Christoph Weidenbach [Team leader, Max-Planck Institut für Informatik, Senior Researcher, HdR]

Faculty Members

Marie Duflot-Kremer [Univ. Lorraine, Associate Professor]
Pascal Fontaine [Univ. Lorraine, Associate Professor]
Dominique Méry [Univ. Lorraine, Professor, HdR]
Martin Quinson [Univ. Lorraine, Associate Professor, until Aug 2015, HdR]

Engineers

Gabriel Corona [Univ. Lorraine]
Simon Cruanes [Inria]
Matthieu Nicolas [Inria]
Martin Riener [Inria]

PhD Students

Gabor Alági [Univ. des Saarlandes, since Nov 2012]
Manamiary Andriamiarina [Univ. de Lorraine, until Mar 2015]
Noran Azmy [Univ. des Saarlandes, since Nov 2012]
Haniel Barbosa [Inria, since Dec 2013]
Martin Bromberger [Univ. des Saarlandes, since Jul 2014]
Pablo Dobal [Univ. Lorraine, until Aug 2015]

Mathias Fleury [Univ. des Saarlandes, since Sep 2015]
Marion Guthmuller [Univ. Lorraine, until Jun 2015]
Souad Kherroubi [Univ. Lorraine, since Jan 2015]
Marek Košta [Univ. des Saarlandes, since Nov 2011]
Marco Voigt [Univ. des Saarlandes, since Nov 2013]
Daniel Wand [Univ. des Saarlandes, since Feb 2011]

Post-Doctoral Fellows

Maximilian Jaroschek [Max-Planck Institut für Informatik]
Hernán Pablo Vanzetto [Inria, Jan to Mar 2015]

Visiting Scientists

Raul Fervari [Univ. Nacional de Córdoba, Sep 2015]
Guillaume Hoffmann [Univ. Nacional de Córdoba, Sep 2015]

Administrative Assistants

Sophie Drouot [Inria]
Martine Kuhlmann [CNRS]
Jennifer Müller [Max-Planck Institut für Informatik]

Other

David Déharbe [Univ. Federal do Rio Grande do Norte, Brazil]

2. Overall Objectives

2.1. Overall Objectives

The VeriDis project team includes members of the MOSEL team at LORIA, the computer science laboratory in Nancy, and members of the Automation of Logic Research Group at Max-Planck-Institut für Informatik (MPI-INF) in Saarbrücken. It is headed by Stephan Merz and Christoph Weidenbach. VeriDis was created in 2010 as a local team 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 formal development of concurrent and distributed algorithms and systems, within the framework of mathematically precise and practically applicable development methods. We intend to assist algorithm and system designers carrying out formally proved developments, where proofs of relevant properties, as well as bugs, can be found with a high degree of automation.

Verification techniques based on theorem proving are already having substantial impact. In particular, they have been successfully applied to the verification and analysis of sequential programs, often in combination with static analysis and software model checking. 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 inherent complexity of the problem this cannot be achieved in general. However, we have observed important advances in theorem proving in recent years. We are particularly interested in the integration of different deduction techniques and tools, such as automated theorem proving for relevant theories, such as different fragments of arithmetic. These advances suggest that a substantially higher degree of automation can be achieved in system verification than what is available in today's verification tools.

VeriDis aims at exploiting and further developing automation in system verification, and at applying its techniques within the context of concurrent and distributed algorithms, which are by now ubiquitous and whose verification is a big challenge. Concurrency problems are central to the development and verification of programs for multi- and many-core architectures, and distributed computation underlies the paradigms of grid and cloud computing. The potential of distributed systems for increased resilience to component failures makes them attractive in many contexts, but also makes formal verification important and challenging. We aim at moving current research in this area to a new level of productivity and quality. To give a concrete example: today the designer of a new distributed protocol may validate it using testing or model checking. Model checking will help finding bugs, but can only guarantee properties of a high-level model of the protocol, usually restricted to finite instances. Testing distributed systems and protocols is notoriously difficult because corner cases are hard to establish and reproduce. Also, many testing techniques require an executable, whose production is expensive and time-consuming, and since an implementation is needed, errors are found only when they are expensive to fix. The techniques that we develop aim at automatically proving significant properties of the protocol already during the design phase. Our methods mainly target designs and algorithms at high levels of abstraction; we aim at components of operating systems, distributed services, and down to the (mobile) network systems industry.

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 SPASS [10], one of the leading automated theorem provers for first-order logic based on the superposition calculus [50]. The group also studies general frameworks for the combination of theories such as the locality principle [70] and automated reasoning mechanisms these induce. Finally, members of the group design effective quantifier elimination methods and decision procedures for algebraic theories, supported by their efficient implementation in the Redlog system [4].

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 (Satisfiability Modulo Theories [52]) solver that combines decision procedures for different fragments of first-order logic and that integrates an automatic theorem prover for full 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.

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 not expressible 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, e.g. by embedding a decision procedure 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 MSR-Inria Centre in Saclay on the development of methods and tools for the formal proof of TLA$^+$ [64] specifications. Our prover relies on a declarative proof language, and calls upon several automatic backends [3]. Trust in the correctness of the overall proof can be ensured when the backends provide justifications that can be checked by the trusted kernel of a proof assistant. During the development of a proof, most obligations that are passed to the prover actually fail – for example, because necessary information is not present in the context or because the invariant is too weak, and we are interested in explaining failed proof attempts to the user, in particular through the construction of counter-models.

3.2. Formal Methods for Developing Algorithms and Systems

Powerful theorem provers are not used in isolation, but they support the application of sound methodologies for modeling and verifying systems. In this respect, members of VeriDis have gained expertise and recognition in making contributions to formal methods for concurrent and distributed algorithms and systems [2], [9], and in applying them to concrete use cases. In particular, the concept of *refinement* [49], [51], [66] 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 verifying specific algorithms. We are also interested in applications of formal methods and their associated tools to the development of systems that underlie specific certification requirements in the sense of, e.g., Common Criteria. Finally, we are interested in the adaptation of model checking techniques for verifying actual distributed programs, rather than high-level models.

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

4. Application Domains

4.1. Application Domains

Distributed algorithms and protocols are found at all levels of computing infrastructure, from many-core processors and systems-on-chip to wide-area networks. We are particularly interested in the verification of algorithms that are developed for supporting novel computing paradigms, including ad-hoc networks that underly mobile and low-power computing or overlay networks and peer-to-peer networking 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 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.

Formal verification techniques can contribute to certifying the correctness of systems. In particular, they help assert under which assumptions an algorithm or system functions as required. For example, the highest levels of the Common Criteria for Information Technology Security Evaluation encourage the use of formal methods. While initially the requirements of certified development have mostly been restricted to safety-critical systems, the cost of unavailable services due to malfunctioning system components and software provides wider incentives for verification. For example, we are working on modeling and verifying medical devices that require closed-loop models of both the system and its environment.

5. Highlights of the Year

5.1. Highlights of the Year

Pascal Fontaine and Thomas Sturm, together with Erika Abraham (RWTH Aachen) and Dongming Wang (Beihang University, Beijing) organized the Dagstuhl Seminar 15471 in November 2015, on the subject of *Symbolic Computation and Satisfiability Checking*, bringing together two communities on subjects that are particularly relevant for our team.

Jasmin Blanchette and Christoph Weidenbach, together with Nikolaj Bjørner (Microsoft) and Viorica Sofronie-Stokkermans (University of Koblenz-Landau) organized the Dagstuhl Seminar 15381 in September 2015, on the subject of *Information from Deduction: Models and Proofs*. That seminar focused on added value of deduction tools beyond a yes/no answer, in particular certificates of (un)satisfiability.

We have made considerable progresses with the symbolic analysis of reaction networks. Within this interdisciplinary project, our methods have been accepted at the leading conference in symbolic computation [33], and our results with those methods have been published in a renowned journal in the natural sciences [17].

6. New Software and Platforms

6.1. The Redlog Computer Logic System

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 has been publicly available since 1995 and is constantly being improved. The name Redlog stands for Reduce Logic System. Andreas Dolzmann from Schloss Dagstuhl Leibniz-Zentrum für Informatik is a co-developer of Redlog.

Reduce and Redlog are open-source and freely available under a modified BSD license at http://reduce-algebra.sourceforge.net/. The Redlog homepage is located at http://www.redlog.eu/. Redlog generally works with interpreted first-order logic in contrast to free first-order logic. Each first-order formula in Redlog must exclusively contain atoms from one particular Redlog-supported theory, which corresponds to a choice of admissible functions and relations with fixed semantics. Redlog-supported theories include Nonlinear Real Arithmetic (Real Closed Fields), Presburger Arithmetic, Parametric QSAT, and many more.

Effective quantifier elimination procedures for the various supported theories establish an important class of methods available in Redlog. For the theories supported by Redlog, quantifier elimination procedures immediately yield decision procedures. Besides these quantifier elimination-based decision methods there are specialized, and partly incomplete, decision methods, which are tailored to input from particular fields of application.

In 2015 there was further significant progress with the identification of bifurcations in biochemical models based on real reasoning [17], [33]. With existential real quantifier elimination Redlog can now produce unsatisfiable cores in the infeasible case [27]. This is of considerable relevance in the course of using Redlog as a theory solver in SMT contexts, e.g., within the SMArT project (section 9.2).

Redlog is a widely accepted tool and highly visible in mathematics, informatics, engineering and the sciences. The seminal article on Redlog [4] has received more than 300 citations in the scientific literature so far.

- Participants: Thomas Sturm, Marek Kosta, and Maximilian Jaroschek
- Contact: Thomas Sturm
- URL: http://www.redlog.eu/

6.2. SPASS

FUNCTIONAL DESCRIPTION

SPASS is an automated theorem prover based on superposition that handles first-order logic with equality and several extensions for particular classes of theories. It has been developed since the mid-1990s at the Max-Planck Institut für Informatik in Saarbrücken. Version 3.8 is the final release of the SPASS first-order prover built on a traditional "select given loop" design; it is distributed under the FreeBSD license at http://www.spass-prover.org.

SPASS will be released in the future in the form of various reasoners for different logics, including combinations of first-order logic with background theories, in particular some forms of arithmetic. In 2015, we have continued our efforts to improve the superposition calculus as well as to develop dedicated arithmetic decision procedures for various arithmetic theories, in particular linear integer arithmetic. Our results are:

- new calculi and decidability results for finite domain fragments,
- specialized reasoning support for finite subsets,
- specialized decision procedures for linear real arithmetic with one quantifier alternation,
- new efficient and complete procedures for (mixed) linear integer arithmetic,
- decidability results and respective procedures for various combinations of linear arithmetic with first-order logic.
- Participants: Martin Bromberger, Thomas Sturm, Marco Voigt, Uwe Waldmann, Christoph Weidenbach
- Contact: Christoph Weidenbach
- URL: http://www.spass-prover.org/

6.3. The TLA+ Proof System

FUNCTIONAL DESCRIPTION

TLAPS, the TLA$^+$ proof system developed at the Joint MSR-Inria Centre, 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 independent proof steps. 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$^+$ 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.

The current version 1.4.3 of TLAPS was released in June 2015, it is distributed under a BSD-like license. The prover fully handles the non-temporal part of TLA$^+$. Basic temporal logic reasoning is supported through an interface with a decision procedure for propositional temporal logic that performs on-the-fly abstraction of first-order subformulas. Symmetrically, subformulas whose main operator is a connective of temporal logic are abstracted before being sent to backends for first-order logic.

A complete rewrite of the proof manager has started in 2015. Its objectives are to replace the ad-hoc parser used so far with an interface to SANY, the standard parser and semantic analyzer for TLA$^+$, to extend the scope of the fragment of TLA$^+$ that is handled by TLAPS, and general code refactoring and performance improvements.

TLAPS has been used in several case studies in 2015, including the proof of determinacy of PharOS (section 8.1) and the verification of the Pastry routing protocol (section 7.2). These case studies feed back into the standard library of the distribution.

- Participants: Stephan Merz, Martin Riener, Hernán Vanzetto
- Contact: Stephan Merz
- URL: http://tla.msr-inria.inria.fr/tlaps/content/Home.html

6.4. The veriT Solver

FUNCTIONAL DESCRIPTION

VeriT is an open, trustable and efficient SMT (Satisfiability Modulo Theories) solver developed in cooperation with David Déharbe from the Federal University of Rio Grande do Norte in Natal, Brazil, on leave for Clearsy. The solver can handle large quantifier-free formulas containing uninterpreted predicates and functions, and arithmetic over integers and reals. It features efficient decision procedures for uninterpreted symbols and linear arithmetic. It also has some support for user-defined theories, quantifiers, and lambda-expressions. This allows users to easily express properties about concepts involving sets, relations, etc. The prover can produce explicit proof traces when it is used as a decision procedure for quantifier-free formulas with uninterpreted symbols and arithmetic. To support the development of the tool, non-regression tests use Inria's grid infrastructure; it allows us to extensively test the solver on thousands of benchmarks in a few minutes. The veriT solver is available as open source under the BSD license at the veriT Web site.

Efforts in 2015 have been focused on efficiency, stability, and expressiveness, with a new ability for handling non-linear arithmetic. The decision procedures for uninterpreted symbols and linear arithmetic have been further improved. The integration of the solver Redlog (section 6.1) for non-linear arithmetic in the context of the SMArT project (section 9.2) now works for quantifier-free formulas with non-linear real arithmetic, but is not yet complete for combinations.

The veriT solver participated in the SMT competition SMT-COMP 2015 with decent results.

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 for discharging proof obligations generated in Event-B [53]; on a large repository of industrial and academic cases, this SMT-based plugin decreased by 75% the number of proof obligations requiring human interactions, compared to the original B prover.

- Participants: Pascal Fontaine, Pablo Dobal, David Déharbe, and Haniel Barbosa
- Partners: Université de Lorraine - Federal University of Rio Grande do Norte
- Contact: Pascal Fontaine
- URL: http://www.veriT-solver.org

7. New Results

7.1. Automated and Interactive Theorem Proving

Participants: Gabor Alági, Haniel Barbosa, Jasmin Christian Blanchette, Martin Bromberger, Simon Cruanes, Pablo Dobal, Mathias Fleury, Pascal Fontaine, Maximilian Jaroschek, Marek Košta, Stephan Merz, Martin Riener, Thomas Sturm, Hernán Pablo Vanzetto, Uwe Waldmann, Daniel Wand, Christoph Weidenbach.

7.1.1. Combination of Satisfiability Procedures

Joint work with Christophe Ringeissen from the CASSIS project-team at Inria Nancy – Grand Est, and Paula Chocron, a student at the University of Buenos Aires.

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 (to ensure the existence of an infinite model). The design of a generic combination method for non-disjoint unions of theories is clearly a hard task, 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).

We defined [24] 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. 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 [25] other theories amenable to similar combinations: this class includes the theory of equality, the theory of absolutely free data structures, and all the theories in between.

7.1.2. Adapting Real Quantifier Elimination Methods for Conflict Set Computation

The satisfiability problem in real closed fields is decidable. In the context of satisfiability modulo theories, the problem restricted to conjunctive sets of literals, that is, sets of polynomial constraints, is of particular importance. One of the central problems is the computation of good explanations of the unsatisfiability of such sets, i.e. obtaining a small subset of the input constraints whose conjunction is already unsatisfiable. We have adapted two commonly used real quantifier elimination methods, cylindrical algebraic decomposition and virtual substitution, to provide such conflict sets and demonstrate the performance of our method in practice [27].

7.1.3. Codatatypes and Corecursion

Joint work with Andrei Popescu and Dmitriy Traytel (Technische Universität München) and Andrew Reynolds (EPFL).

Datatypes and codatatypes are useful for specifying and reasoning about (possibly infinite) computational processes. The Isabelle/HOL proof assistant is being extended with flexible and convenient support for (co)datatypes and (co)recursive functions on them. We extended the emergent framework for (co)codatatypes with automatic generation of nonemptiness witnesses [22], nonemptiness being a proviso for introducing types in many logics, including Isabelle's higher-order logic. As a theoretical step towards a definitional mechanism in Isabelle, we formalized a framework for defining corecursive functions safely, based on corecursion up-to and relational parametricity [21]. The end product is a general corecursor that allows corecursive (and even recursive) calls under "friendly" operations—an improvement over the inflexible syntactic criteria of systems such as Agda and Coq.

In a related line of work, we improved the automation of the SMT solver CVC4 by designing, implementing, and evaluating a combined decision procedure for datatypes and codatatypes [31]. The procedure decides universal problems and is composable via the Nelson–Oppen method, as implemented in SMT solvers. The decision procedure for (co)datatypes is useful both for proving and for model finding. We have commenced work on a higher-order model finder based on CVC4, called Nunchaku, that relies heavily on the decision procedure.

7.1.4. Analysis and Generation of Structured Proofs

Joint work with Sascha Böhme (QAware GmbH), Maximilian Haslbeck and Tobias Nipkow (Technische Universität München), Daniel Matichuk (NICTA), and Steffen J. Smolka (Cornell University).

Isabelle/HOL is probably the most widely used proof assistant besides Coq. The Archive of Formal Proofs is a vast collection of computer-checked proofs developed using Isabelle, containing nearly 65 000 lemmas. We performed an in-depth analysis of the archive, looking at various properties of the proof developments, including size, dependencies, and proof style [18]. This give some insights into the nature of formal proofs.

In the context of the Sledgehammer bridge between automatic theorem provers and proof assistants, we designed a translation of machine-generated proofs into (semi-)intelligible Isabelle proofs that users can simply insert into their proof texts to discharge proof obligations [16]. While the output is designed for certifying the machine-generated proofs, it also has a pedagogical value: Unlike Isabelle's automatic tactics, which are black boxes, the proofs delivered by Sledgehammer can be inspected and understood. The direct proofs also form a good basis for manual tuning.

7.1.5. Encoding Set-Theoretic Formulas in Many-Sorted First-Order Logic

TLA$^+$ is a language for the formal specification of systems and algorithms whose first-order kernel is a variant of untyped Zermelo-Fraenkel set theory. Typical proof obligations that arise during the verification of TLA$^+$ specifications mix reasoning about sets, functions, arithmetic, tuples, and records. Encoding such formulas in the input languages of standard first-order provers (SMT solvers or superposition-based provers for first-order logic) is paramount for obtaining satisfactory levels of automation. For set theory, the basic idea is to represent membership as an uninterpreted predicate for the backend provers, and to reduce set-theoretic expressions to this basic predicate. This is not straightforward for formulas involving set comprehension or for proofs that rely on extensionality for inferring equality of sets. Moreover, a full development of set-theoretic expressions may lead to large formulas that can overwhelm backend provers. We describe a technique that transforms set-theoretic formulas by successively applying rewriting and abstraction until a fixed point is reached. The technique is extended to handling functions, records, and tuples, and it is the kernel of the SMT backend of the TLA$^+$ proof system (section 6.3). A paper describing our technique has been presented at the SETS workshop 2015 [46].

Although the approach was mainly intended to support proofs, we have also started work on adapting it for constructing models of formulas in set theory. Being able to construct (counter-)models can help users understand why proof attempts fail. During his internship, Glen Mével from ENS Rennes designed translation rules for a core fragment of TLA$^+$ set theory. He validated them by using the finite model finding functionality of the SMT solver CVC4 for constructing models, with encouraging preliminary results.

7.1.6. Linear Constraints in Integer Arithmetic

We have investigated linear integer constraint solving. Many existing algorithms rely on solving the rational relaxation and transferring the results to an integer branch and bound approach. This algorithm eventually terminates due to the well-known a priori exponential bounds of an integer solution. De Moura and Jovanović proposed the first model-driven terminating algorithm where the termination relies on the structure of the problem itself but not on a priori bounds [62]. However, the algorithm contained some bugs, in particular it did not terminate. We fixed the bugs by introducing the notion of Weak Cooper elimination. Termination requires adding more rules to the algorithm and refining some existing ones [23].

7.1.7. Decidability of First-Order Clause Sets

Recursion is a necessary source for first-order undecidability of clause sets. If there are no cyclic, i.e., recursive definitions of predicates in such a clause set, (ordered) resolution terminates, showing decidability. In this work we present the first characterization of recursive clause sets enabling non-constant function symbols and depth increasing clauses but still preserving decidability. For this class called BDI (Bounded Depth Increase) we present a specialized superposition calculus. This work was published in the Journal of Logic and Computation [63]. Recursive clause sets also become decidable in the context of finite domain axioms. For this case we developped a new calculus that incorporates explicit partial model assumptions guiding the search [19].

7.1.8. Building Blocks for Automated Reasoning

There are automated reasoning building blocks shared between today's prime calculi for propositional logic (CDCL), propositional logic modulo theories (CDCL(T)), and first-order logic with equality (superposition). Underlying all calculi is a partial model assumption guiding inferences that are not redundant. Deciding the abstract redundancy notion is basically as difficult as the overall satisfiability problem for the respective logic, but for well-chosen partial model assumptions inferences can be guaranteed to be non-redundant at much lower cost. For example, for SAT it is possible to computed inferences in linear time [40] that are guaranteed to be non-redundant.

7.1.9. Beagle – A Hierarchic Superposition Prover

Joint work with Peter Baumgartner and Joshua Bax from NICTA, Canberra, Australia.

Hierarchic superposition is a calculus for automated reasoning in first-order logic extended by some background theory. In [20] we describe an implementation of hierarchic superposition within the Beagle theorem prover, and report on Beagle's performance on the TPTP problem library. Currently implemented background theories are linear integer and linear rational arithmetic. Beagle features new simplification rules for theory reasoning and implements calculus improvements like weak abstraction and determining (un)satisfiability w.r.t. quantification over finite integer domains.

7.1.10. Modal Tableau Systems with Blocking and Congruence Closure

Joint work with Renate A. Schmidt from the University of Manchester, UK.

For many common modal and description logics there are ways to avoid the explicit use of equality in a tableau calculus. For more expressive logics, e.g., with nominals as in hybrid modal logics and description logics, avoiding equality becomes harder, though, and for modal logics where the binary relations satisfy frame conditions expressible as first-order formulae with equality, explicit handling of equations is the easiest and sometimes the only known way to perform equality reasoning. In [32] we describe an approach for efficient handling of equality in tableau systems. We combine Smullyan-style tableaux with a congruence closure algorithm, and demonstrate that this method also permits the use of common blocking restrictions such as ancestor blocking.

7.1.11. Subtropical Real Root Finding

This research is motivated by a series of studies of Hopf bifurcations [60], [59] for reaction systems in chemistry and gene regulatory networks in systems biology. The relevant systems are originally given in terms of ordinary differential equations, for which Hopf bifurcations can be described algebraically [54], [74], [58], [57], typically resulting in one very large multivariate polynomial equation $f = 0$ subject to a few much simpler polynomial side conditions $g_1 > 0$, ..., $g_n > 0$. For these algebraic systems one is interested in feasibility over the reals and, in the positive case, in at least one feasible point. It turns out that, generally, scientifically meaningful information can be obtained already by checking only the feasibility of $f = 0$, which is the focus of this project. For further details on the motivating problems, we refer to our earlier publications [72], [71], [56], [55].

With one of our models, viz. *Mitogen-activated protein kinase (MAPK)*, we obtain and solve polynomials of considerable size. Our currently largest instance `mapke5e6` contains 863,438 monomials in 10 variables. One of the variables occurs with degree 12, all other variables occur with degree 5. Such problem sizes are clearly beyond the scope of classical methods in symbolic computation. To give an impression, the size of an input file with `mapke5e6` in infix notation is 30 MB large. LaTeX-formatted printing of `mapke5e6` would fill more than 5000 pages in this report.

We have developed an incomplete but terminating algorithm for finding real roots of large multivariate polynomials [33]. The principal idea is to take an abstract view of the polynomial as the set of its exponent vectors supplemented with sign information on the corresponding coefficients. To that extent, out approach is quite similar to tropical algebraic geometry [73]. However, after our abstraction we do not consider tropical varieties but employ linear programming to determine certain suitable points in the Newton polytope, which somewhat resembles successful approaches to sum-of-square decompositions [67].

We have implemented our approach in Reduce [61] using direct function calls to the dynamic library of the LP solver Gurobi [48]. In practical computations on several hundred examples originating from the work within an interdisciplinary research group our method has failed due to its incompleteness in only 10 percent of the cases. The longest computation time observed was around 16 s for the above-mentioned `mapke5e6`. With a publication of our computational results in a physics journal, our research had considerable impact beyond computer science [17].

7.1.12. Standard Answers for Virtual Substitution

Joint work with A. Dolzmann from Leibniz-Zentrum für Informatik in Saarbrücken, Germany.

We consider existential problems over the reals. Extended quantifier elimination generalizes the concept of regular quantifier elimination by additionally providing answers which are descriptions of possible assignments for the quantified variables. Implementations of extended quantifier elimination via virtual substitution have been successfully applied to various problems in science and engineering.

So far, the answers produced by these implementations included infinitesimal and infinite numbers, which are hard to interpret in practice. This has been explicitly criticized in the scientific literature. In our article [44], we introduce a complete post-processing procedure to convert, for fixed values of parameters, all answers into standard real numbers. We furthermore demonstrate the successful application of an implementation of our method within Redlog to a number of extended quantifier elimination problems from the scientific literature including computational geometry, motion planning, bifurcation analysis for models of genetic circuits and for mass action, and sizing of electrical networks.

7.1.13. A Generalized Framework for Virtual Substitution

We generalize the framework of virtual substitution for real quantifier elimination to arbitrary but bounded degrees [45]. We make explicit the representation of test points in elimination sets using roots of parametric univariate polynomials described by Thom codes. Our approach follows an early suggestion by Weispfenning, which has never been carried out explicitly.

We give necessary and sufficient conditions for the existence of a root with a given test point representation. These conditions are used to rule out redundant test points. Our encoding allows us to distinguish between test points that represent lower bounds and test points representing upper bounds of a satisfying interval for a given input formula. Furthermore, we show how to reduce the size of elimination sets by generalizing a well-known idea from linear virtual substitution, namely to consider only test points representing lower bounds of a satisfying interval.

Our framework relies on some external algorithm $\mathcal{A}$, which is used to eliminate a single existential quantifier from a finite set of generic formulas. The existence of $\mathcal{A}$ is guaranteed by the fact that $\mathbb{R}$ admits quantifier elimination. We briefly refer to experiments which compared the performance of our framework—when Cylindrical Algebraic Decomposition is used as the external algorithm—to other quantifier elimination algorithms. Unfortunately, our approach is not yet able to compete with other state-of-the-art quantifier

elimination algorithms. However, currently ongoing research suggests the possibility for drastic improvements in practice. Investigating this is left for future work.

7.2. Formal Methods for Developing Algorithms and Systems

Participants: Manamiary Andriamiarina, Noran Azmy, Gabriel Corona, Marie Duflot-Kremer, Marion Guthmuller, Souad Kherroubi, Dominique Méry, Stephan Merz, Martin Quinson, Christoph Weidenbach.

7.2.1. Incremental Development of Distributed Algorithms

Joint work with Mike Poppleton, University of Southampton, UK, and with Neeraj Kumar Singh from the Department of Computing and Software, McMaster University, Hamilton, Canada.

The development of distributed algorithms and, more generally, of distributed systems, is a complex, delicate, and challenging process. The approach based on refinement helps to gain formality by using a proof assistant, and proposes to apply a design methodology that starts from the most abstract model and leads, in an incremental way, to the most concrete model, for producing a distributed solution. Our work helps formalizing pre-existing algorithms, developing new algorithms, as well as developing models for distributed systems.

More concretely, we aim at an integration of the correct-by-construction refinement-based approach for distributed algorithms. Our main results during 2015 are:

- An integrated formal method for verification of liveness properties in distributed systems is introduced [43], and the verification of a self-stabilizing leader election protocol for population protocols illustrates the proposed methodology.
- Manamiary Andriamiarina completed his PhD, illustrating a method for developing distributed algorithms based on a combination of Event-B and fragment of temporal logic TLA.
- The methodology has been applied to take into account resilience in distributed systems. We describe a fully mechanized proof of correctness of self-☆ systems [42] along with an interesting case study related to P2P-based self-healing protocols.

7.2.2. Modeling Medical Devices

Joint work with Neeraj Kumar Singh from the Department of Computing and Software, McMaster University, Hamilton, Canada.

Formal modeling techniques and tools have attained sufficient maturity for formalizing highly critical systems in view of improving their quality and reliability, and the development of such methods has attracted the interest of industrial partners and academic research institutions. Building high quality and zero-defect medical software-based devices is a particular domain where formal modelling techniques can be applied effectively. Medical devices are very prone to showing unexpected system behaviour in operation when traditional methods are used for system testing. Device-related problems have been responsible for a large number of serious injuries. Officials of the US Food and Drug Administration (FDA) found that many deaths and injuries related to these devices are caused by flaws in product design and engineering. Cardiac pacemakers and implantable cardioverter-defibrillators (ICDs) are among the most critical medical devices and require closed-loop modelling (integrated system and environment modelling) for verification purposes before obtaining a certificate from the certification bodies.

Clinical guidelines systematically assist practitioners in providing appropriate health care in specific clinical circumstances. Today, a significant number of guidelines and protocols are lacking in quality. Indeed, ambiguity and incompleteness are likely anomalies in medical practice. The analysis of guidelines using formal methods is a promising approach for improving them.

Analyzing requirements is a major challenge in the area of safety- critical software, where the quality of requirements is an important issue for building a dependable critical system. Many projects fail due to lack of understanding of user needs, missing functional and non-functional system requirements, inadequate methods and tools, and inconsistent system specifications. This often results from the poor quality of system requirements. Based on our experience and knowledge, an environment model has been recognized to be a promising approach to support the requirements engineering to validate a system specification. It is crucial to get an approval and feedback at an early stage of the system development to guarantee the completeness and correctness of the requirements. In [29], we propose a method for analyzing the system requirements using closed-loop modelling technique. The closed-loop model in an integration of system model and environment model, where both the system and environment models are formalized using formal techniques. Formal verification of this closed-loop model helps to identify hidden or missing system requirements and peculiar behaviours, which are not covered earlier during requirements elicitation process. Moreover, the environment model assists in the construction, clarification, and validation of a given system requirements.

7.2.3. Verification of the Pastry routing protocol

In his PhD thesis at Saarbrücken University in 2013, Tianxiang Lu had studied the routing protocol of the Pastry algorithm [69] for maintaining a distributed hash table in a peer-to-peer network. He had discovered several problems in the published algorithm and proposed a modification of the protocol, together with a correctness proof under the hypothesis that no node ever disconnects. The proof had been checked using TLAPS, but it made many assumptions on the underlying data structures that were left unchecked. In particular, support for (modulus) arithmetic in TLAPS was too weak at the time when the proof was written.

As part of her PhD thesis, Noran Azmy studied the assumptions that had been left unproved, and found that several of them were not valid. As a consequence, she was able to find a counter-example to one of the invariants underlying the correctness proof. She corrected the assumptions, proved all of the ones that were needed for the proof using the current version of TLAPS, and also introduced higher-level abstractions that allowed her to rewrite the specification and the correctness proof of the routing protocol in a way that avoids low-level arithmetic reasoning throughout the proof. As a result, she obtained a complete machine-checked proof of Lu's variant of Pastry, still under the assumption that no node leaves the network. A paper describing the result is being submitted.

7.2.4. Proof of Determinacy of PharOS

Joint work with Selma Azaiez and Matthieu Lemerre (CEA Saclay), and Damien Doligez (Inria Paris).

The main contribution of our team to the ADN4SE project (section 8.1), in cooperation with colleagues from CEA, was to write a high-level specification of the real-time operating system PharOS in the TLA$^+$ language, and to prove a determinacy property of the model using TLAPS. Roughly speaking, determinacy means that the sequence of local states of each process during a computation does not depend on the order in which processes are scheduled, as long as there are no missed deadlines. This property simplifies the analysis and verification of programs that run on PharOS. It relies on the fact that every instruction is associated with a time window of execution, and a message can only be received by an instruction if the earliest possible execution time of that instruction is later than the latest possible execution time of the instruction sending the message. The model and proof are based on Lemerre et al. [65]. However, the underlying assumptions are made fully explicit in the formal model, and the proof is carried out in assertional rather than behavioral style. The proof was completed in 2015, and a paper describing the result is being submitted.

7.2.5. Formal Development of Component Semantics in B

Joint work with David Déharbe of Universidade Federal do Rio Grande de Norte (UFRN), Brazil.

We develop a formal model in Isabelle/HOL of the behavioral semantics of software components designed with the B method. We formalize semantic objects, based on labeled transition systems, notions of internal and externally visible behavior, and simulation. In particular, we study a variant of simulation that corresponds to refinement in the B method. We also formally represent the composition of components in the B method.

This work was presented at an invited talk at FACS 2015 in Rio de Janeiro, and an article will be published in LNCS.

7.2.6. Analysis of Distributed Legacy Applications

SimGrid is a toolkit for the study of Large-Scale Distributed Systems. It contains both a simulator with sound and validated performance models for the network, CPUs, and disks, but also an explicit model checker exploring all possible message interleavings in the application, and searching for states violating some properties specified by the user.

We recently added the ability to assess liveness properties over arbitrary and legacy codes, thanks to a system-level introspection tool that provides a detailed view of the running application to the model checker. This can for example be leveraged to verify both safety and liveness properties, on arbitrary MPI code written in C, C++ or Fortran. This work has been published in the Workshop on Formal Approaches to Parallel and Distributed Systems (4PAD) [26], while the full details appear in Guthmuller's PhD thesis [12].

In his master project, Gabriel Rodrigues Santos investigated the feasibility of implementing algorithms for statistical model checking within SimGrid. The basic idea is to sample sufficiently many executions of a program, based on probabilistic parameters associated with the execution platform, for quantifying correctness and reliability properties. By construction, the answers obtained in this way are not exact, but their imprecision can be bounded by an interval of confidence. The results are very encouraging, and we intend to pursue this approach in further work.

7.2.7. Evaluating and Verifying Probabilistic Systems

Joint work with colleagues at ENS Cachan, University Paris Est Créteil, and Ecole Centrale Paris.

Since its introduction in the 1980s, model checking has become a prominent technique for the verification of complex systems. The aim was to decide whether or not a system fulfills its specification. With the rise of probabilistic systems, new techniques have been designed to verify this new type of systems, and appropriate logics have been proposed to describe more subtle properties to be verified. However, some characteristics of such systems fall outside the scope of model checking. In particular, it is often of interest not to decide wether a property is satisfied but how well the system performs with respect to a certain measure. We have designed a statistical tool for tackling both performance and verification issues. Following several conference talks, two journal papers have been published. The first one [14] presents the approach in details together with illustrative applications to flexible manufacturing systems, and to the study of a biological mechanism knwon as circadian clock. The second one [15] focuses on biological applications, and more precisely the use of statistical model checking to detect and measure several indicators of oscillating biological systems.

8. Bilateral Contracts and Grants with Industry

8.1. ADN4SE Project

Participants: Stephan Merz, Martin Riener.

Joint work with Damien Doligez of Inria Paris Rocquencourt.

The ADN4SE project started in 2013 within *Programme d'Investissements d'Avenir: Briques Génériques du Logiciel Embarqué* and is coordinated for Inria by the Gallium team in Rocquencourt. The objective of this project is to develop and commercialize the PharOS real-time micro-kernel operating system. In cooperation with researchers at CEA List, we are contributing to the project by verifying key properties (in particular, determinism) of a high-level model of the system written in TLA$^+$. The proof was completed in the summer of 2015, and the project ended in December 2015.

8.2. Proving formulas over streams

Participants: Pascal Fontaine, Stephan Merz.

In an exploratory project with *Atelier de Qualification Logicielle* of RATP, we studied the use of SAT solving techniques for proving certain formulas expressed over infinite Boolean streams. Such formulas arise as proof obligations generated from SCADE models used by RATP, and they are currently proved using proprietary tools. We showed that in the absence of recursive definitions, checking a small number of instances of a proof obligation ensures its validity for all instances. For models that contain recursive definitions, the bound on the number of instances that must be checked becomes much bigger, making it unwieldy to apply the same technique, and inductive reasoning should be used. We implemented our proposal in a prototype checker and validated it using several benchmarks provided by RATP.

9. Partnerships and Cooperations

9.1. Regional Initiatives

Participants: Pablo Dobal, Pascal Fontaine.

The PhD thesis of Pablo Federico Dobal was jointly funded by Région Lorraine and the ANR-DFG project SMArT (section 9.2) between September 2014 and August 2015.

9.2. National Initiatives

9.2.1. ANR-DFG Project SMArT

Participants: Haniel Barbosa, David Déharbe, Pablo Dobal, Pascal Fontaine, Maximilian Jaroschek, Marek Košta, Stephan Merz, Thomas Sturm.

The SMArT (Satisfiability Modulo Arithmetic Theories) project is funded by *ANR-DFG Programmes blancs 2013*, a program of the Agence Nationale de la Recherche and the (German) Deutsche Forschungsgemeinschaft DFG. It started in April 2014. The partners are both the French and German parts of VeriDis and the Systerel company. The objective of the SMArT project is to provide advanced techniques for arithmetic reasoning beyond linear arithmetic for formal system verification, and particularly for SMT. Arithmetic reasoning is one strong direction of research at MPI, and the state-of-the-art tool Redlog (section 6.1) is mainly developed by Thomas Sturm. The SMT solver veriT (section 6.4), developed in Nancy, serves as an experimentation platform for theories, techniques and methods designed within this project.

In September 2014, Pablo Federico Dobal was hired as a PhD student in joint supervision with Saarland University, co-funded by the SMArT project and the Région Lorraine. For personal reasons, his thesis has been put on hold in September 2015.

More information on the project can be found on http://smart.gforge.inria.fr/.

9.2.2. ANR Project IMPEX

Participants: Manamiary Andriamiarina, Souad Kherroubi, Dominique Méry.

The ANR Project IMPEX is an INS ANR project that started in December 2013 for 4 years. It is coordinated by Dominique Méry, the other partners are IRIT/ENSEIHT, Systerel, Supelec and Telecom Sud Paris. The work reported here also included a cooperation with Pierre Castéran from LaBRI Bordeaux.

Modeling languages provide techniques and tool support for the design, synthesis, and analysis of the models resulting from a given modeling activity, as part of a system development process. These languages quite successfully focused on the analysis of the designed system exploiting the expressed semantic power of the underlying modeling language. The semantics of this modeling languages are well understood by the system designers and the users of the modeling language, i.e. the semantics is implicit in the model. In general, modeling languages are not equipped with resources, concepts or entities handling explicitly domain engineering features and characteristics (domain knowledge) underlying the modeled systems. Indeed, the designer has to explicitly handle the knowledge resulting from an analysis of this application domain [28], i.e. explicit semantics. Nowadays, making explicit the domain knowledge inside system design models does not obey any methodological rules validated by practice. The users of modeling languages introduce these domain knowledge features through types, constraints, profiles, etc. Our claim is that ontologies are good candidates for handling explicit domain knowledge. They define domain theories and provide resources for uniquely identifying domain knowledge concepts. Therefore, allowing models to make references to ontologies is a modular solution for models to explicitly handle domain knowledge. Overcoming the absence of explicit semantics expression in the modeling languages used to specify systems models will increase the robustness of the designed system models. Indeed, the axioms and theorems resulting from the ontologies can be used to strengthen the properties of the designed models. The objective [13] is to offer rigorous mechanisms for handling domain knowledge in design models.

9.2.3. Inria Technological Development Action CUIC

Participants: Jasmin Christian Blanchette, Simon Cruanes.

Most "theorems" initially given to a proof assistant are incorrect, whether because of a typo, a missing assumption, or a fundamental flaw. Novices and experts alike can enter invalid formulas and find themselves wasting hours, or even days, on an impossible proof. This project, funded by Inria and running from 2015 to 2017, supports the development of a counterexample generator for higher-order logic. This new tool, called Nunchaku, will be integrated in various proof assistants, including Isabelle, Coq, and the TLA$^+$ Proof System. The project is coordinated by Jasmin Blanchette and also involves Inria Saclay (EPI Toccata) and Inria Rennes (EPI Celtique), among others. Simon Cruanes was hired in October 2015 and has started the development of Nunchaku, whereas Blanchette has developed a preliminary version of the Isabelle frontend. We expect a first release in early 2016.

9.2.3.1. Inria ADT PLM (2014-2016)

Participants: Martin Quinson, Matthieu Nicolas.

Joint work with Gérald Oster (project-team Coast, Inria Nancy – Grand Est).

The goal of this project is to establish an experimental platform for studying the didactics of informatics, specifically centered on introductory programming courses.

The project builds upon a pedagogical platform for supervising programming exercises developed for our own teaching, and improves this base in several ways. We want to provide more adapted feedback to the learners, and gather more data to better understand how beginners learn programming.

This year, we heavily refactored the software into a web application, to grow the user community amongst learners and thus gather more learning analytics. We also added the ability to solve PLM exercises by assembling code blocks as in Scratch. Finally, we started working on an integrated exercise editor in the hope of growing the user community amongst teachers that will be able to propose their own exercises on top of PLM.

9.3. European Initiatives

9.3.1. FP7 & H2020 Projects

9.3.1.1. MEALS

Title: Mobility between Europe and Argentina applying Logics to Systems

Programm: FP7

Duration: October 2011 – September 2015

Coordinator: Université de la sarre

Partners:

Imperial College of Science, Technology and Medicine (United Kingdom)

Rheinisch-Westfälische Technische Hochschule Aachen (Germany)

Technische Universiteit Eindhoven (Netherlands)

Technische Universitaet Dresden (Germany)

University of Leicester (United Kingdom)

Universität des Saarlandes (Germany)

Universidad de Buenos Aires (Argentina)

Universidad Nacional de Córdoba (Argentina)

Universidad Nacional de Rio Cuarto (Argentina)

Instituto Tecnológico Buenos Aires (Argentina)

Inria contact: Castuscia Palamidessi

The MEALS project funds staff exchanges between institutions in Europe and Argentina. It is structured in five work packages (Quantitative Analysis of Concurrent Program Behaviour, Reasoning Tasks for Specification and Verification, Security and Information Flow Properties, Synthesis in Model-based Systems Engineering, Foundations for the Elaboration and Analysis of Requirements Specifications). Our team mainly cooperates with the group led by Carlos Areces in Córdoba within work package 2. In 2015, the project funded visits by Raúl Fervari and Guillaume Hoffmann in Nancy.

9.3.2. Collaborations with Major European Organizations

9.3.2.1. Cooperation with EPFL

Participants: Haniel Barbosa, Jasmin Christian Blanchette, Simon Cruanes, Pascal Fontaine.

We cooperate with Andrew Reynolds from the École polytechnique fédérale de Lausanne, Switzerland, on improving SMT solvers and bridging the gap between SMT solvers and proof assistants. This cooperation started in 2014 between Blanchette and Reynolds and has been pursued in 2015, with mutual one-week visits. The outcomes are manifold:

- We developed a decision procedure that combines reasoning about datatypes and codatatypes and implemented it in the SMT solver CVC4 [31]. This procedure is useful both for proving theorems and for model finding (counterexample generation).
- We designed an encoding of recursive and corecursive function definitions on datatypes and codatatypes that makes it possible to employ finite model finding techniques on functions with infinite domains, as long as they satisfy a wide, semantic criterion [36]. We started the development of a model finder for higher-order logic, called Nunchaku, based on this idea.
- We started work on a general framework for handling quantified formulas in SMT solving. Its focus is on the derivation of instances conflicting with a ground context, redefining the approach introduced by Reynolds et al. [68]. We enhanced the classical congruence closure algorithm so that it can handle free variables [34]. We expect the fruits of this research to be implemented in veriT and CVC4.

9.3.2.2. Cooperation with NUI Maynooth, Ireland

Participant: Dominique Méry.

The project *Building Reliable Systems: Software Refinement meets Software Verification* was a one-year project funded by PHC Ulysses. The academic Irish partner is Rosemary Monahan of NUI Maynooth. The verification of software requires the specification of preconditions and postconditions as well as other properties of the code. These properties are expressed as annotations and provide a detailed understanding of how the software is implemented. In program verification, the annotation process is often done *a posteriori*, with verification tools used to check that annotations are sound according to the semantics of the program. Determining the correct annotations to provide a complete specification is difficult, especially when specifying invariant properties of the code. *A priori* techniques for developing correct software are based on the correct-by-construction paradigm. The refinement-based approach is such a technique, providing for the construction of a correct program through the step-by-step refinement of an initial high-level model of the software. In this way, the program specification is developed alongside the code, discharging the conditions that need to be proved. We focus on combining these two software engineering techniques, to benefit from the strengths of both. We have proposed a framework for integrating the *a posteriori* paradigm Spec# and the *a priori* paradigm Event-B. This integration induces a methodology that bridges the gap between software modeling and program verification in the software development life cycle. For validating this methodology, we have designed the Rodin plugin EB2RC that implements transformations of Event-B models into algorithms.

9.4. International Initiatives

9.4.1. Participation In other International Programs

9.4.1.1. STIC AmSud MISMT

Participants: Haniel Barbosa, David Déharbe, Pablo Dobal, Pascal Fontaine, Stephan Merz.

VeriDis has a close working relationship with two South American teams at Universidade Federal do Rio Grande de Norte (UFRN), Brazil (more specifically with Prof. David Déharbe), and at Universidad Nacional de Córdoba, Argentina (more specifically with Prof. Carlos Areces). The STIC AmSud MISMT project, including both teams and VeriDis, started in 2014. It complements the MEALS project (section 9.3) and extends it to cooperation with UFRN.

The project is centered around Satisfiability Modulo Theories, with a focus on applications to Modal Logic [37]. Notably, the project supports the development of the veriT solver (section 6.4), of which David Déharbe and Pascal Fontaine are the main developers.

The project helped fund the stay of Haniel Barbosa in Natal (PhD in joint supervision between Nancy and Natal) from October to December, 2015. The project has been terminated prematurely due to funding problems.

9.4.1.2. Cooperation with NASA Ames Research Center, U.S.A.

Participant: Dominique Méry.

Joint work with Didier Fass of LORIA, Nancy.

Didier Fass and Dominique Méry have started a close working relationship with Brian Gore and his colleagues at the NASA Ames Reseach Center, Human Systems Integration Division (HSI). It is anticipated that collaboration among the researchers at NASA Ames and LORIA will lead to more formal understanding of the methods required to optimize human-systems integration issues in the design of complex human-automation systems.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Organization of scientific events

10.1.1.1. General chair, scientific chair

Jasmin Blanchette and Christoph Weidenbach co-organized the Dagstuhl seminar 15381 "Information from Deduction: Models and Proofs" with Nikolaj Bjørner (Microsoft Research, Redmond, USA) and Viorica Sofronie-Stokkermans (Universität Koblenz-Landau, Germany).

Jasmin Blanchette is an ex officio member of the steering committee of the conference series *Interactive Theorem Proving*.

Pascal Fontaine and Thomas Sturm co-organized the Dagstuhl seminar 15471 "Symbolic Computation and Satisfiability Checking" with Erika Abraham (RWTH, Aachen, Germany) and Dongming Wang (Beihang University, Beijing, China).

Stephan Merz is a member of the steering committee of the workshop on Automated Verification of Critical Systems (AVoCS). He chaired the organizing committee of the *Journées Nationales* of the working groups *Géométrie du Calcul*, *Logique, Algèbre et Calcul*, and *Logique, Types et Preuve* of GDR IM and GDR GPL in October 2015 in Nancy.

Thomas Sturm is chair of the steering committee of the conference series *Mathematical Aspects of Computer and Information Sciences* (MACIS).

Christoph Weidenbach is a member of the steering committee of *Bundeswettbewerb Informatik*, the German competition among high-school students in computer science.

10.1.1.2. Membership in organizing committees

Jasmin Blanchette was co-chair of workshops, tutorials, and system competitions of CADE-25 in Berlin.

The International Summer School on Verification Techniques, Systems, and Applications (VTSA) has been organized since 2008 in the Greater Region (Nancy, Saarbrücken, Luxembourg, Liège, and Koblenz), and Stephan Merz and Christoph Weidenbach are co-organizers of VTSA. In 2015, VTSA took place during the last week of August in Koblenz, Germany.

Dominique Méry and Stephan Merz were part of the organizing committee of the 10th edition of the conference *Modélisation des Systèmes Réactifs* (MSR 2015) in November 2015 in Nancy.

10.1.2. Service in Program Committees

10.1.2.1. Chair of conference program committees

Jasmin Blanchette co-chaired the program committee of the Ninth International Conference on Tests and Proofs in July, as part of the STAF Conference Series in L'Aquila, Italy.

Dominique Méry co-chaired the program committee of the Second Workshop on Formal Integrated Development Environments [38], a satellite of the Formal Methods conference in Oslo, in June 2015.

Stephan Merz co-chaired the program committee of the Second International Workshop on Formal Reasoning in Distributed Algorithms (FRIDA) in June, as part of the DisCoTec federation of conferences in Grenoble, and of the 10th edition of the French conference *Modélisation des Systèmes Réactifs* (MSR 2015) in Nancy.

10.1.2.2. Membership in conference program committees

Jasmin Blanchette served on the program committee of the International Conference on Automated Deduction (CADE).

Pascal Fontaine served on the program committees of the International Conference on Frontiers of Combining Systems (FroCoS) and of the workshop SMT.

Dominique Méry served on the program committees of FM2015, ICTAC 2015, ICECCS 2015, and MEDI 2015.

Stephan Merz served on the program committees of the international conferences ICFEM, NETYS, SEFM, and of the workshops AVoCS, GRSRD, SETS, and SYNASC.

Thomas Sturm served on the program committees of CASC 2015, MACIS 2015, and ISSAC 2015.

Uwe Waldmann served on the program committee of the 11th International Workshop on the Implementation of Logics.

Christoph Weidenbach served on the program committee of the International Conference on Frontiers of Combining Systems (FroCoS).

10.1.3. Editorial boards of journals

Pascal Fontaine, Thomas Sturm, and Uwe Waldmann edited Mathematics in Computer Science 9(3), a special focus volume on *Constraints and Combinations*.

Dominique Méry is a member of the editorial board of the journal *Formal Aspects of Computing* (Springer).

Stephan Merz, together with Jun Pang of the University of Luxembourg, is the editor of a special issue of Formal Engineering Methods in the journal *Formal Aspects of Computing*.

Martin Quinson is a member of the editorial board of the Interstices journal, aiming at increasing the scientific outreach of informatics.

Thomas Sturm is a member of the editorial boards of the *Journal of Symbolic Computation* (Elsevier) and *Mathematics in Computer Science* (Springer).

Christoph Weidenbach is an editor of the Journal of Automated Reasoning.

10.1.4. Invited talks

Stephan Merz gave an invited lecture at *Journées Francophones des Langages Applicatifs* in January 2015. He also gave an invited talk at Aarhus University on *Modeling Distributed Algorithms in TLA*$^+$.

Thomas Sturm gave an invited talk at FroCoS 2015 titled *From Complete Elimination Procedures to Subtropical Decisions over the Reals*.

Christoph Weidenbach gave an invited tutorial "25th Anniversary of Superposition: Status and Future" at the International Conference on Automated Deduction (CADE), together with Stephan Schulz. He gave an invited tutorial "Automated Reasoning Building Blocks" at the International Conference on Frontiers of Combining Systems (FroCoS).

10.1.5. Leadership within the scientific community

Jasmin Blanchette was elected editor of the newsletter of the Association for Automated Reasoning (AAR) and member of the AAR board.

Pascal Fontaine is one of three SMT-LIB managers. He was elected CADE trustee in October 2014.

Dominique Méry is a member of the IFIP Working Group 1.3 on *Foundations of System Specification*.

Stephan Merz is a member of the IFIP Working Group 2.2 on *Formal Description of Programming Concepts*.

Christoph Weidenbach is a trustee of CADE (elected 2009, reelected 2012).

10.1.6. Scientific expertise

Jasmin Blanchette is an expert for the French Agence Nationale de la Recherche (ANR) and the Deutsche Forschungsgemeinschaft (DFG).

Pascal Fontaine was a panel member for the CASC-25 competition of first-order theorem provers.

Dominique Méry is an expert for the French Agence Nationale de la Recherche (ANR) and for HCERES.

Stephan Merz is an expert for the French Agence Nationale de la Recherche (ANR), for the German DFG, and for the European Research Council (ERC). He helped IRISA Rennes evaluate the opportunity of creating a new research team.

Christoph Weidenbach is an advisor for the German-Israeli Foundation for Scientific Research.

10.1.7. Research administration

Dominique Méry is the head of the Doctoral School IAEM Lorraine for the University of Lorraine and head of the Formal Methods department of the LORIA laboratory.

Stephan Merz is a member of the Scientific Directorate of the International Computer Science Meeting Center in Schloss Dagstuhl. He is the delegate for the organization of conferences at the Inria Nancy – Grand Est research center and head of the PhD committee for computer science of the Doctoral School IAEM Lorraine. He is also the coordinator of the CPER *Sciences du Numérique* in Lorraine (2015–2020).

Thomas Sturm is a member of the selection committee for MSc and PhD students at the International Max Planck Research School.

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

10.2. Teaching, Supervision, Juries

10.2.1. Teaching

Licence: Jasmin Blanchette and Daniel Wand, Concrete Semantics with Isabelle/HOL, 60 HETD, Universität des Saarlandes, Germany

Licence: Marie Duflot-Kremer, Algorithmique et Programmation 1, 113 HETD L1 Mathématiques, Informatiques Sciences pour l'Ingénieur, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Projet personnel, 15 HETD, L2 informatique, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Bases de données 2, 10 HETD, L2 informatique, Université de Lorraine, France.

Licence: Marie Duflot-Kremer, Projet personnel et communication, 50 HETD, L2 informatique, Université de Lorraine, France.

Master : Marie Duflot-Kremer, Vérification de systèmes, 30 HETD, M1 Informatique, Université de Lorraine, France.

Master: Marie Duflot-Kremer and Stephan Merz, Vérification algorithmique, 30 HETD, M2 Informatique, Université de Lorraine, France.

Master: Marie Duflot-Kremer and Stephan Merz, Elements of Model Checking, 36 HETD, M2 Informatique and Master Erasmus Mundus DESEM, Université de Lorraine, France.

Licence: Pascal Fontaine, Structure des ordinateurs, 67 HETD, L2 MIASHS, parcours MIAGE, Université de Lorraine, France.

Licence: Pascal Fontaine, Logique des prédicats, 32 HETD, L2 MIASHS, Université de Lorraine, France.

Master: Pascal Fontaine, Réseaux, 50 HETD, M1 MIAGE, Université de Lorraine, France.

Master: Pascal Fontaine, Génie Logiciel, 30 HETD, M1 MIAGE, IGA Rabbat et Université de Lorraine, Maroc.

Master: Pascal Fontaine, Procédures de décision et vérification de programmes, 6 HETD, M2 informatique, Université de Lorraine, France.

Master: Dominique Méry, Models and algorithms, 60 HETD, M1, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Modeling and verification of distributed algorithms, 30 HETD, M1, Université de Lorraine, France.

Master: Dominique Méry, Modeling and verification of distributed algorithms, 30 HETD, M1 informatique, Université de Lorraine, France.

Master: Dominique Méry, Modeling and verification of software systems, 60 HETD, M1 (continued education), Telecom Nancy, Université de Lorraine, Nancy.

Master: Dominique Méry, Formal model engineering, 24 HETD, M2, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Modeling Systems, 30 HETD, M2, Telecom Nancy, Université de Lorraine, France.

Master: Dominique Méry, Modeling Systems, 36 HETD, M2 informatique and Master Erasmus Mundus DESEM, Université de Lorraine, France.

Master: Dominique Méry, Event-B modeling, 8 HETD, NUI Maynooth.

Licence: Martin Quinson, Algorithmique et Programmation, 48 HETD, L3, Telecom Nancy, Université de Lorraine, France.

Licence: Martin Quinson, Langage C et programmation shell, 48 HETD, L3, Telecom Nancy, Université de Lorraine, France.

Master: Martin Quinson, Programmation Système, 24 HETD, M1, Telecom Nancy, Université de Lorraine, France.

Master: Thomas Sturm and Christoph Weidenbach, seminar on decision procedures, Universität des Saarlandes, Germany.

Master: Uwe Waldmann, Automated Reasoning, 90 HETD, Universität des Saarlandes, Germany

10.2.2. Supervision

PhD: Manamiary Andriamiarina, Développement d'algorithmes répartis corrects par construction, Université de Lorraine. Supervised by Dominique Méry, defended on October 20, 2015.

PhD: Marion Guthmuller, Dynamic verification of distributed applications, using a model-checking approach, Université de Lorraine. Supervised by Martin Quinson and Sylvain Contassot-Vivier, defended on June 29, 2015.

PhD: Manuel Lamotte Schubert, Automatic Authorization Analysis, Saarland University. Supervised by Christoph Weidenbach, defended on September 18, 2015.

PhD: Martin Suda, Resolution-based Methods for Linear Temporal Reasoning, Universität des Saarlandes. Supervised by Christoph Weidenbach, defended on October 16, 2015.

PhD in progress: Noran Azmy, On the Automation of Proofs in TLAPS, Saarland University. Supervised by Stephan Merz and Christoph Weidenbach, since 11/2012.

PhD in progress: Haniel Barbosa, Refutational Completeness in Satisfiability Modulo Theories, Université de Lorraine and UFRN (Natal, Brazil). Supervised by David Déharbe, Pascal Fontaine, and Stephan Merz, since 12/2013.

PhD in progress: Gabor Alági, Efficient Reasoning in Finite Domains, Saarland University. Supervised by Christoph Weidenbach, since 11/2012.

PhD in progress: Martin Bromberger, Arithmetic Reasoning, Saarland University. Supervised by Christoph Weidenbach, since 07/2014.

PhD (on hold for personal reasons): Pablo Federico Dobal, Satisfiability Modulo Arithmetic Theories, Université de Lorraine and Saarland University. Supervised by Pascal Fontaine, Stephan Merz, and Thomas Sturm, since 09/2014.

PhD in progress: Mathias Fleury, Formalization of Logical Calculi, Saarland University. Supervised by Christoph Weidenbach and Jasmin Blanchette, since 09/2015.

PhD in progress: Marek Košta, Computational Logic, Universität des Saarlandes. Supervised by Thomas Sturm, since 11/2011.

PhD in progress: Marco Voigt, Decidable Hierarchic Combinations, Saarland University. Supervised by Christoph Weidenbach, since 11/2013.
PhD in progress: Daniel Wand, First-Order Extensions to Support Higher-Order Reasoning, Saarland University. Supervised by Christoph Weidenbach, since 02/2011.

10.2.3. Thesis committees

Dominique Méry served as PhD examiner at IRIT.
Stephan Merz served as a reviewer for the PhD theses of Luís Diogo Couto (Aarhus University, Denmark), Hugues Evrard (University of Grenoble), Laure Millet (University Pierre et Marie Curie, Paris), and Alexander Schimpf (University of Freiburg, Germany).
Uwe Waldmann served as a reviewer for the PhD thesis of Simon Cruanes (École Polytechnique).

10.3. Popularization

Marie Duflot-Kremer took part in various popularization activities, with a public ranging from primary school kids to high school and potential University students. A selection of these activities is given below:

- two days at "Fête de la science" in Nancy;
- a course on databases for high school professors in charge of teaching optional course ISN (Informatique et Sciences du Numérique);
- her explanations of unplugged activities have been recorded by Inria, and a Youtube channel has been created by the Interstice team to promote and share such activities;
- she is a member of two groups dedicated to the training of math teachers who will teach computer science to students of age 11 to 18;
- "Journée femmes de Sciences": one day dedicated to the promotion of science among 14 years old girls;
- she is a member of the steering committee preparing an itinerant exposition intended for explaining computer science to the public, to be released in 2016;
- she presented a poster at the Scratch2015 conference about teaching computer programming to kids from the age of 8 years onward.

Pascal Fontaine and Stephan Merz illustrated techniques that underly formal verification of protocols and algorithms at events like "Fête de la Science". Using wooden puzzles and Sudoku sheets, they explained how real-life problems can be represented in logical form and then solved using automated tools based on formal logic.
Martin Quinson organized a 2-days workshop for secondary math teachers on how algorithms could be used to reinforce the abilities of pupils to verbalize, set out arguments and conduct rigorous demonstrations (Nancy, March 2-3 2015). This workshop was co-organized by the IREM (Institut de Recherche en Enseignement des Mathématiques) of Nancy. He also co-organized a one day workshop for about 100 secondary computer science teachers. This event spreads the best practices to teach our topic (Nancy, March 12th 2015). Martin Quinson was a scientific expert in an experiment in which we explored how Scratch can be used to teach Computer Science in after school activities every week for the whole week. The results of this experiment, conducted in a collaboration between Inria and the MJC Nomade (Nancy) were published in [35].
Christoph Weidenbach lectured within the series "Perspektiven der Informatik" at Saarland University and within the public lecture series of the federal state of Saarland.

11. Bibliography

Major publications by the team in recent years

[1] T. BOUTON, D. C. B. DE OLIVEIRA, D. DÉHARBE, P. FONTAINE. *veriT: an open, trustable and efficient SMT-solver*, in "Proc. Conference on Automated Deduction (CADE)", Montreal, Canada, R. SCHMIDT (editor), Lecture Notes in Computer Science, Springer, 2009, vol. 5663, pp. 151-156

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

[3] D. COUSINEAU, D. DOLIGEZ, L. LAMPORT, S. MERZ, D. RICKETTS, H. VANZETTO. *TLA+ Proofs*, in "18th International Symposium On Formal Methods - FM 2012", Paris, France, D. GIANNAKOPOULOU, D. MÉRY (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7436, pp. 147-154

[4] A. DOLZMANN, T. STURM. *Redlog: Computer algebra meets computer logic*, in "ACM SIGSAM Bull.", 1997, vol. 31, n^{o} 2, pp. 2-9

[5] D. DÉHARBE, P. FONTAINE, S. MERZ, B. WOLTZENLOGEL PALEO. *Exploiting Symmetry in SMT Problems*, in "23rd Intl. Conf. Automated Deduction (CADE 2011)", Wroclaw, Poland, N. BJØRNER, V. SOFRONIE-STOKKERMANS (editors), LNCS, Springer, 2011, vol. 6803, pp. 222-236

[6] A. FIETZKE, C. WEIDENBACH. *Superposition as a Decision Procedure for Timed Automata*, in "Mathematics in Computer Science", 2012, vol. 6, n^{o} 4, pp. 409-425

[7] E. KRUGLOV, C. WEIDENBACH. *Superposition Decides the First-Order Logic Fragment Over Ground Theories*, in "Mathematics in Computer Science", 2012, vol. 6, n^{o} 4, pp. 427-456

[8] F. KRÖGER, S. MERZ. *Temporal Logic and State Systems*, Texts in Theoretical Computer Science, Springer, 2008, 436 p. , http://hal.inria.fr/inria-00274806/en/

[9] S. MERZ. *The Specification Language TLA^{+}*, in "Logics of Specification Languages", Berlin-Heidelberg, D. BJØRNER, M. C. HENSON (editors), Monographs in Theoretical Computer Science, Springer, 2008, pp. 401-451

[10] C. WEIDENBACH, D. DIMOVA, A. FIETZKE, M. SUDA, P. WISCHNEWSKI. *SPASS Version 3.5*, in "22nd International Conference on Automated Deduction (CADE-22)", Montreal, Canada, R. SCHMIDT (editor), LNAI, Springer, 2009, vol. 5663, pp. 140-145

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] M. B. ANDRIAMIARINA. *Developing correct-by-construction distributed algorithms*, Université de Lorraine ; Loria & Inria Grand Est, October 2015, https://hal.inria.fr/tel-01258363

[12] M. GUTHMULLER. *Dynamic formal verification of temporal properties on legacy distributed applications*, Université de Lorraine, June 2015, https://tel.archives-ouvertes.fr/tel-01231868

Articles in International Peer-Reviewed Journals

[13] Y. AIT AMEUR, D. MÉRY. *Making explicit domain knowledge in formal system development*, in "Science of Computer Programming", December 2015, https://hal.inria.fr/hal-01245832

[14] P. BALLARINI, B. BARBOT, M. DUFLOT, S. HADDAD, N. PEKERGIN. *HASL: A new approach for performance evaluation and model checking from concepts to experimentation*, in "Performance Evaluation", August 2015, vol. 90, pp. 53-77 [*DOI :* 10.1016/J.PEVA.2015.04.003], https://hal.inria.fr/hal-01221815

[15] P. BALLARINI, M. DUFLOT. *Applications of an expressive statistical model checking approach to the analysis of genetic circuits*, in "Journal of Theoretical Computer Science (TCS)", 2015, vol. 599, 30 p. [*DOI :* 10.1016/J.TCS.2015.05.018], https://hal.inria.fr/hal-01250521

[16] J. C. BLANCHETTE, S. BÖHME, M. FLEURY, S. J. SMOLKA, A. STECKERMEIER. *Semi-intelligible Isar Proofs from Machine-Generated Proofs*, in "Journal of Automated Reasoning", 2016 [*DOI :* 10.1007/S10817-015-9335-3], https://hal.inria.fr/hal-01211748

[17] H. ERRAMI, M. EISWIRTH, D. GRIGORIEV, W. M. SEILER, T. STURM, A. WEBER. *Detection of Hopf bifurcations in chemical reaction networks using convex coordinates*, in "Journal of Computational Physics", June 2015, vol. 291, pp. 279–302 [*DOI :* 10.1016/J.JCP.2015.02.050], https://hal.inria.fr/hal-01239486

Invited Conferences

[18] J. C. BLANCHETTE, M. HASLBECK, D. MATICHUK, T. NIPKOW. *Mining the Archive of Formal Proofs*, in "CICM 2015", Washington DC, United States, Intelligent Computer Mathematics - International Conference, CICM 2015, July 13-17, Proceedings, July 2015 [*DOI :* 10.1007/978-3-319-20615-8_1], https://hal.inria.fr/hal-01212594

International Conferences with Proceedings

[19] G. ALAGI, C. WEIDENBACH. *NRCL - a model building approach to the Bernays-Schönfinkel fragment*, in "Frontiers of Combining Systems, 10th International Symposium (FroCos 2015)", Wroclaw, Poland, C. LUTZ, S. RANISE (editors), Springer, 2015, vol. 9322, pp. 69-84 [*DOI :* 10.1007/978-3-319-24246-0_5], https://hal.inria.fr/hal-01247991

[20] P. BAUMGARTNER, J. BAX, U. WALDMANN. *Beagle – A Hierarchic Superposition Prover*, in "25th International Conference on Automated Deduction (CADE-25)", Berlin, Germany, A. P. FELTY, A. MIDDELDORP (editors), Lecture Notes in Computer Science, Springer, August 2015, vol. 9195, pp. 367-377 [*DOI :* 10.1007/978-3-319-21401-6_25], https://hal.inria.fr/hal-01251377

[21] J. C. BLANCHETTE, A. POPESCU, D. TRAYTEL. *Foundational Extensible Corecursion: A Proof Assistant Perspective*, in "ICFP 2015", Vancouver, Canada, Proceedings of the 20th ACM SIGPLAN International Conference on Functional Programming, September 1-3, August 2015 [*DOI :* 10.1145/2784731.2784732], https://hal.inria.fr/hal-01212589

[22] J. C. BLANCHETTE, A. POPESCU, D. TRAYTEL. *Witnessing (Co)datatypes*, in "ESOP 2015", London, United Kingdom, Programming Languages and Systems - 24th European Symposium on Programming, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, April 11-18, Proceedings, April 2015 [*DOI :* 10.1007/978-3-662-46669-8_15], https://hal.inria.fr/hal-01212587

[23] M. BROMBERGER, T. STURM, C. WEIDENBACH. *Linear Integer Arithmetic Revisited*, in "25th International Conference on Automated Deduction (CADE-25)", Berlin, Germany, Springer, August 2015, vol. 9195, pp. 623-637, https://hal.inria.fr/hal-01239394

[24] P. CHOCRON, P. FONTAINE, C. RINGEISSEN. *A Polite Non-Disjoint Combination Method: Theories with Bridging Functions Revisited*, in "25th International Conference on Automated Deduction, CADE-25", Berlin, Germany, A. P. FELTY, A. MIDDELDORP (editors), Lecture Notes in Computer Science, Springer, August 2015, vol. 9195, pp. 419-433 [*DOI :* 10.1007/978-3-319-21401-6_29], https://hal.inria.fr/hal-01157898

[25] P. CHOCRON, P. FONTAINE, C. RINGEISSEN. *A Rewriting Approach to the Combination of Data Structures with Bridging Theories*, in "Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015", Wroclaw, Poland, C. LUTZ, S. RANISE (editors), Lecture Notes in Computer Science, Springer, September 2015, vol. 9322, pp. 275–290 [*DOI :* 10.1007/978-3-319-24246-0_17], https://hal.inria.fr/hal-01206187

[26] M. GUTHMULLER, M. QUINSON, G. CORONA. *System-level State Equality Detection for the Formal Dynamic Verification of Legacy Distributed Applications*, in "Formal Approaches to Parallel and Distributed Systems (4PAD) - Special Session of Parallel, Distributed and network-based Processing (PDP)", Turku, Finland, March 2015, https://hal.archives-ouvertes.fr/hal-01097204

[27] M. JAROSCHEK, P. F. DOBAL, P. FONTAINE. *Adapting Real Quantifier Elimination Methods for Conflict Set Computation*, in "Frontiers of Combining Systems (FroCoS)", Wroclaw, Poland, C. LUTZ, S. RANISE (editors), LNCS, Springer, 2015, vol. 9322, pp. 151-166 [*DOI :* 10.1007/978-3-319-24246-0_10], https://hal.inria.fr/hal-01240343

[28] D. MÉRY, S. RUSHIKESH, A. TARASYUK. *Integrating Domain-Based Features into Event-B: a Nose Gear Velocity Case Study*, in "Model and Data Engineering - 5th International Conference, MEDI 2015", Rhodes, Greece, L. BELLATRECHE, Y. MANOLOPOULOS (editors), springer, September 2015, vol. lncs 9344, pp. 89-102, https://hal.inria.fr/hal-01245991

[29] D. MÉRY, N. K. SINGH. *Analyzing Requirements Using Environment Modelling*, in "Digital Human Modeling - Applications in Health, Safety, Ergonomics and Risk Management: Ergonomics and Health - 6th International Conference, DHM 2015", Los Angeles, United States, V. G. DUFFY (editor), Lecture Notes in Computer Science - Held as Part of HCI International 2015, Springer, August 2015, vol. 9185, https://hal.inria.fr/hal-01245994

[30] M. QUINSON, G. OSTER. *A Teaching System To Learn Programming: the Programmer's Learning Machine*, in "ACM Conference on Innovation and Technology in Computer Science Education 2015", Vilnius, Lithuania, ACM, July 2015 [*DOI :* 10.1145/2729094.2742626], https://hal.inria.fr/hal-01238377

[31] A. REYNOLDS, J. C. BLANCHETTE. *A Decision Procedure for (Co)datatypes in SMT Solvers*, in "CADE-25", Berlin, Germany, Automated Deduction - 25th International Conference on Automated Deduction, August 1-7, 2015, Proceedings, August 2015 [*DOI :* 10.1007/978-3-319-21401-6_13], https://hal.inria.fr/hal-01212585

[32] R. SCHMIDT, U. WALDMANN. *Modal Tableau Systems with Blocking and Congruence Closure*, in "24th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods", Wroclaw, Poland, H. DE NIVELLE (editor), Lecture Notes in Computer Science, Springer, September 2015, vol. 9323, pp. 38-53 [*DOI :* 10.1007/978-3-319-24312-2_4], https://hal.inria.fr/hal-01251380

[33] T. STURM. *Subtropical Real Root Finding*, in "Proceedings of the 2015 International Symposium on Symbolic and Algebraic Computation", Bath, United Kingdom, ACM, July 2015, https://hal.inria.fr/hal-01239489

Conferences without Proceedings

[34] H. BARBOSA, P. FONTAINE. *Congruence Closure with Free Variables (Work in Progress)*, in "Quantify 2015 : 2nd International Workshop on Quantification", Berlin, Germany, 2015, https://hal.inria.fr/hal-01246036

[35] M. DUFLOT, M. QUINSON, F. MASSEGLIA, D. ROY, J. VAUBOURG, T. VIÉVILLE. *When sharing computer science with everyone also helps avoiding digital prejudices*, in "Scratch2015AMS", Amsterdam, Netherlands, August 2015, https://hal.inria.fr/hal-01154767

[36] A. REYNOLDS, J. C. BLANCHETTE, C. TINELLI. *Model Finding for Recursive Functions in SMT*, in "SMT Workshop 2015", San Francisco, United States, July 2015, https://hal.inria.fr/hal-01242509

Scientific Books (or Scientific Book chapters)

[37] C. ARECES, P. FONTAINE, S. MERZ. *Modal Satisfiability via SMT Solving*, in "Software, Services, and Systems. Essays Dedicated to Martin Wirsing on the Occasion of His Retirement from the Chair of Programming and Software Engineering", Lecture Notes in Computer Science, Springer, 2015, vol. 8950, pp. 30-45 [*DOI :* 10.1007/978-3-319-15545-6_5], https://hal.inria.fr/hal-01127966

[38] C. DUBOIS, P. MASCI, D. MÉRY. *Second International Workshop on Formal Integrated Development Environment*, EPTCS, June 2015, n^{o} 187 [*DOI :* 10.4204/EPTCS.187], https://hal.inria.fr/hal-01246691

[39] P. FONTAINE, T. STURM, U. WALDMANN. , D. WANG (editor) *Foreword to the Special Focus on Constraints and Combinations*, Mathematics in Computer Science, Springer, October 2015, vol. 9, n^{o} 3 [*DOI :* 10.1007/s11786-015-0239-8], https://hal.inria.fr/hal-01239438

[40] C. WEIDENBACH. *Automated Reasoning Building Blocks*, in "Correct System Design – Symposium in Honor of Ernst-Rüdiger Olderog", R. MEYER, A. PLATZER, H. WEHRHEIM (editors), Springer, September 2015, vol. 9360, pp. 172-188 [*DOI :* 10.1007/978-3-319-23506-6_12], https://hal.inria.fr/hal-01239428

Research Reports

[41] H. BARBOSA, P. FONTAINE. *Congruence Closure with Free Variables (Work in Progress)*, Inria Nancy - Grand Est (Villers-lès-Nancy, France), August 2015, https://hal.inria.fr/hal-01235912

Scientific Popularization

[42] M. B. ANDRIAMIARINA, D. MÉRY, N. K. SINGH. *Incremental Proof-Based Development for Resilient Distributed Systems*, in "Trustworthy Cyber-Physical Systems Engineering", Tylor and Francis Group, December 2015, https://hal.archives-ouvertes.fr/hal-01246669

[43] D. MÉRY, M. POPPLETON. *Towards An Integrated Formal Method for Verification of Liveness Properties in Distributed Systems*, in "Software and Systems Modeling (SoSyM)", December 2015, https://hal.inria.fr/hal-01245819

Other Publications

[44] M. KOSTA, T. STURM, A. DOLZMANN. *Better Answers to Real Questions*, 2015, working paper or preprint, https://hal.inria.fr/hal-01248053

[45] M. KOSTA, T. STURM. *A Generalized Framework for Virtual Substitution*, January 2015, 17 p. , preprint, https://hal.inria.fr/hal-01239431

[46] S. MERZ, H. VANZETTO. *Encoding TLA+ set theory into many-sorted first-order logic*, December 2015, working paper or preprint, https://hal.inria.fr/hal-01244627

[47] M. VOIGT, C. WEIDENBACH. *Bernays-Schönfinkel-Ramsey with Simple Bounds is NEXPTIME-complete*, January 2015, preprint, https://hal.inria.fr/hal-01239399

References in notes

[48] *Gurobi Optimizer Reference Manual*, Gurobi Optimization, Inc., 2014

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

[50] L. BACHMAIR, H. GANZINGER. *Rewrite-Based Equational Theorem Proving with Selection and Simplification*, in "Journal of Logic and Computation", 1994, vol. 4, n^{o} 3, pp. 217–247

[51] R. BACK, J. VON WRIGHT. *Refinement calculus—A systematic introduction*, Springer Verlag, 1998

[52] C. BARRETT, R. SEBASTIANI, S. A. SESHIA, C. TINELLI. *Satisfiability Modulo Theories*, in "Handbook of Satisfiability", A. BIERE, MARIJN J. H. HEULE, H. VAN MAAREN, T. WALSH (editors), Frontiers in Artificial Intelligence and Applications, IOS Press, February 2009, vol. 185, chap. 26, pp. 825-885

[53] D. DÉHARBE, P. FONTAINE, Y. GUYOT, L. VOISIN. *SMT solvers for Rodin*, in "ABZ - Third International Conference on Abstract State Machines, Alloy, B, VDM, and Z - 2012", Pisa, Italy, J. DERRICK, J. A. FITZGERALD, S. GNESI, S. KHURSHID, M. LEUSCHEL, S. REEVES, E. RICCOBENE (editors), Lecture Notes in Computer Science, Springer, 2012, vol. 7316, pp. 194-207

[54] M. EL KAHOUI, A. WEBER. *Deciding Hopf Bifurcations by Quantifier Elimination in a Software-Component Architecture*, in "Journal of Symbolic Computation", 2000, vol. 30, n^{o} 2, pp. 161-179

[55] H. ERRAMI, M. EISWIRTH, D. GRIGORIEV, W. M. SEILER, T. STURM, A. WEBER. *Efficient Methods to Compute Hopf Bifurcations in Chemical Reaction Networks Using Reaction Coordinates*, in "Computer Algebra in Scientific Computing", Berlin, V. P. GERDT, W. KOEPF, E. W. MAYR, E. V. VOROZHTSOV (editors), Lecture Notes in Computer Science, Springer, 2013, vol. 8136, pp. 88–99

[56] H. ERRAMI, W. M. SEILER, T. STURM, A. WEBER. *On Muldowney's Criteria for Polynomial Vector Fields with Constraints*, in "Proceedings of the CASC 2011", LNCS, Springer, 2011, vol. 6885, pp. 135–143

[57] K. GATERMANN, M. EISWIRTH, A. SENSSE. *Toric ideals and graph theory to analyze Hopf bifurcations in mass action systems*, in "Journal of Symbolic Computation", 2005, vol. 40, pp. 1361–1382

[58] K. GATERMANN, S. HOSTEN. *Computational Algebra for Bifurcation Theory*, in "Journal of Symbolic Computation", 2005, vol. 40, n^{o} 4–5, pp. 1180-1207

[59] E. HAIRER, S. NORSETT, G. WANNER. *Solving Ordinary Differential Equations I. Nonstiff Problems*, Series in Computational Mathematics, Springer, 1993, vol. 8

[60] J. K. HALE, H. KOCAK. *Dynamics and Bifurcations*, Texts in Applied Mathematics, Springer, 1991, vol. 3

[61] A. C. HEARN, R. SCHÖPF. *Reduce User's Manual, Free Version*, October 2014

[62] D. JOVANOVIĆ, L. DE MOURA. *Cutting to the Chase*, in "Journal of Automated Reasoning", 2013, vol. 51, n^{o} 1, pp. 79-108

[63] M. LAMOTTE-SCHUBERT, C. WEIDENBACH. *BDI: a new decidable clause class*, in "Journal of Logic and Computation", 2014, vol. 24, n^{o} 6, 28 p. , https://hal.inria.fr/hal-01098084

[64] L. LAMPORT. *Specifying Systems*, Addison-Wesley, Boston, Mass., 2002

[65] M. LEMERRE, E. OHAYON. *A Model of Parallel Deterministic Real-Time Computation*, in "Proc. 33rd IEEE Real-Time Systems Symposium (RTSS 2012)", San Juan, PR, U.S.A., IEEE Comp. Soc., 2012, pp. 273-282

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

[67] H. PEYRL, P. A. PARRILO. *Computing sum of squares decompositions with rational coefficients*, in "Theor. Comput. Sci.", 2008, vol. 409, n^{o} 2, pp. 269–281

[68] A. REYNOLDS, C. TINELLI, L. DE MOURA. *Finding conflicting instances of quantified formulas in SMT*, in "Formal Methods in Computer-Aided Design, FMCAD 2014", IEEE, 2014, pp. 195-202

[69] A. ROWSTRON, P. DRUSCHEL. *Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems*, in "IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)", Heidelberg, Germany, R. GUERRAOUI (editor), Lecture Notes in Computer Science, Springer, 2001, vol. 2218, pp. 329-350

[70] V. SOFRONIE-STOKKERMANS. *Hierarchical and modular reasoning in complex theories: The case of local theory extensions*, in "Frontiers of Combining Systems. 6th International Symposium FroCos 2007, Proceedings", Liverpool, UK, B. KONEV, F. WOLTER (editors), Lecture Notes in Computer Science, Springer, 2007, vol. 4720, pp. 47-71, Invited paper

[71] T. STURM, A. WEBER, E. O. ABDEL-RAHMAN, M. EL KAHOUI. *Investigating Algebraic and Logical Algorithms to Solve Hopf Bifurcation Problems in Algebraic Biology*, in "Mathematics in Computer Science", March 2009, vol. 2, n^{o} 3, pp. 493–515

[72] T. STURM, A. WEBER. *Investigating Generic Methods to Solve Hopf Bifurcation Problems in Algebraic Biology*, in "Proceedings of Algebraic Biology 2008, RISC, Castle of Hagenberg, Austria, July 31–August 2,", Berlin, Heidelberg, K. HORIMOTO (editor), Lecture Notes in Computer Science (LNCS), Springer-Verlag, 2008, vol. 5147, pp. 200–215

[73] B. STURMFELS. *Solving Systems of Polynomial Equations*, AMS, Providence, RI, 2002

[74] D. WANG, B. XIA. *Stability Analysis of Biological Systems with Real Solution Classification*, in "Proceedings of the ISSAC 2005", ACM Press, 2005, pp. 354-361