INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Cassis

Combining approaches for the security of infinite state systems

Lorraine

THEME SYM

Activity Report

2005

Table of contents

1. Team

Head of project-team
Michaël Rusinowitch [DR INRIA]

Vice-Head of project-team
Françoise Bellegarde [PR, Université Franche-Comté, LIFC]

Administrative assistant
Sophie Drouot [INRIA, until May 31, 2005]
Aurélie Prevost [From June 1, 2005]

Staff members
Véronique Cortier [CR, CNRS-LORIA]
Silvio Ranise [CR, INRIA-LORIA, on sabbatical from October 1, 2005]
Christophe Ringeissen [CR, INRIA-LORIA, from April 5, 2005]
Mathieu Turuani [CR, INRIA-LORIA, from January 1, 2005]

Faculty members (LORIA)
Laurent Vigneron [MC, Université Nancy 2, associated CNRS researcher until August 31, 2005]

Faculty members (Université Franche-Comté)
Fabrice Ambert [MC, until August 31, 2005]
Fabrice Bouquet [MC]
Alain Giorgetti [MC]
Pierre-Cyrille Héam [MC, on leave from February to May, 2005]
Olga Kouchnarenko [MC, on leave from January to August, 2005]
Bruno Legeard [PR]
Fabien Peureux [MC]

Post-doctoral fellows
Tarek Abbes [ATER, LORIA, until August 31, 2005]
Nikolaï Kosmatov [RNTL PROUVÉ, LIFC, from September 15, 2005]
Tomasz Truderung [RNTL PROUVÉ, LORIA, until September 30, 2005]
Bogdan Warinschi [ACI CRYPTO, LORIA, from August 30, 2005]
Calogero Zarba [ACI GECCOO, LORIA, until March 31, 2005]

Ph. D. Students
Yohan Boichut [INRIA, LIFC]
Sébastien Chemin [ANVAR, LIFC]
Najah Chridi [MENRT, LORIA, from October 1, 2005]
Jean-François Couchot [PRAG UFC, LIFC]
Frédéric Dadeau [MENRT, LIFC]
Heinrich Hoerdegen [MENRT, LORIA, from October 1, 2005]
Abdessamad Imine [ATER, LORIA]
Vincent Pretre [INTERREG, LIFC, from October 1, 2005]
Augusto Oliveira Viana da Silva [ALBAN, LORIA]
Judson Santos Santiago [INRIA, LORIA]
Duc-Khanh Tran [MENRT, LORIA, from September 1, 2005]
Eugen Zalinescu [MENRT, LORIA]

2. Overall Objectives

2.1. Background

CASSIS is a joint project between *Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA - UMR 7503)* and *Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC - FRE 2661).*

The objective of the project is to design and develop tools to verify the safety of systems with an infinite number of states. The analysis of such systems is based on a symbolic representation of sets of states in terms of formal languages or logical formulas. Safety is obtained via automatic proof, symbolic exploration of models or test generation. These validation methods are complementary. They rely on the study of accessibility problems and their reduction to constraint solving.

An originality of the project is its focus on infinite systems, parameterized or large scale, for which each technique taken separately shows its limits. This is the case for example of protocols operating on topologies of arbitrary size (ring networks), systems handling data structures of any size (sets), or whose control is infinite (automata communicating through an unbounded buffer). Ongoing or envisioned applications concern embedded software (e.g. smart cards, automotive controllers), cryptographic protocols (IKE, SET, TLS, Kerberos) designed to ensure trust in electronic transactions, and distributed systems.

The problem of validating or verifying reactive systems is crucial with respect to the increasing number of security-sensitive systems. The failure of these critical systems can have dramatic consequences since for instance they are embedded in vehicles components, or they control power stations or telecommunication networks. Beside obvious security issues the reliability of products whose destination is millions of end-users has a tremendous economical impact.

There are several approaches to system verification: automated deduction, reachability analysis or model-checking, and testing. These approaches have different advantages and drawbacks. Automated deduction can address practical verification however it remains complex to handle and requires a lot of expertise and guidance from the user. Model-checking is exhaustive but must face combinatorial explosion and becomes problematic with large-size or infinite systems. Testing is fundamental for validating requirements since it allows discovering many errors. However, it is almost never exhaustive and therefore only leads to partial solutions. Hence we believe that these approaches should not be considered as competing but complementary.

The goal of our project is to contribute to new combinations of these three verification techniques in a framework that would allow applying them in an industrial context. In particular we expect some breakthrough in the infinite-state verification domain by joint applications of deductive, model-checking and testing techniques.

2.2. Context

For verifying the security of infinite state systems we rely on

- Different ways to express the safety, reachability or liveness properties of systems, linear-time or branching-time logics, and the application of abstraction or abstract interpretation.
- Test generation techniques.
- The modeling of systems by encoding states as words, terms or trees and by representing infinite sets of states by languages. To each of these structures corresponds appropriate action families, such as transductions or rewritings.

Our goal is to apply these different approaches for ensuring the security of industrial systems by providing adequate methods and tools. In more details we aim at the following contributions (see continuous lines in Figure 1):

1. verification of abstract models derived from existing systems;
2. tests generation from the abstract model for validating the existing model;
3. cross-fertilization of the different validation techniques (deduction, model-checking, test) by taking advantage of the complementarity scopes and of their respective algorithmic contributions.

Let us mention that all these techniques comply with various development methodologies.

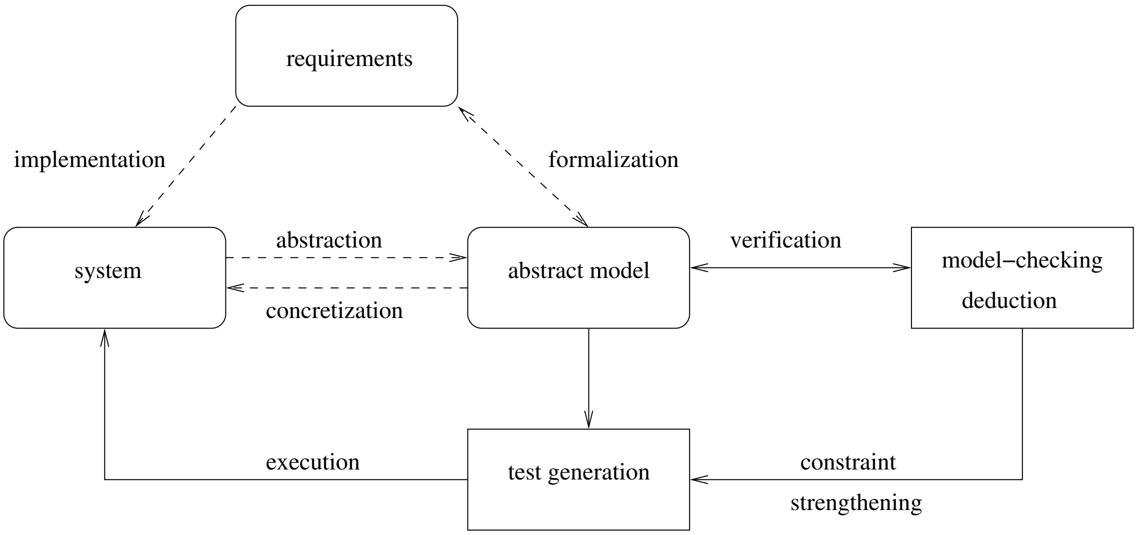

Figure 1. Software validation in CASSIS

2.3. Challenge

Verifying the safety of infinite state systems is a challenge: nowadays algorithmic techniques only apply to very specific infinite state systems. On the other hand the deductive approaches are good candidates to capture infinite system safety verification but are difficult to bring into operation and require a deep expertise. A solution consists of integrating several verification methods by combining theorem-proving and model-checking for instance.

The behavior of infinite states systems is expressed in the various models by composing or iterating actions. One of the main problems with algorithmic techniques is to compute the effect of these actions on the initial state. This computation is called *reachability analysis*. The verification of safety properties as well as the automatic generation of test cases rely heavily on the accuracy of reachability analysis.

The transverse goal is to push away the limitations on the use of formal verification techniques, to ease their applications, and to let them scale-up.

1. For properties that can be checked by reachability analysis we have proposed models based on regular languages and rational transductions. We have completed them by designing algorithms for verifying a refinement relation between two models $\mathcal{S}$ and $\mathcal{T}$ [60]. This refinement relation when satisfied preserves the safety properties and therefore allows them to be inherited. We shall investigate this approach with other representations.
2. In order to generate boundary-value functional test cases, we abstract models as constrained states. These constraints are solved by a customized solver, called CLPS. The test cases are derived in two steps [6]:

    1. partitioning of the formal model and extraction of boundary values,
    2. reachability graph exploration from constrained states in order to reach boundary values and generate state sequences (trace) as test cases with the oracle.

After the generation phase, a concretization is used to produce the test drivers [7]. Furthermore, the kernel of the engine allows one to perform specification animations in order to validate the model [63].

3. For the safety on infinite state systems we have designed automated deduction tools based on term rewriting (*SPIKE*, *daTac* [3], *haRVey*) and an extensible and modular platform for detecting flaws and potential attacks on security protocols (*AVISPA*). The tools have been built on the modeling of systems by terms and rewrite rules. Our works with other models based on regular languages of words or trees and of transducers should complement these term rewriting models.

In order to address this challenge, we rely on complementary skills within the project. We believe that each of the three techniques will benefit from concepts and algorithms designed for the two others.

3. Scientific Foundations

3.1. Introduction

Our main goal is to design techniques and to develop tools for the verification of (safety-critical) systems, such as programs or protocols. To this end, we develop a combination of techniques based on automated deduction for program verification, constraint resolution for test generation, and reachability analysis for the verification of infinite state systems.

3.2. Automated Deduction

The main goal is to prove the validity of assertions obtained from program analysis. To this end, we develop techniques and automated deduction systems based on rewriting and constraint solving. The verification of recursive data structures relies on inductive reasoning or the manipulation of equations and it also exploits some form of reasoning modulo properties of selected operators (such as associativity and/or commutativity).

Rewriting, which allows us to simplify expressions and formulae, is a key ingredient for the effectiveness of many state-of-the-art automated reasoning systems. Furthermore, a well-founded rewriting relation can be also exploited to implement reasoning by induction. This observation forms the basis of our approach to inductive reasoning, with high degree of automation and the possibility to refute false conjectures.

The constraints are the key ingredient to postpone the activity of solving complex symbolic problems only when this is really necessary. They also allow us to increase the expressivity of the specification language and to refine theorem-proving strategies. As an example of this, the handling of constraints for unification problems or for the orientation of equalities in the presence of interpreted operators (e.g. commutativity and/or associativity function symbols) will possibly yield shorter automated proofs.

Finally, decision procedures are being considered as a key ingredient for the successful application of automated reasoning systems to verification problems. A decision procedure is an algorithm capable of efficiently deciding whether formulae from certain theories (such as Presburger arithmetic, lists, arrays, and their combination) are valid or not. We develop techniques to build and combine decision procedures for the domains which are relevant to verification problems. We also perform experimental evaluation of the proposed techniques by combining propositional reasoning (implemented by means of Boolean solvers – Binary Decision Diagrams or SAT solvers) and decision procedures, and their extensions to semi-decision procedures for handling larger (possibly undecidable) fragments of first-order logic.

3.3. Synthesizing and solving set constraints

Applying constraint logic programming technology in the validation and verification area is currently an active way of research. It usually requires the design of specific solvers to deal with the description language's vocabulary. We are interested in using a solver for set constraints based on the CLPS core [2], to evaluate set-oriented formal specifications. By evaluation, we mean the encoding of the formal model into a constraint system, and the ability for the solver to verify the invariant on the current constraint graph, to propagate preconditions or guards, and to apply the substitution calculus on this graph. The constraint solver is used for animating specifications and automatically generating abstract test cases.

3.4. Reachability analysis for infinite state systems

The main objective of this task is deciding whether non-desirable states can or cannot be reached by a huge-size or an infinite system. This reachability problem is obviously crucial to guarantee the safety of critical systems. More precisely, reachability decides if a non-desirable state relates to an initial state by the transitive closure of a relation simulating the system. Unfortunately, reachability is not decidable in the general case, and even in decidable practical cases, its complexity is exponential. A solution to this problem may consist in performing reachability on a suitable abstraction of the system. Here the challenge is to choose a suitable model. When the direct approach is impossible, one can try to use upper approximations. For the (safety) properties verifiable by reachability analysis, we are studying models based on automata and words rewriting. In parallel, we develop a deductive approach, where states are defined by first order formulae with equality, to be checked by an automatic theorem prover, like *haRVey*. In both approaches, we are looking for semi-decision procedures making the verification of security properties of critical infinite systems accessible to engineers.

4. Application Domains

4.1. Verification of security protocols

Security protocols such as SET, TLS and Kerberos, are designed for establishing the confidence of electronic transactions. They rely on cryptographic primitives, the purpose of which is to ensure integrity of data, authentication or anonymity of participants, confidentiality of transactions, etc.

The experience has shown that the design of those protocols is often erroneous, even when assuming that cryptographic primitives are perfect, i.e. that an encoded message cannot be decrypted without the appropriate key. An intruder can intercept, analyze and modify the exchanged messages with very few computations and therefore, for example, generate important economic damage.

Analyzing cryptographic protocols is complex because the set of configurations to consider is very large, and can even be *infinite*: one has to consider any number of sessions, any size of messages, sessions interleaving, algebraic properties of encryption or data structures.

Our objective is to automatize as much as possible the analysis of protocols starting from their specification. This consists in designing a tool easy to use, permitting to specify a large number of protocols thanks to a standard high-level language, and permitting either to look for flaws in a given protocol or to check whether it satisfies a given property. Such a tool is essential for verifying existing protocols, but also for helping in designing new ones. For our tool to be easy to use, it has to provide a graphical interface allowing a user to do only click-button.

Our tools for verifying security protocols are available as components of the AVISPA platform. As an extension of the AVISPA specification language, we are working on a new environment called *CASRUL* for handling more general protocols like e-business protocols for example.

4.2. Automated boundary testing from formal specifications

In [7], we have presented a new approach for test generation from set-oriented formal specifications: the BZ-TT method. This method is based on Constraint Logic Programming (CLP) techniques. The goal is to test every operation of the system at every boundary state using all input boundary values of that operation. It has been validated in several industry case studies for smart card OS and application validation (GSM 11-11 standard [61] and Java Card Virtual Machine Transaction mechanism [62]) and for embedded automotive software (an automobile wind-screen wiper controller).

This test generation method can be summed up as follows: from the formal model, the system computes boundary values to create boundary states; test cases are generated by traversal of the state space with a preamble part (sequences of operations from the initial state to a boundary state), a body part (critical invocations), an identification part (observation and Oracle state computation) and a post-amble part (return path to initial or boundary state). Then, an executable test scripts file is generated using a test pattern and a table of correspondence between abstract operations (from the model) and concrete ones. This approach differs on several main points from the work of Dick, Faivre *et al*: first, using boundary goals as test objectives avoids the complete construction of the reachability graph; second, this process is fully automated and the test engineer could just drive it at the boundary value computation level or for the path computation.

The BZ-TT method is fully supported by the BZ-Testing-Tools tool-set. This environment is a set of tools dedicated to animation and test cases generation from B, Z or State-Chart formal specifications. It is based on the CLPS constraint solver, able to simulate the execution of the specification. By execution, we mean that the solver computes a so-called constrained state by applying the pre- and post-condition of operations. A constrained state is a constraint store where state variables and also input and output variables support constraints.

One orientation of the current work is to go beyond the finiteness assumption limitations by using symbolic constraint propagation during the test generation process and to extend the result to object oriented specifications.

4.3. Program debugging and verification

Catching bugs in programs is difficult and time-consuming. The effort of debugging and proving correct even small units of code can surpass the effort of programming. Bugs inserted while “programming in the small” can have dramatic consequences for the consistency of a whole software system as shown, e.g., by viruses which can spread by exploiting buffer overflows, a bug which typically arises while coding a small portion of code. To detect this kind of errors, many verification techniques have been put forward such as static analysis and software model checking.

Recently, in the program verification community, there seems to be a growing demand for more declarative approaches in order to make the results of the analysis readily available to the end user[1]. To meet this requirement, a growing number of program verification tools integrate some form of theorem proving.

The goals of our research are twofold. First, we perform theoretical investigations of various combinations of propositional and first-order satisfiability checking so to automate the theorem proving activity required to solve a large class of program analysis problems which can be encoded as first-order formulae. Second, we experimentally investigate how our techniques behave on real problems so to make program analysis more precise and scalable. Building tools capable of providing a good balance between precision and scalability is one of the crucial challenge to transfer theorem proving technology to the industrial domains.

5. Software

5.1. Protocols verification tools

Keywords: *Cryptography*, *Security Protocols*, *Verification*.

Participants: Laurent Vigneron, Yohan Boichut, Pierre-Cyrille Héam, Olga Kouchnarenko, Nikolaï Kosmatov, Michaël Rusinowitch, Judson Santos Santiago, Mathieu Turuani.

5.1.1. AVISPA

A major contribution of the European project AVISPA, is the distribution of a tool for automated verification of Internet security protocols, named *AVISPA*. It is freely available on the web [2] and supported since June 2005.

[1] See, for example, the challenge at http://research.microsoft.com/specncheck/consel_challenge.htm.

[2] http://www.avispa-project.org/

In addition to this first release, a version with a web interface is available on the web site, together with a library of protocols containing 50 IETF protocols, 4 e-business protocols and 14 other ones.

The *AVISPA* tool has been presented at the Conference on Automated Verification [21], and a tutorial has been given at the Conference on Security and Network Architectures [54]. Two of its four verification engines are developed in the CASSIS project, namely *ATSÉ* and *TA4SP*. The translator from protocol specifications to the input language of the verification tools, as well as the web-interface, and its graphical attack display have also been developed by CASSIS.

The *AVISPA* tool is still evolving, from the fundamental results recently obtained by our project. Since last release the tool has been extended for handling new security properties (such as non-repudiation), for considering algebraic operators (such as the exclusive-or and the exponentiation), for considering different levels of security over channels. The next release should be available at the end of 2005.

5.1.2. CASRUL

CASRUL is the subsystem of *AVISPA* that comprises the translator and the CASSIS verification back-ends. In the context of RNTL project PROUVÉ, we have been working on the design of a different version of the translator. Its input specification language has been defined as an evolution of the AVISPA specification language, for considering more complex protocols such as electronic purses and electronic vote systems, that have been provided by France Telecom R&D. This language is being linked to different verification tools, including those developed in the CASSIS group, *ATSÉ* and *TA4SP*.

5.1.3. ATSÉ

ATSÉ is a *Constraint Logic based Attack Searcher*. The *ATSÉ* approach to verification consists in a symbolic state exploration of the protocol execution, for a bounded number of sessions. This necessary restriction (for decidability, see [67]) allows *ATSÉ* to be correct and complete, i.e. any attack found by *ATSÉ* is a valid attack, and if no attack is found, then the protocol is secured for the given number of sessions. Each protocol step is represented by a constraint on the protocol state. These constraints are checked lazily for satisfiability, where satisfiability means reachability of the protocol state. *ATSÉ* includes now a proper handling of sets (operations and tests), choice points (for honest participants), specification of any attack states through a language for expressing fairness, non-abuse freeness, etc..., advanced protocol simplifications and optimizations (to reduce the problem complexity *before* its dynamic analysis), and protocol analysis modulo the algebraic properties of cryptographic operators. In particular, *ATSÉ* is now able to analyze protocols modulo the properties of XOR (exclusive or) or Exp (exponentiation modulo). This required to implement an optimized version of the combination algorithm due to Baader & Schulz [59] for the unification problem in disjoint unions of arbitrary theories.

5.1.4. TA4SP

TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols) is a system for verifying security protocols. The approach, initiated by T. Genet (LANDE project) and F. Klay [68], is to over-approximate or to under-approximate the intruder knowledge by using regular tree languages. This method allows us to show whether some particular states are reachable or not, and hence whether the intruder is able to know some particular terms or not. The tool admits three possible outcomes: Either the protocol is flawed (using under-approximation), or the protocol is secure (using over-approximation), or no conclusion can be drawn. A new feature has been developed to prove secrecy not only on the given scenario but also for any scenario. Indeed, by abstracting a protocol specification into a *two agents* models, the above result can be obtained if the abstracted model satisfies some conditions described in [4]. *TA4SP* includes a translator from protocol specifications to Timbuk 2.0[3] (a tree automata library developed by the LANDE project), and a package for the automatic generation of an approximation function.

5.2. Testing tools

Keywords: *Animation of Specifications, CLP, Formal Specification, Test generation*.

[3] http://www.irisa.fr/lande/genet/timbuk/

Participants: Fabrice Ambert, Fabrice Bouquet, Sébastien Chemin, Frédéric Dadeau, Bruno Legeard, Fabien Peureux.

The Testing Tools is a tool-set for animation and test generation from B, JML, Z and State-chart specifications. It consists of two components:

- **BZ-Testing-Tools**[4] – BZ-TT – is a tool-set for animation and test generation from B, Z and State-chart specifications. BZ-TT provides several testing strategies (partition analysis, cause-effect testing, boundary-value testing and domain testing), and several test model coverage criteria (multiple condition coverage, boundary coverage and transition coverage).
- **JML-Testing-Tools**[5] – JML-TT – is a framework for the symbolic animation of formal models written using JML annotations [69] embedded within Java programs. JML-TT provides a simple and efficient way to semi-automatically validate a JML specification and to check model properties such as class invariant or history constraints during the animation. This tool is used in the ACI GECCOO project[6].

5.3. haRVey

Keywords: *Automated Deduction*, *Boolean Reasoning*, *Equational Reasoning*, *Satisfiability*, *Saturation Theorem Proving*.

Participants: Jean-François Couchot, Alain Giorgetti, Silvio Ranise, Christophe Ringeissen, Duc-Khanh Tran.

haRVey[7] is a theorem prover for first-order logic with equality [66]. It works by refutation and checks whether a first-order formula is a logical consequence of a first-order theory T, axiomatized by a finite set of formulae. Recently, the capability of reasoning in the combination of T and the theory of linear arithmetic over integers has been added. The main feature of *haRVey* is its capability of behaving as a decision procedure for the problem of checking the validity of certain classes of quantifier-free formulae modulo some theories of relevance in verification such as lists, arrays, and their combinations. The system features a combination of Boolean reasoning (supplied by a BDD or a SAT solver) to efficiently handle the boolean structure of formulae and a (generalization of the) Nelson-Oppen combination method between superposition theorem proving to flexibly reason in T and an implementation of Fourier-Motzkin method for linear arithmetic. The new version of *haRVey* integrating a SAT solver has been designed and implemented by P. Fontaine (MOSEL project). *haRVey* has been especially designed to be integrated in larger verification systems. It is integrated in Barvey[8], a tool to check the consistency of B specifications. It takes a B abstract machine as input, generates proof obligations encoding the fact that the invariant is inductive, and translates them into a validity problem that *haRVey* can discharge. The tool *Why* developed by J.-C. Filliâtre (LRI, Université Paris Sud, Orsay) can generate proof obligations for *haRVey* to check the correctness of ML or C programs.

5.4. Others tools

Most of the software described in previous sections are using tools that we have developed in the past: BZ-TT uses the set constraints solver CLPS; *CASRUL* uses the theorem prover *daTac*; and *SPIKE*, our induction-based theorem prover is used in the system VOTE in collaboration with the ECOO project.

6. New Results

6.1. Automated Deduction

Keywords: *Consistency*, *Decision Procedure*, *Proof*, *Satisfiability*, *Tree Automata*.

[4] http://lifc.univ-fcomte.fr/~bztt
[5] http://lifc.univ-fcomte.fr/~jmltt
[6] http://geccoo.lri.fr
[7] http://www.loria.fr/equipes/cassis/softwares/haRVey/
[8] http://lifc.univ-fcomte.fr/~couchot/soft/barvey/

6.1.1. Decision procedures and their extensions

Participants: Silvio Ranise, Christophe Ringeissen, Duc-Khanh Tran, Calogero Zarba.

Decision procedures are ubiquitous to automatize verification systems. We study general techniques to build decision procedures, namely rewriting techniques for some axiomatizable theories and combining techniques for unions of arbitrary theories.

Most computer programs store elements of a given nature into container-based data structures such as lists, arrays, sets, and multisets. To verify the correctness of these programs, one needs to combine a theory S modeling the data structure with a theory T modeling the elements. This combination can be achieved using the classic Nelson-Oppen method only if both S and T are stably infinite. In [42], [49], our goal is to relax the stable infiniteness requirement. To achieve this goal, we first consider as a case-study the theory of lists, by showing how to combine it with an arbitrary theory of elements [42] (joint work with Pascal Fontaine, MOSEL project). Then, we introduce in [49] the notion of *polite* theories, and we show that natural examples of polite theories include those modeling data structures such as lists, arrays, sets, and multisets. Furthemore, we provide a method that is able to combine a polite theory S with any theory T of the elements, regardless of whether T is stably infinite or not. These results generalize to many-sorted logic those recently obtained by Tinelli and Zarba concerning the combination of *shiny* theories with nonstably infinite theories in one-sorted logic.

The rewriting approach to T-satisfiability is based on establishing termination of a superposition-based calculus on the satisfiability problem. Extending previous such results, including the quantifier-free theory of equality and the theory of arrays with or without extensionality [1], we prove in [22], [23] termination for the theories of records with or without extensionality, integer offsets and integer offsets modulo. A general theorem for termination on combinations of theories, that covers any combination of the theories above, is also given.

In [44], we study, in collaboration with Hélène Kirchner (PROTHEO project), how to efficiently combine superposition-based satisfiability procedures with arbitrary satisfiability procedures for theories, for which the superposition calculus may not apply (e.g., for various decidable fragments of Arithmetic). Our starting point is the Nelson-Oppen method, where satisfiability procedures cooperate by exchanging entailed (disjunction of) equalities between variables. We show that the superposition calculus deduces sufficiently many such equalities for convex theories (e.g., the theory of equality and the theory of lists) and disjunction of equalities for non-convex theories (e.g., the theory of arrays) to guarantee the completeness of the combination method. Experimental results on proof obligations extracted from the certification of auto-generated aerospace software [48] confirm the efficiency of the approach. Finally, we show how to make satisfiability procedures built by superposition both incremental and resettable by using a hierarchic variant of the Nelson-Oppen method.

Many approaches to the decision of quantifier free formulae with respect to a background theory T – also known as Satisfiability Modulo Theory, or SMT – rely on the integration between an enumerator of truth assignments and a decision procedure for conjunction of literals in T. When the background theory is the combination $T_1 \cup T_2$ of two simpler theories, the approach is typically instantiated by means of a combination method (e.g. Nelson-Oppen, Shostak). In [33] we propose a new approach to SMT($T_1 \cup T_2$), where the enumerator of truth assignments is integrated with two decision procedures for T_1 and for T_2, which act independently from each other.

The impact of various pre-processing transformations of ground formulae on the performances of Satisfiability Modulo Theory tools is briefly discussed in our preliminary experiences reported in [26].

6.1.2. Tree automata and their extensions

Participants: Michaël Rusinowitch, Laurent Vigneron.

Tree automata are widely used in verification. We have investigated recently several extensions of tree automata for handling equality of subtrees. Dag automata operate on dags instead of on trees. We have shown with S. Anatharaman and P. Narendran that the class of dag automata is not closed under complementation, dag automata are not determinizable, their membership problem is NP-complete, their universality problem is undecidable, and their emptiness problem is NP-complete [20], [12]. We have also considered classes of tree automata combining automata with equality test and automata modulo equational theories with F. Jacquemard (SECSI project) [57]. These tree automata are obtained by extending their standard Horn clause representations with equational conditions and rewrite systems. We show in particular that a generalized membership problem (embedding the emptiness problem) is decidable by proving that the saturation of tree automata presentations with suitable paramodulation strategies terminates. These tree automata classes can be applied to the reachability problem for a fragment of pi-calculus.

6.1.3. Verification of copies convergence in distributed groupware systems

Participants: Abdessamad Imine, Michaël Rusinowitch.

Ensuring the convergence of replicated data in collaborative editing systems is still a challenging issue, especially when the data has a linear structure such as a text or an ordered XML tree. We are interested in data convergence using the Operational Transformation (OT) approach. Based on our algebraic framework for designing OT algorithms [18], [47], and with the help of a theorem-prover we have shown that all previously published OT algorithms are incorrect. Then, we have proposed [43] in collaboration with ECOO project a new OT algorithm for which we have provided a complete machine-checked proof. Currently, we plan to build a collaborative editing system based on this OT algorithm and to extend it to a peer-to-peer context where the number of users is arbitrary.

6.2. Security protocol verification

Keywords: *Exclusive-Or*, *Exponentiation*, *Protocol*, *Security*, *Verification*.

Cryptographic protocols are successfully analyzed using formal methods and many techniques have appeared in the litterature. We describe these techniques in a first survey [15]. However, formal approaches usually consider the encryption schemes as black boxes and assume that an adversary cannot learn anything from an encrypted message except if he has the key. Such an assumption is too strong in general since some attacks exploit in a clever way the interaction between protocol rules and properties of cryptographic operators. In [16], we give a list of some relevant algebraic properties of cryptographic operators, and for each of them, we provide examples of protocols or attacks using these properties. We also give an overview of the existing methods in formal approaches for analyzing cryptographic protocols under equational theories.

6.2.1. Extension of the Dolev-Yao model

Participants: Véronique Cortier, Michaël Rusinowitch, Tomasz Truderung, Mathieu Turuani, Eugen Zalinescu.

Formal methods have proved to be very useful for analyzing cryptographic protocols. However, most existing techniques apply to the case of abstract encryption schemes and pairing. In [38], we have considered more complex, less studied cryptographic primitives like CBC encryption and blind signatures. This led us to introduce a new fragment of Horn clauses. We have shown decidability of this fragment using a combination of several resolution strategies. As a consequence, we have obtained a new decidability result for a class of cryptographic protocols (with an unbounded number of sessions and a bounded number of nonces) that may use for example CBC encryption and blind signatures. We have applied this result to fix the Needham-Schroeder symmetric key authentication protocol, which is known to be flawed when CBC mode is used.

Decidability results under general equational theories have been rare. We have focused here on decidability of ground deducibility, *i.e.* checking whether a message can be derived by an intruder, and static equivalence. Indeed, deduction does not always suffice for expressing the knowledge of an attacker; static equivalence enables to capture the comparison power of an attacker: two sequences of messages are statically equivalent if they verify the same equalities, for arbitrary tests. In [19], we have pursued the study of these two relations. We have established general decidability theorems for both. These theorems require only loose, abstract conditions on the equational theory for messages. They subsume previous results for a syntactically defined class of theories that allows basic equations for functions such as encryption, decryption, and digital signatures. They also apply to many other useful theories, for example with blind digital signatures, homomorphic encryption, XOR, and other associative-commutative functions.

Most of the decision procedures for symbolic analysis of protocols are limited to a fixed set of algebraic operators associated with a fixed intruder theory. Examples of such sets of operators comprise XOR, abelian groups, abstract encryption/decryption. We have obtained recently, in collaboration with Y. Chevalier (IRIT) an algorithm for combining decision procedures for arbitrary intruder theories with disjoint sets of operators, provided that solvability of ordered intruder constraints, a slight generalization of intruder constraints, can be decided in each theory [34], [35]. This is the case for most of the intruder theories for which a decision procedure has been given. In particular our result allows us to decide trace-based security properties of protocols that employ any combination of the above mentioned operators with a bounded number of sessions, even for the case of active intruders.

Another useful extension of the model is to consider the case where the initial intruder knowledge is given by a regular tree language or where the protocol messages are restricted to belong to some regular tree language. In both cases decidability results in the bounded number of sessions cases can be derived [51]. Finally in order to allow agents to compare and store arbitrary messages, a new formalism, called *selecting theories* has been introduced for modeling recursive protocols, where agents, in each protocol step, are able to send an unbounded number of messages [52].

6.2.2. Soundness of the Dolev-Yao model

Participants: Véronique Cortier, Mathieu Turuani, Bogdan Warinschi.

Since the 1980s, two approaches have been developed for analyzing security protocols. One of the approaches relies on a computational model that considers issues of complexity and probability. This approach captures a strong notion of security, guaranteed against all probabilistic polynomial-time attacks. The other approach relies on a symbolic model of protocol executions in which cryptographic primitives are treated as black boxes. Since the seminal work of Dolev and Yao, it has been realized that this latter approach enables significantly simpler and often automated proofs. However, the guarantees that it offers have been quite unclear.

We have shown [39] that it is possible to obtain the best of both worlds in the case of public encryption: fully automated proofs and strong, clear security guarantees. Specifically, for the case of protocols that use signatures and asymmetric encryption, we have established that symbolic integrity and secrecy proofs are sound with respect to the computational model. The main new challenges concern secrecy properties for which we obtain the first soundness result for the case of active adversaries. Our proofs are carried out using *CASRUL*, a fully automated tool. More generally, we are working on soundness results for the Dolev-Yao model and on the link between formal and cryptographic models.

We have also considered this correspondence problem in a more general framework, but for passive attacker. In contrast to other works, we did not consider a fixed set of primitives but aimed at results for an arbitrary equational theory. We defined a framework for comparing a cryptographic implementation and its idealization *w.r.t.* various security notions. In particular, we concentrated on the computational soundness of static equivalence, a standard tool in cryptographic π-calculi. We have presented a soundness criterion, which for many theories is not only sufficient but also necessary. Finally, we have established new soundness results for the Exclusive Or, as well as a theory of ciphers and lists. These results have appeared in [25].

Following a different method, we have also studied how to directly prove properties on cryptographic protocols without any restriction, and with respect to an active, polynomial-time adversary [41]. This approach relies on a cryptographically sound formal logic, which does not require explicit reasoning about probability, asymptotic complexity, or the actions of a malicious intruder. That is, while the soundness of the logic w.r.t any active polynomial time adversary is proved with such reasoning and relies on strong cryptographic properties of the operators (like IND-CCA2 for encryption), it is enough to logically combine the axioms and rules provided by the logic to build a security proof of a protocol against such “real” adversaries. A proof in this logic is necessarily done by hand, but it can be automatically validated in Isabelle proof assistant with an appropriate encoding of the logic [53].

6.2.3. Security properties

Participants: Najah Chridi, Judson Santos Santiago, Mathieu Turuani, Laurent Vigneron.

While security issues such as secrecy and authentication have been studied intensively, interest in non-repudiation protocols was raised only in recent years. Non-repudiation services must ensure that when two parties exchange informations over a network, neither one nor the other can deny having participated in this communication. Consequently a non-repudiation protocol has to generate evidences of participation that are exhibited in the case of a dispute.

We have given [50] a formal description of non-repudiation services, and illustrated it with the Fair Zhou-Gollmann protocol. We have also defined non-repudiation properties within the *AVISPA* tool and explained how they can be automatically verified.

Verifying security properties of group key agreement protocols raises challenging problems. Apart from their unbounded number of participants, group protocols have to satisfy specific non standard security properties. These properties, generally related to dynamic membership in groups, are difficult to express and are usually not formally specified in the literature. We have investigated [58], [37] the modeling of group protocols and more generally key contributing ones. This analysis has helped us to detect several attack types on protocols, such as A-GDH.2, SA-GDH.2, Asokan-Ginzboorg and Bresson-Chevassaut-Essiari-Pointcheval.

We have proposed in [24] a general method for reasoning about contract-signing protocols using a specialized protocol logic, allowing us to prove properties of the Asokan-Shoup-Waidner (ASW) and the Garay-Jacobson-MacKenzie (GJM) protocols. These protocols validate specific properties like fairness, abuse freeness, accountability, etc.. Our method improves upon previous analysis techniques. First, the method is compositional, i.e. the security guarantees are obtained by combining independent proofs for each subprotocols. Second, the formal proofs are carried out in a “template” form, i.e. we can prove the security of an abstract protocol which can be instantiated by ASW, GJM, or any other protocol with the same global design. Finally, our result holds even when an unbounded number of sessions are executed, and against an active (Dolev-Yao) adversary. Finally, the fairness proof for ASW with this approach has been validated in Isabelle [53].

6.2.4. Intruder knowledge approximation

Participants: Yohan Boichut, Pierre-Cyrille Héam, Olga Kouchnarenko.

When the number of sessions is unbounded, the security problem of cryptographic protocols is undecidable. Since the security of real protocols such as internet-protocols is crucial, it is important to develop methods to handle this problem. For that purpose, we have investigated the semi-algorithmic method introduced by T. Genet and F. Klay. This method allows one to show that some states are unreachable, and hence that the intruder will never be able to know certain terms. Regular tree-languages are employed to model effectively the knowledge that the intruder might have acquired from previous sessions. In our approach, tree automaton languages either over-approximate or under-approximate intruder’s knowledge. We can prove that a protocol is secure with respect to secrecy when the over-approximation of the set of reachable terms contains no term representing a violation of secrecy. Otherwise, no conclusion can be drawn.

In order to accelerate the approximation function generation, we have developed a new coarser abstraction function following the approach in [4]. We have applied the assumptions given in [4] to verify safety properties to our abstraction function. Considering only two agents allows us to speed up the over-approximation function generation. This way, we are able to automatically verify more protocols. We have also developed an automatically generated under-approximation function for intruder knowledge. With under-approximation we can show that a protocol is flawed, for instance LPD-MSR protocol. Our next goal is to provide an automatic reconstruction of these attacks.

These new approximation techniques have been implemented in *TA4SP*. The verification is totally automatic for protocols specified in a high level specification language. We have successfully verified nine protocols provided by Siemens; one protocol is proved to be flawed with respect to secrecy.

6.3. Reachability analysis

Keywords: *Formal Specifications*, *Model-based Testing*, *Parametric Systems*, *Reachability*, *Regular Languages*, *Test Case Generation*.

6.3.1. Reachability with generalized substitutions

Participants: Françoise Bellegarde, Jean-François Couchot, Alain Giorgetti, Nikolaï Kosmatov, Silvio Ranise.

Deductive reachability analysis denotes the restriction of model checking with a rich assertional language to safety properties expressed as nonreachability of some critical states. This deductive reachability analysis of infinite state systems arises four questions: how data are structured, how transitions are defined, which logic is used by the (semi-)algorithms to express their evolution conditions and which prover is called for discharging these proof obligations.

This deductive approach has been applied for distributed systems composed of any number of similar processes communicating by rendez-vous and broadcast, for safety properties including mutual exclusion and cache coherence [40]. The global system behavior is specified in a language of generalized substitutions on array data structures, state configurations (i.e. assertions) are expressed in a many-sorted first-order language with equality.

This context has been shown adequate for deductive model-checking in the sense that backward reaching an initial state or the fixpoint is decidable under some natural conditions. These conditions are satisfied for a large class of realistic distributed protocols including most of the examples from the litterature of parameterized verification. An implementation using the *haRVey* prover gives good experimental results in time, comparable to or better than those obtained with other approaches.

6.3.2. Reachability computation with regular languages

Participants: Françoise Bellegarde, Pierre-Cyrille Héam.

The *regular model checking* techniques are based on regular languages for reachability analysis, where states are represented by finite automata or regular expressions and actions are modelized by transducers or rewrite rules on words.

In our approach, we obtain the language L' reachable from a language L by using the transitive closure of a semi-commutation relation, *i.e.*, a finite union of rewrite rules of the form $ab \to ba$.

Following previous work, algorithms developed in [65], [64][17] were implemented in a prototype GSAT (for Generic Simple Automata Toolkit). The automata can be of any kind — the more general purpose of this tool being to provide to the scientific community a generic tool for writing algorithms which use finite automata.

6.3.3. Test case and test driver generation from a formal model

Participants: Fabrice Ambert, Fabrice Bouquet, Sébastien Chemin, Frédéric Dadeau, Nikolaï Kosmatov, Bruno Legeard, Fabien Peureux.

The need to offer better methods and tools for functional black-box testing of large scale systems has raised a large amount of research on generating tests from formal specifications. The BZ-TT approach is based on an original method of boundary-value extraction and preamble computation based on a customized constraint logic programming technology. This method has been validated on several real-size industrial applications. The new research directions that we follow concern various research challenges.

We try to improve the quality of a specification, with two approaches. The first one is to use BZ-TT technologies to help in the design of specifications [28]. The second one is to use results and works developed for B to verify JML specifications [27].

On JML specifications, we improve the mechanisms of the constraint solver to animate object specification [29]. We improve the animation process [31] and we present the JML-TT tool implementing the results reported in [30]. We integrate the traceability of the requirements from the model to test sequences [32].

To take into account sequences in our test generation tools, a constraint solver for sequences is necessary. A solver prototype for sequences in Constraint Handling Rules language (CHR) has been developed and its applications to software validation and test generation has been studied [46], [45].

7. Contracts and Grants with Industry

7.1. RNTL

RNTL project PROUVÉ[9] — *"Protocoles cryptographiques: Outils de Vérification automatique"*, duration: 3 years, started on November 2003. The goal of this project is the automatic verification of cryptographic protocols, for a large class of security properties and algebraic properties of the cryptographic primitives. There are five partners: CRIL Technology Systèmes Avancés, France Telecom R&D, INRIA Lorraine, LSV (ENS de Cachan), Verimag (Grenoble).

RNTL project DANOCOPS — *"Détection Automatique de NOn-COnformités d'un Programme vis à vis de ses Spécifications"*, duration: 39 months, started on 1st January 2004. The goal of this project is to confront specification and program to find no-conformity. We propose to use an abstract representation of specification and source program, with constraints. The are five partners, two industrials: Thales division Systèmes Aéroportés, Axlog (SS2I), and three academics: I3S/Nice, LSR/Grenoble and LIFC/Besancon. The local coordinator is F. Bouquet.

RNTL project POSÉ — *Security policies conformance testing for embedded systems*, duration: 2 years, started in december 2005. The objective is to provide automated tools for generating tests of security policies conformance of embedded systems. There are four partners: LEIRIOS, AXALTO, SILICOMP AQ, and IMAG/LSR. The local coordinator is F. Bouquet.

7.2. Research Result Transfer

The BZ-Testing-Tools technology has been transfered to Leirios Technologie, at the end of 2004. The partnership between the Cassis project and the R&D Leirios Department, located at the TEMIS Scientific and Industrial area at Besançon, will be continued through projects (national and international call of work) or with a new transfer protocol. According to the law of innovation, F. Ambert, F. Bouquet, B. Legeard and F. Peureux are scientific consultants of Leirios Technology.

7.3. IST AVISPA

AVISPA[10] is a shared-cost RTD (FET open) project, funded by the European Commission under the Information Society Technologies Program operating within the Fifth Framework Program, started on January 1st, 2003, and ended on June, 2005. The participants are: Mechanized Reasoning Group at DIST, Università di Genova (Genova, Italy), Cassis project at INRIA, Information Security Group at ETHZ (Zürich, Switzerland) and Siemens AG (Munich, Germany).

AVISPA aims at developing a push-button, industrial-strength technology for the analysis of large-scale Internet security-sensitive protocols and applications. This technology will speed up the development of the next generation of network protocols, improve their security, and therefore increase the public acceptance of advanced, distributed IT applications based on them. A central aim of the project is to integrate this technology into a robust automated tool, tuned on practical, large-scale problems collected from IETF drafts, and migrated to standardization bodies.

7.4. INTERREG

INTERREG Test-UML — In the European InterReg III project (Suisse - Franche-Comté), LIFC is a member of the TestUML project with the École Polytechnique Fédérale de Lausanne - EPFL - concerning Test

[9] http://www.lsv.ens-cachan.fr/prouve/
[10] http://www.avispa-project.org/

generation from UML/OCL formal models. The duration of the project is 2 years and it was started in November 2002.

INTERREG VALID — We are working with the university of Geneve, Leirios Technologies and Centre des Technologies de l'Information - État de Genève. The project concerns the test generation for the web services. The duration of the project is 18 months and it was started in July 2005.

8. Other Grants and Activities

8.1. International grants

Project INRIA-CNPq (Brazil), DA CAPO — *Automated deduction for the verification of specifications and programs*. It is a project on the development of proof systems (like *haRVey*) for the verification of specifications and software components. The coordinators of this project are David Déharbe (UFRN Natal, Brazil) and Christophe Ringeissen. On the french side, DA CAPO also involves the PROTHEO project.

Project INRIA-Tunisian Universities — *"Vérification et analyse de la sécurité et de la sûreté des systèmes critiques"*. The coordinators of this project are Nejib Ben Hadj-Alouane (ENSI Tunis, University Manouba) and Michael Rusinowitch. On the french side, this project also involves the laboratory LAG (Grenoble).

8.2. National grants

ACI GECCOO—*"Génération de code certifié pour des applications orientées objet (Spécification, raffinement, preuve et détection d'erreurs)"*, duration: 3 years, started on July 2003.

This project aims at developing methods and tools for the design of object-oriented systems that require a high degree of security. In particular, the project focuses on the design of smart card applications, written in a subset of Java (like JavaCard), annotated with JML specifications. Partners are: TFC (LIFC), EVEREST (INRIA Sophia-Antipolis), PROVAL (INRIA Futurs & LRI), VASCO (LSR). The local coordinators are F. Bellegarde (LIFC) and S. Ranise (Nancy).

ACI V3F—*Validation & Verification of programs with floating-point numbers*, duration: 3 years, started on October 2003. Cassis (B. Legeard) is the principal coordinator.

The goal of this project is to provide tools to support the verification and validation process of programs with floating-point numbers. The underlying technology is based on constraint solving. Partners are: I3S-INRIA Sophia Antipolis, IRISA-INRIA Vertecs & Lande, CEA-LIST.

ACI EDEMOI—*Formal Modeling and Verification of Airport Security*, duration: 3 years, started on October 2003.

The EDEMOI project aims at defining an approach for the construction and analysis of a precise reference document that models and structures current standards and associated recommendations. The exploitation of this model by the civil aviation authorities will improve airport security. Partners are: LSR-IMAG, CEDRIC-CNAM, ONERA Toulouse, GET ENST Paris.

ACI SATIN[11] —*Security Analysis for Trusted Infrastructures and Network protocols*, duration: 3 years, started on July 2004. Cassis (M. Rusinowitch) is the principal coordinator.

The SATIN project aims at working on formal analysis and design of secure distributed systems, by taking advantage of the recent advances in algebraic modeling techniques. Partners are: CEA-DAM, France Telecom R&D, LANDE project - IRISA, VPS team, LIFO.

ACI Jeunes Chercheurs CRYPTO[12] —*"Lien entre la cryptanalyse et l'étude logique des protocoles cryptographiques"*, duration: 3 years, started on September 2004.

The CRYPTO project aims at establishing a link between the formal and the computational approaches for cryptographic protocols.

[11] http://lifc.univ-fcomte.fr/~heampc/SATIN

[12] http://www.loria.fr/~cortier/aci.html

ARA SSIA FormaCrypt—*Formal proofs and probabilistic semantics in cryptography*, duration: 3 years, started in automn 2005.

The verification of cryptographic protocols is a very active research area. Most works on this topic use either the computational approach, in which messages are bitstrings, or the formal approach, in which messages are terms. The computational approach is more realistic but more difficult to automate. The FormaCrypt project aims at briding together these orthogonal approaches in order to get the best of the two worlds. Partners are: Liens (coordinator), SECSI project - LSV, Cachan.

ARA SSIA COPS—*Composition Of Policies and Services*, duration: 3 years, started in december 2005.

The aim is to build technologies enabling the security analysis of web services that take into account the potential flaws at communication level, at the access policy level or at the interface between communications and access policy. Partners are: IRIT Toulouse, LIM Marseille, Microsoft R&D.

ARA SSIA ARROWS—*Safe Pointer-Based Data Structures: A Declarative Approach to their Specification and Analysis*, duration: 3 years, started in automn 2005.

Programming with pointers is quite a powerful and widely used technique to build many software systems with limited resources such as embedded systems or programs requiring recursive data structures. The goal of this project is to develop new specification languages for programs manipulating pointers which are sufficiently precise to express many interesting properties and, at the same time, support automatic analyses. Partners are: CAPP-LEIBNIZ Grenoble (coordinator), LILaC-Irit Toulouse. The local coordinator is S. Ranise.

QSL VALDA2—*Automated Software Verification using Automated Deduction*, duration: 2 years, started in 2005. With the action, we are working in the *Pôle de Recherche Scientifique et Technologique Intelligence Logicielle* within the theme *Qualité et sûreté des logiciels et systèmes informatiques*, funded by the *Contrat de Plan État-Région Lorraine 2000-2006*.

8.3. International collaborations

- In the continuation of a PAI PROCOPE, we are working on the combination of automata-theoretic and rewriting techniques for the analysis of cryptographic protocols with the group of professor Thomas Wilke, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel.
- In the area of automated test generation from a formal model, we have an active collaboration with Dr Mark Utting from the Formal Method group from the University of Waikato.[13] This cooperation is supported by the France-New-Zealand scientific program.

8.4. Individual involvement

F. Bellegarde: director of the research team *Techniques Formelles et à Contraintes (TFC)* of the *Laboratoire d'informatique de Franche Comté (LIFC)*, board member of the *LIFC*, Editorial committee member of *Techniques et Science Informatique (TSI)*.

F. Bouquet: in charge of the Mobilization area (9 research projects, with 46 researchers and 8 laboratories) in ISTI Institute;[14] coordinator of Tools session of AFADL'04; coordinator of INTERREG VALID.

V. Cortier: coordinator of the ACI Jeunes Chercheurs CRYPTO; PC member of *Mathematical Foundations of Computer Science conference (MFCS 2005)*; co-organizer of Workshop on the Link between Formal and Computational Models in June 2005, Paris;

O. Kouchnarenko: co-chair of the "CSE 27" of the University of Franche-Comté; PC member of "*Approches Formelles dans l'Assistance au Développement de Logiciels*", AFADL'06 and of "*Formal Specification and Development in B*".

B. Legeard: member of the Scientific council of the University of Franche-Comté. Coordinator of the ACI V3F.

[13] http://www.cs.waikato.ac.nz/Research/fm/index.html

[14] http://www.isti.info

S. Ranise: trustee of the project CALCULEMUS (Systems for Integrated Computation and Deduction); coordinator (with Cesare Tinelli) of the Satisfiability Modulo Theories Library (SMT-LIB) initiative; PC member of the workshops “Pragmatics of Decision Procedures in Automated Reasoning (PDPAR) 2005", co-located with CAV’05, “Calculemus 2005" co-located with FM’05, “Empirically Successful Classical Automated Reasoning (ESCAR) 2005", co-located with CADE’05; member of the "équipe pedagogique du Master Informatique (recherche)".

C. Ringeissen: PC member of “Frontiers of Combining Systems (FroCoS) 2005”; co-editor (with Alessandro Armando) of a special issue of Information & Computation on “Combining Logical Systems”; correspondent for the european projects at LORIA.

M. Rusinowitch: member of the IFIP Working Group 1.6 (Rewriting); member of the scientific committee of the CRIL (CNRS, Computer Science Laboratory of Lens); member of the “World Class” evaluation committee for Security sector of France Telecom R&D; member of COST-GTAP prospective committee at INRIA; coordinator of the project ACI Sécurité SATIN. PC member of Rencontres Sécurité et Architecture Réseaux 2005, 2006 and workshops FTP, ARSPA, CPSec, Crisis; PC member of IEEE Symposium on Security and Privacy 2006, IEEE Computer Security Foundation Workshop 2006, Symposium on Functional and Logic Programming 2006; invited editor of Journal of Automated Reasoning.

L. Vigneron: member of the scientific council of Université Nancy 2; member of the FTP steering committee; PC chair of the 19th International Workshop on Unification; secretary of the IFIP Working Group 1.6; web master of the site *Rewriting Home Page*, of the RTA conference site, and of the web page for the IFIP Working Group 1.6.

We are involved in several lectures of the “Master Informatique” of the universities of Nancy. V. Cortier is in charge of the lecture on *Theory of the security*, S. Ranise and C. Ringeissen are in charge of the lecture on *Decision procedures and program verification*.

8.5. Visits of foreign researchers

Adel Bouhoula (SupCom Tunis) visited our group (Nancy) in July 2005. The subject of the collaboration was intrusion detection.

Ralf Küsters (University of Kiel, Germany) visited our group (Nancy) one week in July 2005. The subject of the collaboration was about fairness properties for contract-signature protocols and strong secrecy in computational models.

Ralf Küsters, *Thomas Wilke* and *Detlef Kähler* visited our group (Nancy) from February 28 to March 4. We worked together on protocol analysis.

Mark Utting (University of Waikato, New Zealand) visited our group (LIFC) from September 2004 to February 2005 as associated CNRS researcher.

8.6. Visits of team members

V. Cortier visited Ralf Küsters at the University of Kiel, Germany, during one week in June 2005. The subject of the collaboration was about fairness properties for contract-signature protocols.

9. Dissemination

9.1. Habilitation thesis

F. Bouquet has defended his habilitation thesis entitled “*Sémantique interprétative de spécifications formelles en animation symbolique et génération de tests à partir de modèles*", on June 24th, 2005.

9.2. Awards

N. Chridi has received the best paper award of the *1st CRiSiS conference*, on October 2005.

9.3. Committees

F. Bouquet is referee for the Ph. D. thesis of Olivier Maury, University J. Fourier - Grenoble, November 2005; member of the Ph. D. thesis committee of Emmanuel Mory (Montbéliard), November, 2005.

V. Cortier is member of the SPECIF committee to award the best Ph. D. dissertations in theoretical computer science.

O. Kouchnarenko is the supervisor of the Ph. D. thesis by A. Lanoix, entitled “*Systèmes à composants synchronisés : contribution à la vérification compositionnelle du raffinement et des propriétés*”, defended on August 31st 2005.

M. Rusinowitch is referee for the habilitation thesis of Ralf Treinen (Cachan) and for the Ph. D. theses of Olivier Ponsini (Nice) and Moussa Demba (Tunis), member of the Ph. D. thesis committee of Felipe Luna (Nice) and of the habilitation thesis of Jean-Marc Talbot (Lille).

9.4. Seminars, workshops, and conferences

Besides conference talks mentioned in the publication list, we have given the following talks.

Y. BOICHUT, *TA4SP : outil pour la vérification de protocoles de sécurité*, CEA/DIF (Bruyère le Châtel), April 13th, 2005.

Y. BOICHUT, *The AVISPA Tool*, talk at the Spring School on Security, Centre International de Rencontres Mathématique (Marseille), April 27th, 2005.

F. BOUQUET, *Génération automatique de tests à partir de modèle formelle*, seminar at LIFL (Lille), December 1st, 2005.

V. CORTIER, *Verification of Cryptographic Protocols II*, talk at the Spring School on Security, Centre International de Rencontres Mathématique (Marseille), April 26th, 2005.

V. CORTIER, *La cryptographie : comment sécuriser nos communications ?*, scientific popularization conference at “les Journées du CNRS", Marseille, May 26th, 2005.

V. CORTIER, *Computationally Sound Security Proof using Formal Models*, seminar at the University of Kiel, Germany, June 3rd, 2005.

V. CORTIER, *Computationally Sound, Automated Proofs for Security Protocols*, workshop in Paris, June 23rd, 2005.

V. CORTIER, *Verification of cryptographic protocols: techniques, tools and link to cryptanalysis*, talk at the French-Japanese Symposium on Security, Tokyo, Japan, September 6th, 2005.

V. CORTIER, *Computationally Sound Security Proof using Formal Models*, seminar at the Security day of IRIT, Toulouse, September 29th, 2005.

V. CORTIER, *Relations between abstract and probalistic models of cryptographic protocols*, talk at the Nancy-Saarbruecken workshop, LORIA, October 14th, 2005.

S. RANISE, invited talk, 4th Annual Formal Equivalence and Assertion Based Verification Workshop, Madonna di Campiglio, Italy, July 2005.

S. RANISE, C. RINGEISSEN, *Building, Combining, and Integrating Decision Procedures for Software Verification*, tutorial (in collaboration with D. DÉHARBE), 3rd IEEE International Conference on Software Engineering and Formal Methods (SEFM), Koblenz, September 5th, 2005.

S. RANISE, invited talk, workshop *Decision Procedures* organized by Byron Cook in Microsoft Research, Cambridge, UK, September 2005.

S. RANISE, invited talk, onzième réunion du groupe de travail LAC du GDR-ALP, *CALCULS SYMBOLIQUES : calcul formel, réécriture, logique et preuve*, Grenoble, December 2005.

M. RUSINOWITCH, talk on security protocols, INRIA scientific committee, March 18th, 2005.

M. TURUANI, *Probabilistic polynomial-time semantics for a protocol security logic*, workshop in Paris, June 23rd, 2005.

10. Bibliography

Major publications by the team in recent years

[1] A. ARMANDO, S. RANISE, M. RUSINOWITCH. *A Rewriting Approach to Satisfiability Procedures*, in "Journal of Information and Computation — Special Issue on Rewriting Techniques and Applications (RTA'01)", vol. 183, n^o 2, June 2003, p. 140–164.

[2] F. BOUQUET, B. LEGEARD, F. PEUREUX. *CLPS-B: A Constraint Solver to Animate a B Specification*, in "International Journal of Software Tools for Technology Transfer, STTT", vol. 6, n^o 2, August 2004, p. 143–157.

[3] Y. CHEVALIER, L. VIGNERON. *Strategy for Verifying Security Protocols with Unbounded Message Size*, in "Journal of Automated Software Engineering", vol. 11, n^o 2, April 2004, p. 141-166.

[4] H. COMON-LUNDH, V. CORTIER. *Security properties: two agents are sufficient*, in "Science of Computer Programming", vol. 50, n^o 1-3, March 2004, p. 51-71, http://www.loria.fr/~cortier/Papiers/ComonCortierSCP03.ps.

[5] F. JACQUEMARD, M. RUSINOWITCH, L. VIGNERON. *Compiling and Verifying Security Protocols*, in "Logic for Programming and Automated Reasoning (LPAR'00), Reunion Island, France", A. VORONKOV, M. PARIGOT (editors). , Lecture Notes in Computer Science, vol. 1955, Springer, 2000, p. 131–160.

[6] B. LEGEARD, F. PEUREUX. *B-Testing-Tools : génération de tests aux limites à partir de spécifications B*, in "TSI, Techniques et Sciences Informatiques, Hermès-Lavoisier", vol. 21, n^o 9, 2002, p. 1189–1218.

[7] B. LEGEARD, F. PEUREUX, M. UTTING. *Automated Boundary Testing from Z and B*, in "Formal Methods Europe (FME 2002)", L.-H. ERIKSSON, P. LINDSAY (editors). , Lecture Notes in Computer Science, vol. 2391, Springer, 2002, p. 21–40.

[8] M. RUSINOWITCH, M. TURUANI. *Protocol Insecurity with Finite Number of Sessions and Composed Keys is NP-complete*, in "Theoretical Computer Science", vol. 299, April 2003, p. 451-475, http://www.loria.fr/~rusi/pub/tcsprotocol.ps.gz.

[9] C. TINELLI, C. RINGEISSEN. *Unions of Non-Disjoint Theories and Combinations of Satisfiability Procedures*, in "Theoretical Computer Science", vol. 290, n^o 1, 2003, p. 291–353.

Books and Monographs

[10] L. VIGNERON (editor). *Proceedings of the 19th International Workshop on Unification*, Available as LORIA Research Report A05-R-022, Nancy, France, April 2005, http://rewriting.loria.fr/UNIF-2005/unif05.ps.gz.

Doctoral dissertations and Habilitation theses

[11] F. BOUQUET. *Sémantique interprétative de spécifications formelles en animation symbolique et génération de tests à partir de modèles*, Habilitation, Laboratoire d'Informatique de l'Université de Franche-Comté, Besançon, France, June 2005.

Articles in refereed journals and book chapters

[12] S. Anantharaman, P. Narendran, M. Rusinowitch. *Closure properties and decision problems of dag automata*, in "Information Processing Letters", vol. 94, 2005, p. 231–240.

[13] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani. *An NP Decision Procedure for Protocol Insecurity with XOR*, in "Theoretical Computer Science", vol. 338, n^{o} 1-3, June 2005, p. 247-274, http://www.loria.fr/~rusi/pub/xortcs.pdf.

[14] H. Comon-Lundh, V. Cortier. *Tree automata with one memory, set constraints and cryptographic protocols*, in "Theoretical Computer Science", vol. 331, n^{o} 1, February 2005, p. 143-214, http://www.loria.fr/~cortier/Papiers/ComonCortierTCS1.ps.

[15] V. Cortier. *Vérifier les protocoles cryptographiques*, in "Technique et Science Informatique", vol. 24, n^{o} 1, 2005, p. 115-140, http://www.loria.fr/~cortier/Papiers/protocolesTSI.ps.

[16] V. Cortier, S. Delaune, P. Lafourcade. *A Survey of Algebraic Properties Used in Cryptographic Protocols*, in "Journal of Computer Security", to appear.

[17] G. Cécé, P.-C. Héam, Y. Mainier. *Clôtures transitives de semi-commutations et model-checking régulier*, in "Technique et Science Informatiques", to appear.

[18] A. Imine, M. Rusinowitch, G. Oster, P. Molli. *Formal Design and Verification of Operational Transformation Algorithms for Copies Convergence*, in "Theoretical Computer Science", to appear.

Publications in Conferences and Workshops

[19] M. Abadi, V. Cortier. *Deciding knowledge in security protocols under (many more) equational theories*, in "Proc. 18th IEEE Computer Security Foundations Workshop (CSFW'05), Aix-en-Provence, France", IEEE Comp. Soc. Press, June 2005, p. 62-76.

[20] S. Anantharaman, P. Narendran, M. Rusinowitch. *Tree vs Dag Automata*, in "Proceedings of the 19th International Workshop on Unification, Nara, Japan", L. Vigneron (editor). , April 2005, p. 93-103.

[21] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santos Santiago, M. Turuani, L. Viganò, L. Vigneron. *The AVISPA Tool for the automated validation of internet security protocols and applications*, in "17th International Conference on Computer Aided Verification, CAV'2005, Edinburgh, Scotland", K. Etessami, S. Rajamani (editors). , Lecture Notes in Computer Science, vol. 3576, Springer, 2005, p. 281-285.

[22] A. Armando, M. P. Bonacina, S. Ranise, S. Schulz. *Big proof engines as little proof engines: new results on rewrite-based satisfiability procedures (Extended abstract)*, in "Notes of the Third Workshop on Pragmatics of Decision Procedures in Automated Reasoning (PDPAR), co-located with the Seventeenth International Conference on Computer Aided Verification (CAV'05), Edinburgh, Scotland (UK)", 2005.

[23] A. Armando, M. P. Bonacina, S. Ranise, S. Schulz. *On a rewriting approach to satisfiability*

procedures: extension, combination of theories and an experimental appraisal, in "Proc. of the 5th Int. Workshop on Frontiers of Combining Systems (FroCoS'05), Vienna, Austria", B. GRAMLICH (editor). , Lecture Notes in Artificial Intelligence, vol. 3717, Springer, September 2005, p. 65–80.

[24] M. BACKES, A. DATTA, A. DEREK, J. C. MITCHELL, M. TURUANI. *Compositional Analysis of Contract Signing Protocols*, in "Proc. of 18th IEEE Computer Security Foundations Workshop (CSFW 05)", Aix-en-Provence, France, 2005, p. 94-110.

[25] M. BAUDET, V. CORTIER, S. KREMER. *Computationally Sound Implementations of Equational Theories against Passive Adversaries*, in "Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05), Lisboa, Portugal", Lecture Notes in Computer Science, vol. 3580, Springer, July 2005, p. 652-663, http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BCK05-eprint.pdf.

[26] N. BOUGHANMI, S. RANISE, C. RINGEISSEN. *On Structural Information and the Experimental Evaluation of SMT Tools*, in "Proc. of the International Workshop on First-Order Theorem Proving (FTP'05), Koblenz, Germany", 2005.

[27] F. BOUQUET, F. DADEAU, J. GROSLAMBERT. *Checking JML Specifications with B Machines*, in "Procs of the Int. Conf. on Formal Specification and Development in Z and B, (ZB'05), Guildford, UK", H. TREHARNE, S. KING, M. HENSON, S. SCHNEIDER (editors). , LNCS, vol. 3455, Springer-Verlag, April 2005, p. 435–454.

[28] F. BOUQUET, F. DADEAU, B. LEGEARD. *How Symbolic Animation can help designing an Efficient Formal Model*, in "Proc. of the 7th Int. Conf. on Formal Engineering Methods (ICFEM'05), Manchester, UK", LNCS, vol. 3785, Springer-Verlag, November 2005, p. 96–110.

[29] F. BOUQUET, F. DADEAU, B. LEGEARD. *Using Constraint Logic Programming for the Symbolic Animation of Formal Models*, in "Procs of the Int. Workshop on Constraints in Formal Verification (CFV'05) – Co-located with the Int. Conf. on Automated Deduction (CADE'05), Tallinn, Estonia", J. MARQUES-SILVA, M. VELEV (editors). , July 2005, p. 32–46.

[30] F. BOUQUET, F. DADEAU, B. LEGEARD, M. UTTING. *JML-Testing-Tools: a Symbolic Animator for JML Specifications using CLP*, in "Procs of the 11th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems, Tool session (TACAS'05), Edinburgh, UK", N. HALBWACHS, L. ZUCK (editors). , LNCS, vol. 3440, Springer-Verlag, April 2005, p. 551–556.

[31] F. BOUQUET, F. DADEAU, B. LEGEARD, M. UTTING. *Symbolic Animation of JML Specifications*, in "Procs of the Int. Conf. on Formal Methods (FM'2005), Newcastle Upon Tyne, UK", J. FITZGERALD, I. HAYES, A. TARLECKI (editors). , LNCS, vol. 3582, Springer-Verlag, July 2005, p. 75–90.

[32] F. BOUQUET, E. JAFFUEL, B. LEGEARD, F. PEUREUX, M. UTTING. *Requirement Traceability in Automated Test Generation - Application to Smart Card Software Validation*, in "Procs. of the ICSE Int. Workshop on Advances in Model-Based Software Testing (A-MOST'05), St. Louis, USA", ACM Press, May 2005.

[33] M. BOZZANO, R. BRUTTOMESSO, A. CIMATTI, T. JUNTTILA, P. VAN ROSSUM, S. RANISE, R. SEBASTIANI. *Efficient Satisfiability Modulo Theories via Delayed Theory Combination*, in "Proc. of the Int. Conf. on Computer-Aided Verification (CAV 2005), Edinburg, Scotland (UK)", Lecture Notes in Computer Science, n° 3576, Springer, 2005.

[34] Y. CHEVALIER, M. RUSINOWITCH. *Combining Intruder Theories*, in "Proc. of the Int. Coll. on Automata, Languages and Programming, ICALP, Lisbon, Portugal", Lecture Notes in Computer Science, vol. 3580, Springer, 2005, p. 639-651.

[35] Y. CHEVALIER, M. RUSINOWITCH. *Combining Intruder Theories*, in "Proceedings of the 19th International Workshop on Unification, Nara, Japan", L. VIGNERON (editor). , April 2005, p. 63-76.

[36] P. CHEVALLEY, B. LEGEARD, J. ORSAT. *Automated Test Case Generation for Space On-Board Software*, in "DASIA 2005, Data Systems In Aerospace Int. Conf., Edinburgh, UK", EUROSPACE (editor). , May 2005, p. 153–159.

[37] N. CHRIDI, L. VIGNERON. *Modélisation des propriétés de sécurité de protocoles de groupe*, in "Actes du 1er Colloque sur les Risques et la Sécurité d'Internet et des Systèmes, CRiSIS, Bourges, France", Best paper award of the conference, October 2005, p. 119-132.

[38] V. CORTIER, M. RUSINOWITCH, E. ZALINESCU. *A resolution Strategy for Verifying Cryptographic Protocols with CBC Encryption and Blind Signatures*, in "Proc. of the 7th ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP'05), Lisboa, Portugal", ACM press, July 2005, p. 12-22.

[39] V. CORTIER, B. WARINSCHI. *Computationally Sound, Automated Proofs for Security Protocols*, in "Proc. 14th European Symposium on Programming (ESOP'05), Edinburgh, U.K", Lecture Notes in Computer Science, vol. 3444, Springer, April 2005, p. 157-171, http://www.loria.fr/~cortier/Papiers/CortierWarinschi.ps.

[40] J.-F. COUCHOT, A. GIORGETTI, N. KOSMATOV. *A Uniform Deductive Approach for Parameterized Protocol Safety*, in "ASE '05: Proceedings of the 20th IEEE International Conference on Automated Software Engineering", 2005, p. 364–367.

[41] A. DATTA, A. DEREK, J. C. MITCHELL, V. SHMATIKOV, M. TURUANI. *Probabilistic Polynomial-time Semantics for a Protocol Security Logic*, in "Proc. of 32nd International Colloquium on Automata, Languages and Programming (ICALP 05)", Lisboa, Portugal, 2005, p. 16-29.

[42] P. FONTAINE, S. RANISE, C. G. ZARBA. *Combining lists with non-stably infinite theories*, in "Proc. of 11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), Montevideo, Uruguay", F. BAADER, A. VORONKOV (editors). , Lecture Notes in Computer Science, vol. 3452, Springer, March 2005, p. 51–66.

[43] A. IMINE, P. MOLLI, G. OSTER, M. RUSINOWITCH. *Towards Synchronizing Linear Collaborative Objects with Operational Transformation*, in "Twenty-Fifth IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems - FORTE'2005, Taipei, Taiwan", F. WANG (editor). , Lecture Notes in Computer Science, vol. 3731, Springer, October 2005, p. 411-427.

[44] H. KIRCHNER, S. RANISE, C. RINGEISSEN, D.-K. TRAN. *On Superposition-Based Satisfiability Procedures and their Combination*, in "Proc. of the 2nd International Colloquium on Theoretical Aspects of Computing (ICTAC'05), Hanoi, Vietnam", D. V. HUNG, M. WIRSING (editors). , Lecture Notes in Computer Science, vol. 3722, Springer, October 2005, p. 594–608.

[45] N. KOSMATOV. *A Constraint Solver for Sequences*, in "Proceedings of the 1st International Workshop on Constraint Programming Beyond Finite Integer Domains (BeyondFD'05), Sitges, Spain", October 2005, p. 49-54.

[46] N. KOSMATOV. *A Constraint Solver for Sequences and its Applications*, in "Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC'06), Dijon, France", to appear, April 2006.

[47] G. OSTER, P. URSO, P. MOLLI, A. IMINE. *Edition collaborative sur réseau pair-à-pair à large échelle*, in "Journées Francophones sur la Cohérence des Données en Univers Réparti, CDUR, Paris, France", November 2005.

[48] S. RANISE, D. DÉHARBE. *Satisfiability Checking for Software Verification*, in "Proc. of IEEE/NASA Workshop on Leveraging Applications of Formal Methods, Verification, and Validation (ISOLA'05), Baltimore, Maryland (USA)", 2005.

[49] S. RANISE, C. RINGEISSEN, C. G. ZARBA. *Combining data structures with nonstably infinite theories using many-sorted logic*, in "Proc. of the 5th Int. Workshop on Frontiers of Combining Sys tems (FroCoS'05), Vienna, Austria", B. GRAMLICH (editor). , Lecture Notes in Artificial Intelligence, vol. 3717, Springer, September 2005, p. 48–64.

[50] J. SANTIAGO, L. VIGNERON. *Study for Automatically Analysing Non-repudiation*, in "Actes du 1er Colloque sur les Risques et la Sécurité d'Internet et des Systèmes, CRiSIS, Bourges, France", October 2005, p. 157-171.

[51] T. TRUDERUNG. *Regular Protocols and Attacks with Regular Knowledge*, in "CADE-20, 20th International Conference on Automated Deduction, Tallinn, Estonia, July 22-27, 2005, Proceedings", R. NIEUWENHUIS (editor). , Lecture Notes in Computer Science, vol. 3632, Springer, 2005, p. 377-391.

[52] T. TRUDERUNG. *Selecting Theories and Recursive Protocols*, in "CONCUR 2005 - Concurrency Theory, 16th International Conference, San Francisco, CA, USA, August 23-26, 2005, Proceedings", M. ABADI, L. DE ALFARO (editors). , Lecture Notes in Computer Science, vol. 3653, Springer, 2005, p. 217-232.

[53] M. TURUANI, H. VU-VAN. *Validation of the ASW Contract Signing Protocol*, in "Proc. of APPSEM II Workshop (APPSEM05)", Germany, 2005.

[54] L. VIGNERON. *A Tool helping to Design Cryptographic Protocols (tutorial)*, in "4th Conference on Security and Network Architectures (SAR'05), Batz sur Mer, France", June 2005.

Internal Reports

[55] Y. BOICHUT, P.-C. HÉAM, O. KOUCHNARENKO. *Automatic Verification of Security Protocols Using Approximations*, Research Report, n^o RR-5727, INRIA, October 2005, http://www.inria.fr/rrrt/rr-5727.html.

[56] V. CORTIER, X. GOAOC, M. LEE, H.-S. NA. *A note on maximally repeated sub-patterns of a point set*, Research Report, n^o RR-5773, INRIA, December 2005, http://www.inria.fr/rrrt/rr-5773.html.

[57] F. JACQUEMARD, M. RUSINOWITCH, L. VIGNERON. *Tree automata with equality constraints modulo equational theories*, Research Report, n^o LSV-05-16, Laboratoire Spécification et Vérification, ENS Cachan,

France, August 2005.

Miscellaneous

[58] N. CHRIDI. *Vérification de protocoles de groupes*, Rapport de DEA, LORIA – Université Henri Poincaré, Nancy, 2005.

Bibliography in notes

[59] F. BAADER, K. U. SCHULZ. *Unification in the Union of Disjoint Equational Theories: Combining Decision Procedures*, in "Journal of Symbolic Computation", vol. 21, n^o 2, February 1996, p. 211–243.

[60] F. BELLEGARDE, C. DARLOT, J. JULLIAND, O. KOUCHNARENKO. *Reformulation: a Way to Combine Dynamic Properties and Refinement*, in "International Symposium Formal Methods Europe (FME 2001)", LNCS, vol. 2021, Springer-Verlag, 2001.

[61] E. BERNARD, B. LEGEARD, X. LUCK, F. PEUREUX. *Generation of Test Sequences from Formal Specifications: GSM 11-11 Standard Case-Study*, in "International Journal on Software Practice and Experience", vol. 34, n^o 10, 2004, p. 915–948.

[62] F. BOUQUET, B. LEGEARD. *Reification of Executable Test Scripts in Formal Specification-Based Test Generation: The Java Card Transaction Mechanism Case Study*, in "Formal Methods, FME 2003", vol. 2805, Springer-Verlag, September 2003, p. 778–795.

[63] F. BOUQUET, B. LEGEARD, F. PEUREUX. *CLPS-B - A Constraint Solver for B*, in "International Conference on Tools and Algorithms for Construction and Analysis of Systems, TACAS2002, Grenoble, France", Lecture Notes in Computer Science, vol. 2280, Springer, April 2002, p. 188–204.

[64] G. CÉCÉ, P.-C. HÉAM, Y. MAINIER. *Clôtures transitives de semi-commutations et model-checking régulier*, in "Congrès Approches Formelles dans l'Assistance au Développement de Logiciels, AFADL'04, Besançon, France", J. JULLIAND (editor). , June 2004, p. 257–268.

[65] G. CÉCÉ, P.-C. HÉAM, Y. MAINIER. *Efficiency of Automata in Semi-Commutation Verification Techniques*, in "Theoretical Informatics and Applications", Also available as Research Report 5001, INRIA, France, 2003, http://www.inria.fr/rrrt/rr-5001.html.

[66] D. DÉHARBE, S. RANISE. *Light-Weight Theorem Proving for Debugging and Verifying Units of Code*, in "Proc. of the International Conference on Software Engineering and Formal Methods (SEFM03), Brisbane, Australiab", IEEE Computer Society Press., September 2003, http://www.loria.fr/~ranise/pubs/sefm03.ps.gz.

[67] S. EVEN, O. GOLDREICH. *On the Security of Multi-Party Ping-Pong Protocols*, in "IEEE Symposium on Foundations of Computer Science", 1983, p. 34-39, http://citeseer.ist.psu.edu/46982.html.

[68] T. GENET, F. KLAY. *Rewriting for Cryptographic Protocol Verification*, in "Proceedings of CADE'00", LNCS 1831, Springer-Verlag, 2000, p. 271–290.

[69] G. T. Leavens, A. L. Baker, C. Ruby. *JML: a Java Modeling Language*, in "Formal Underpinnings of Java Workshop (at OOPSLA '98)", October 1998, http://www-dse.doc.ic.ac.uk/~sue/oopsla/cfp.html.