INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Cassis

Combinaison d'Approches pour la Sécurité des Systèmes InfiniS

Lorraine

THEME SYM

Table of contents

1. Team

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).

Team Leader
M. Rusinowitch [DR INRIA]

Team Vice-Leader
F. Bellegarde [PR, Université Franche-Comté, LIFC]

Administrative Assistant
S. Drouot [LORIA]

Staff members
V. Cortier [CR, CNRS au LORIA]
S. Ranise [CR, INRIA au LORIA]

Faculty members (LORIA)
L. Vigneron [MC, Université Nancy 2, en délégation au CNRS]

Faculty members (Université Franche-Comté)
F. Ambert [MC]
F. Bouquet [MC]
A. Giorgetti [MC]
P.-C. Héam [MC]
O. Kouchnarenko [MC]
B. Legeard [PR]
F. Peureux [MC]

Post-doctoral Fellows
Y. Chevalier [Teaching Assistant until Aug. 31st, 2004]
T. Truderung [From Oct. 1st, 2004, Poland]
C. Zarba [From April 1st, 2004, Italy]

Ph. D. Students
T. Abbes [BDI-PED, LORIA]
Y. Boichut [INRIA, LIFC]
S. Chemin [ANVAR, LIFC]
J.-F. Couchot [PRAG UFC, LIFC]
F. Dadeau [ACI, LIFC]
A. Imine [Teaching Assistant, LORIA]
A. A. Oliveira Viana da Silva [ALBAN, LORIA from Dec. 1st, 2004]
J. Santos Santiago [INRIA, LORIA]
N. Vacelet [MENRT, LIFC, until Sept. 30th, 2004]
E. Zalinescu [MENRT, LORIA, from Oct. 1st, 2004]

Project Technical Staff
S. Guenaud [ANVAR, LIFC, until Aug. 31st, 2004]
A. Haloi [INRIA, LORIA, until Sept. 30th, 2004]
F. Nicolet [ACI EDEMOI and V3F, LIFC, from Sept. 1st, 2004]

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.

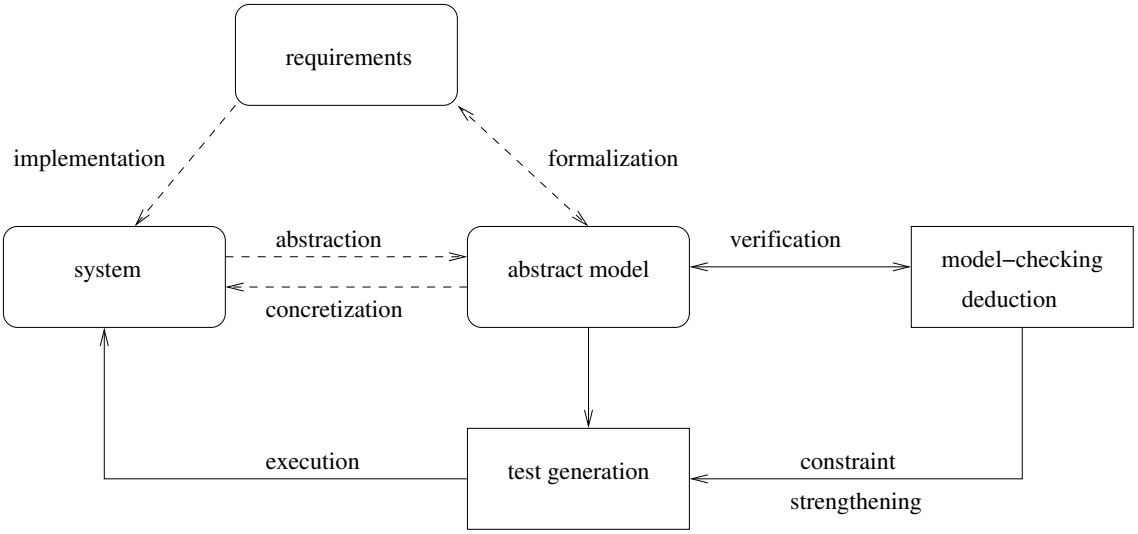

Figure 1. Software validation in CASSIS

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 $\mathcal{T}$ [56]). 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:

   i. partitioning of the formal model and extraction of boundary values,

   ii. 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. Furthermore, the kernel of the engine allows one to perform specification animations in order to validate the model [58].

3. For the safety on infinite state systems we have designed automated deduction tools based on term rewriting (**SPIKE** [2], ***daTac*** [3], ***haRVey***)and an extensible and modular platform for detecting flaws and potential attacks on security protocols (**CASRUL** [4]). 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 Binary Decision Diagrams) 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 concepts on domain finiteness and also the writing of specific solvers to deal with the description language's vocabulary.

The aim of this research is the evaluation of set-oriented formal specifications. The current work concerns the development of a set constraint solving system based on the CLPS core.

By evaluation, we mean the rewriting 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 makes it possible to look into the reachability graph, representing the states defined by the specification. It is directly used for animating specifications and automatically generating abstract test cases. Moreover, the current version of the solver is able to deal with B notation, Z notation, and State-chart notation. JML notation is being currently integrated. The principles used in animation, invariant verification and test generation can be generalized to every set-oriented notation.

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 of infinite states 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. Presently, the data are arrays transformed by generalized substitutions; they are able to capture enough non-determinism to express, for example, the parallel execution of an arbitrary number of processes. Here, the general purpose is to integrate automatic deduction inside deductive tools, 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. In the deductive approach, the semi-decidability of the automatic proof can also be a source of divergence. Therefore, this is a motivation for guaranteeing proof termination.

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.

The system **CASRUL** [4] that we have developed is a first prototype, giving an idea of what will be the final tool. It allows to consider many protocols and is maintained as a component of AVISPA platform. His specification language is extended for handling more general protocols like e-business protocols for example.

4.2. Automated boundary testing from formal specifications

In [6], 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 [12] and Java Card Virtual Machine Transaction mechanism [57]) 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 the object oriented specification.

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 two. 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. For example, there exist pointer analysis techniques capable of handling huge programs whose results are not very informative, see e.g. [62] for a discussion on this issue.

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

5. Software

5.1. CASRUL

Keywords: *Protocols, cryptography, verification.*

Participants: Y. Chevalier, V. Cortier, M. Rusinowitch, J. Santos Santiago, L. Vigneron.

CASRUL[2] is an environment for automated verification of security protocols in Dolev-Yao models and some extensions. It encompasses protocol specification in a high-level language, automated translation of such specifications to rule-based programs, and automated verification by constraint solving techniques.

In the context the European project AVISPA, the specification language has been totally renewed into a more expressive role-based language [31], permitting protocols modelers to precisely describe the semantics of each action thanks to TLA-based guarded transitions.
Specifications are automatically translated into an intermediate format based on first-order multiset rewriting [33], defining an operational semantics for the protocol.
This intermediate format is then used by verification tools for either looking for flaws, or proving the validity of a property.

As back-ends, we use AtSe [47], a constraint solver, and TA4SP, an approximation-based verification tool.
CASRUL has already been used for analyzing many real Internet security protocols (UMTS-AKA, ChapV2, EKE, AAAMobileIP, TLS, Kerberos).

5.2. TA4SP

Keywords: *Tree automata, approximation function, under and over-approximation.*

Participants: Y. Boichut, P.-C. Héam, O. Kouchnarenko.

The TA4SP system (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols) is a prototype for verifying security protocols. The investigated approach, initiated by Genet and Klay [61], is to over-estimate or to under-estimate the intruder knowledge by using regular tree languages. This method allows one to show that some states are reachable or not, and hence that the intruder will be able to know certain terms or not.

Thus, three kinds of conclusions are proposed by this prototype: either a protocol is flawed (under-approximation), or a protocol is secured (over-approximation), or no conclusion can be raised (over-approximation + a term is known by the intruder while it is not supposed to be).

TA4SP is developed in Ocaml and is composed of a translator from an HLPSL (High-Level Protocol Specification Language) specification to Timbuk input format including an automatic generation of a new approximation function and a package handling a new approximation function for Timbuk 2.0[3].

TA4SP is documented and is accessible on the web[4]. ***TA4SP*** is used in the CASSIS group for verifying security protocols proposed by an industrial partner in AVISPA project.

5.3. BZ-Testing-Tools

Keywords: *Test generation, animation of specifications, formal specification.*

Participants: F. Ambert, F. Bouquet, S. Guenaud, B. Legeard, F. Peureux, N. Vacelet.

BZ-Testing-Tools[5] (BZ-TT, for short) is a tool-set for animation and test generation from B, Z and State-chart specifications. BZ-Testing-Tools 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). The tool-set is composed of three graphic

[2] http://www.loria.fr/equipes/cassis/softwares/casrul/
[3] http://www.irisa.fr/lande/genet/timbuk/
[4] http://lifc.univ-fcomte.fr/~boichut/TA4SP/
[5] http://lifc.univ-fcomte.fr/~bztt

user interfaces, for animation, test generation and reification. Animation is used to validate the model. Test generation is used to define the test to validate. Reification is used to translate abstract tests into concrete executable scripts.

5.4. JML-Testing-Tools

Keywords: *Formal specification, Java Modeling Language, model-based, symbolic animation.*

Participants: F. Bouquet, F. Dadeau, B. Legeard.

JML-Testing-Tools[6] – JML-TT – is a framework for the symbolic animation of formal models written using JML annotations [63] embedded within Java programs. It relies on the BZ-Testing-Tools technology of symbolic execution and constraints solving. 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. The tool-set is implemented with an attractive graphical user interface, which makes it possible to generate test cases from a user-defined execution sequence. Therefore, the implementation can be compared with the model by applying the technique of runtime assertion checking. This tool is also a support by the ACI GECCOO project.

5.5. haRVey and related tools

Keywords: *Automated deduction, BDDs, boolean reasoning, equational theories, satisfiability, saturation theorem proving.*

Participants: J.-F. Couchot, A. Giorgetti, S. Ranise [correspondent].

haRVey[7] is a theorem prover for first-order logic with equality [60]. 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 schema between superposition theorem proving to flexibly reason in T and an implementation of Fourier-Motzkin method for linear arithmetic [40]. ***haRVey*** has been especially designed to be integrated in larger verification systems. It is integrated in Barvey[8], a tool [36] 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 of LRI, Université Paris Sud, Orsay) can generate proof obligations for ***haRVey*** to check the correctness of ML or C programs. The integration of ***haRVey*** in the Athena proof development system(developed by K. Arkoudas of MIT, Boston, USA) is undergoing. Finally, the system will be the core reasoning module of a verification system for checking TLA$^+$ specifications which is being designed and developed in colloraboration with Stephan Merz of the Model group at LORIA.

5.6. Others Tools

Most of the softwares 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] http://lifc.univ-fcomte.fr/~jmltt/
[7] http://www.loria.fr/equipes/cassis/softwares/haRVey/
[8] http://lifc.univ-fcomte.fr/~couchot/soft/barvey/

6. New Results

6.1. Automated deduction

6.1.1. Decision procedures and their extensions

Keywords: *Propositional and first-order satisfiability, combination of decision procedures, integration, quantifier instantiation.*

Participants: A. Imine, J.-F. Couchot, A. Giorgetti, S. Ranise, M. Rusinowitch, C. Zarba.

In many cases, applying software analysis techniques reduces to the problem of proving the unsatisfiability of a formula of first-order logic with a complex Boolean structure modulo a background theory. Automatically, discharging this kind of proof obligation requires to carefully integrate three different reasoning activities: Boolean solving, checking satisfiability in (a combination of) background theories, quantifier handling. Motivated by the practical need of discharging the proof obligations resulting from B specifications, we have developed a technique for handling quantified formulae in a fragment of set theory and implemented it in a tool whose main reasoning module is ***haRVey*** [59]. The experimental results are encouraging (we discharge proof obligations which cannot be handled by the commercial tool AtelierB) and we are currently investigating the decidability of a class of formulae which arise in this kind of verification problems.

Since arithmetic reasoning is present in virtually any verification problems, we have proposed a generalization of the Nelson-Oppen combination schema to integrate a decision procedure for linear arithmetic with equational reasoning and quantifier instantiation [40]. Experiments with an implementation in ***haRVey*** on proof obligations arising in the verification of sorting algorithms shows that the technique is better (in terms of number of proof obligations automatically discharged) than the state-of-the-art tool, *Simplify*.

Usually, the background theory of proof obligations is obtained as the combination of many different theories. Hence, it is crucial to address the problem of combining decision procedures for the constituent theories. This research line has recently seen a lot of activity which has resulted in quite heterogenous and technical presentations. As a result, non-experts have difficulties in using existing results and experts do not have a reference framework to work in. In order to overcome these difficulties, we have proposed a rational reconstruction of the combination schemas of Nelson-Oppen and Shostak in a uniform and abstract framework [45]. This allows us to derive all recent results related to such schemas and to propose a new combination schema which combines the best of the previous two and paves the way to transfer rewriting techniques in this context.

Equational unification is central in automated deduction. We have investigated unification modulo theories that extend the well-known ACI (associative, commutative, idempotent) by adding a binary symbol '$*$' that distributes over the ACI-symbol '$+$'. We report some decidability results in [11].

6.1.2. Verification of Copies Convergence in Collaborative Editing Systems

Keywords: *Collaborative, consistency, editor synchronization, proof.*

Participants: A. Imine, M. Rusinowitch.

Distributed groupware systems provide computer support for manipulating objects such as text documents or filesystems, shared by two or more geographically separated users. Data replication is a technology to improve performance and availability of data in distributed groupware systems. Each user has a local copy of the shared objects upon which he may perform updates. Locally executed updates are then transmitted to the other users. However replication potentially leads to divergent copies. Then Operational Transformation (OT) algorithms are applied for achieving convergence of all copies. However the design of such algorithms is a difficult and error-prone activity since building the correct updates for maintaining convergence properties of the local copies requires examining a large number of situations.

To solve this problem we have proposed an algebraic framework for designing OT algorithms [41]. We specify interactions between a shared object and a user with Observational Semantics. Operations are divided into *methods* for modifying the states and *attributes* for consulting (or observing) the current state. We have

implemented in collaboration with ECOO project an environment for defining object operations and the associated OT algorithm. From this description the tool generates an algebraic specification that can be verified with our theorem-prover **SPIKE**, which is well-suited for reasoning about conditional theories.

Ensuring convergence with an OT approach remains a challenging issue when the shared object has a linear structure such as a list or an ordered XML tree. In [53], we have showed that all previously published OT algorithms are incorrect by providing tricky counter-examples. We also have analyzed thoroughly the source of divergences and we have proposed a new OT algorithm that is drastically simpler than the previously published ones. Moreover, our OT algorithm is generic since it can be extended to any linear structure-based object [44].

6.2. Security protocol verification

Keywords: *Protocol*, *exclusive-or*, *exponentiation*, *security*, *verification*.

6.2.1. Extension of the Dolev-Yao model

Participants: Y. Chevalier, V. Cortier, M. Rusinowitch, J. Santos Santiago, L. Vigneron.

In the domain of protocol verification, we have obtained new results relaxing the *perfect cryptography* assumption. Our first results regard a specific algebraic property of several protocols, namely commutative encryption. It occurs in protocols using RSA encryption with common modulus for example. Our second results hold for general classes of equational theories but are dedicated to the study of the intruder knowledge.

For commutative encryption [32], we have shown that, under reasonable hypotheses, checking whether a message can be derived by an intruder (using the commutativity of encryption) is in PTIME. More generally, we obtain that protocol insecurity is in NP for a standard Dolev-Yao intruder that can also exploit the properties of commutative encryption.

Decidability results under general equational theories have been rare. We 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. We have shown [23] that both ground deducibility and static equivalence are decidable in PTIME for *convergent subterm theories*, defined by a convergent rewriting system with rewriting rules of the form $M \rightarrow N$ where N is a subterm of M. We are currently extending these results to more general theories, allowing AC-symbols to capture theories like the homomorphism, the modular exponentiation, or the exclusive or.

6.2.2. Soundness of the Dolev-Yao model

Participant: V. Cortier.

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 [52] 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 this work is supported by the ACI Jeunes Chercheurs Crypto.

6.2.3. Intruder knowledge approximation

Participants: Y. Boichut, P.-C. Héam, O. Kouchnarenko.

When the number of sessions is unbounded, the problem of the safety of a security protocol is undecidable, even in restricted cases. Nevertheless, since the safety of real protocols such as internet-protocols is crucial, it turns out to be necessary to be able to handle nevertheless this problem. For that purpose, we have investigated the semi-algorithmic method introduced by Genet and Klay: intruder's knowledge is approximated by a tree automaton language. 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 can be used here to effectively model the knowledge that the intruder might have acquired from previous sessions.

The challenge is then to obtain a not too coarse approximation. We have developed a new approximation function which is automatically generated and we have extended the class of analyzable protocols [26]. These techniques have been implemented in the tool TA4SP (Tree Automata based Approximation for the Analysis of Security Protocols). We have successfully verified 7 protocols provided by Siemens, our industrial partner in AVISPA project; the verification is totally automatic for protocols specified in an high level specification language (HLPSL).

For the completeness issue and in order to be able to re-build an attack, we have also been developing an automatically generated under-approximation of intruder knowledge. This approach provides encouraging results on toy examples, such as NSPK protocols. We are investigating more relevant applications.

6.2.4. On-line intrusion detection

Participants: T. Abbes, M. Rusinowitch.

Cyber-criminality development has raised an increasing interest for intrusion detection as a way to complement firewall supervision and thus to discover more attacks. We have studied some bottlenecks for intrusion detection: namely high load traffic, evasion techniques and false alerts generation.

In order to supervise overloaded networks, we propose to classify traffics using Intrusion Detection Systems (IDS) characteristics and network security policies. Therefore each IDS has less IP traffic to observe and less detection rules to manage. Moreover we show how to reduce the packets processing time on each IDS by a careful attack detection rules application. We propose two methods for selecting the detection rules. The first one builds a Direct Acyclic Graph which allows a quick choice of candidate rules. The second one uses common properties of rules to gather them in different groups. This clustering accelerates the search by avoiding redundant tests.

The rules selection stage is followed by an IP traffic analysis. During this phase, we rely on an on-the-fly pattern matching strategy to look simultaneously for several signatures [10][25]. Thus, we avoid the costly traffic reassembly phase that was previously employed to deceive evasion techniques. Besides we perform some protocol analysis by decision trees in order to accelerate the intrusion detection [24]. This allows one to reduce the number of false positives compared with a rough pattern matching method. Different protocols have been integrated that way such as FTP, TELNET, SMTP, SNMP, RPC, DNS and HTTP.

6.3. Reachability analysis

6.3.1. Reachability with generalized substitutions

Keywords: *Set-theoretic specification, boundary test, first-order logic*.

Participants: F. Bellegarde, J.-F. Couchot, A. Giorgetti, S. Ranise.

The automatic verification of safety properties of infinite state systems by deductive reachability analysis involves four ingredients: 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.

We focused on B specifications where data are structured in arrays, transitions are defined by generalized substitutions, state configurations are represented by first order formulae with equality and ***haRVey*** is used as

a semi-decision procedure. As a prerequisite to reachability analysis we showed how to prove the correctness of a B abstract machine by checking its invariant [59]. When the proof fails, we showed how to automatically strength the invariant predicate [35]. We described an efficient method [38] to compute symbolic reverse images and showed its applicability for a class of uniform distributed systems.

6.3.2. Reachability Computation with Regular Languages

Keywords: *Parametric systems*, *reachability*, *regular languages*.

Participants: F. Bellegarde, P.-C. Héam.

The *regular model checking* techniques use regular languages for reachability analysis: states of the system are represented by finite automata or regular expressions and actions are modelized by transducers or rewriting rules on words.

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

Following previous work, we have proposed in [39] a technical modification which extends the domain of the applications that we can handle with the method. For example, we have computed the accessibility set of the configurations of a two lifts controller – this could not have been done by the previous algorithms. These techniques have been conclusively implemented in OCAML.

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

Keywords: *Formal Specifications*, *Model-Based Testing*, *Test Case Generation*, *test Driver Generation*.

Participants: F. Ambert, F. Bouquet, S. Chemin, F. Dadeau, B. Legeard, F. Peureux, N. Vacelet.

The need to offer better methods and tools for functional black-box testing of large scale systems has risen 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 have addressed the problem of mastering the test number explosion by the formalization of several model coverage criteria to allow the test engineer to choose the level of model coverage during test generation [12][28].

We have tried to improve the approach in the preamble calculus by studying of backward-chaining algorithm using the constraint solver CLPS [15], in the optimization of internal graph representation of predicates [29] and in solver by completing the data-structure and by optimizing the resolution of CLPS [13].

We have extended the capacity of tools by the extension of input language with the state-charts and the test generation based on specification behavior [34][27] and the output by introducing the generation of test drivers from the abstract generated test cases [42].

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

[9] http://www.lsv.ens-cachan.fr/prouve/

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.

7.2. Research Result Transfer

The BZ-Testing-Tools technology has been transferred to Leirios Technologie, at the end of the year 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 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. 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 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 Lausane - EPFL - concerning Test generation from UML/OCL formal models. The duration of the project is 2 years and it was started in November 2002.

8. Other Grants and Activities

8.1. International grants

- SECURYPTO[11], a new France-Quebec research network between LIFO (Orléans), IMAG (Grenoble), LORIA (Nancy), UQAM (Montréal) investigates the specification and verification of security properties in process algebras, using notions and techniques from: rewriting, constraints, information flow and interference, process localities and performance.
- We (Nancy) are collaborating with David Deharbe of Department of Computer Science and Applied Mathematics, of UFRN, Federal University of Rio Grande do Norte (Natal, Brazil) on the development of the system ***haRVey*** and its application to various verification problems.
- We (Besançon) are collaborating with Mark Utting, senior lecturer of University of Waikato, New-Zealand, on the development of the system BZ-TT and its application to Z specifications.

[10] http://www.avispa-project.org/
[11] http://www.hains.org/securypto

8.2. National grants

- ACI Securité 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. The methods and tools will be developed to be integrated, *i.e.* together they will form a coherent design method from specification to certified code generation using refinement, simulation, testing and verification techniques. 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: TFC (LIFC), Lemme (INRIA), LogiCal (LRI), VASCO (LSR).
- 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. More precisely, V3F will investigate techniques to check that a program satisfies the calculations hypothesis on the real numbers that have been done during the modeling step. The underlying technology is based on constraint solving.
  Partners: 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: LSR-IMAG, CEDRIC-CNAM, ONERA Toulouse, GET ENST Paris.
- ACI SATIN[12] —"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 taking up the challenge of formal analysis and design of secure distributed systems, by taking advantage of the recent advances in algebraic modeling techniques, constraint solving, tree automata, and observation criteria for concurent systems.
  Partners: CEA-DAM, France Telecom R&D, LANDE project - IRISA, VPS team, LIFO.
- ACI Jeunes Chercheurs CRYTO[13] —"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. The computational approach 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 formal approach relies on a symbolic model of protocol executions in which cryptographic primitives are treated as black boxes. It enables significantly simpler and often automated proofs. However, the guarantees that it offers have been quite unclear. Linking the two approaches would enables to obtain the best of both worlds: fully automated proofs and strong, clear security guarantees.
  Members: Yannick Chevalier, Véronique Cortier (recipient), Judson Santos Santiago, Emmanuel Thomé, Mathieu Turuani.
- With respect to the *Contrat de Plan État-Région Lorraine 2000-2006*, we are working in the *Pôle de Recherche Scientifique et Technologique Intelligence Logicielle* within the theme: - Qualité et sûreté des logiciels et systèmes informatiques - with the action VALDA (2003-2004).

[12] http://lifc.univ-fcomte.fr/~heampc/SATIN
[13] http://www.loria.fr/~cortier/aci.html

- In the Institut des Sciences et Technologies de l'Information, we animate one of three research area. It is composed of 9 research projects. We have one project labelled B-Testing-Tools and a second will begin in December 2004 about Test from Statemate/Simulink specification.

8.3. International collaborations

- PAI PROCOPE: Combining automata-theoretic and rewriting techniques for the analysis of cryptographic protocols. The participants are the CASSIS project and the team 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.[14] This cooperation is supported by the France-New-Zealand scientific program.

8.4. Individual involvement

The *TFC team* has hosted and organized the national Conference *Approches Formelles dans l' Assistance au Développement de Logiciels 2004 (AFADL'2004)* on June 16-18 2004.
F. Bellegarde: director of the research team *Techniques Formelles et à Contraintes (TFC)* of the *Laboratoire d'informatique de Franche Comté (LIFC)*, Editorial committee member of *Techniques et Science Informatique (TSI)*, Local Coordinator *(LIFC)* of the ACI GECCOO.
F. Bouquet: In charge of the Mobilization area (9 research projects, with 46 researchers and 8 laboratories) in ISTI Institute.[15] Coordinator of Tools session of AFADL'04.
V. Cortier: recipient of the ACI Jeunes Chercheurs Crypto. Member of the program committee of the *Workshop on Formal Aspects in Security and Trust (FAST2004)*.
A. Imine: Member of the program committee of the first IEEE *International Workshop on Electronic Contracting (WEC)*, July 6 2004, San Diego (California, USA).
O. Kouchnarenko: vice-president of the CSE 27 of the Franche-Comté Université; PC member of "*Approches Formelles dans l'Assistance au Développement de Logiciels*", AFADL'04. Besançon, 2004, June.
B. Legeard: Member of the Scientific council of the University of Franche-Comté. Coordinator of the ACI V3F.
S. Ranise: *Trustee* of the project CALCULEMUS (Systems for Integrated Computation and Deduction). Local coordinator of the ACI GECCOO. Co-chair of the Second Workshop on *Pragmatics of Decision Procedures in Automated Deduction (PDPAR'04)*, affiliated to IJCAR'04. Coordinator (with Cesare Tinelli) of the Satisfiability Modulo Theories Library (SMT-LIB) initiative. Organizer of the *Séminaire d'informatique fondamentale (SIF)* 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); coordinator of the project ACI Sécurité SATIN. Chairman (with D. Basin) of International Joint Conference on Automated Reasoning, 2004. PC member of ACM Conference on Computer and Communications Security, of Computer Science Logic 2004, of Rencontres Sécurité et Architecture Réseaux 2004. He is invited Editor of Technique et Science Informatiques, and Theory of Computing Systems.
L. Vigneron: member of the scientific council of Université Nancy 2; invited editor of JAR for a special issue on FTP; member of the FTP steering committee; 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.

8.5. Visits of foreign researchers

Adel Bouhoula (SupCom Tunis) visited our group (Nancy) from July 5th to July 9th, 2004. The subject of the collaboration was intruder detection.

[14] http://www.cs.waikato.ac.nz/Research/fm/index.html
[15] http://www.isti.info

David Déharbe UFRN–DIMAp, Natal (PB, Brazil) visited our group to continue the development of ***haRVey*** from February 24 to March 8 and October 12 to November 2, 2004.
Mark Utting (University of Waikato, New Zealand) is visiting our Group (LIFC) from September 2004 to February 2005 as associated CNRS researcher.
Bogdan Warinschi (University of California of Santa Cruz, USA) visited our group (Nancy) one week in July 2004. The subject of the collaboration was how to link the formal and concrete approaches for security protocols.

8.6. Visits of team members

F. Bouquet visited Mark Utting at the University of Waikato in New Zealand during two weeks (April 16 to May 3), giving a seminar on BZ-TT technology and working into the integration of Z into the BZ-Testing-Tools.
V. Cortier visited Martin Abadi, at the University of California Santa Cruz (USA) during two weeks in April 2004 and two weeks in December 2004. The subject of the collaboration is decidability results of the static equivalence under general equational theories.
S. Ranise visited the group of Prof. D. Déharbe of UFRN–DIMAp, Natal (PB, Brazil) to continue the development of ***haRVey*** from November 12 to November 19, 2004.
L. Vigneron has visited the CRAC in Quebec (May 22-29), for the Securypto summer meeting, giving a seminar on rule-based programs for describing internet security protocols. He has also visited the DIMAp in Natal, Brasil (November 12-20).

9. Dissemination

9.1. Ph. D. theses

Tarek Abbes defenses his Ph.D thesis, title "*Classification du trafic et optimisation des règles de filtrage pour la détection d'intrusions*", supervisors: Michael Rusinowitch and Adel Bouhoula, on December 14th 2004.
Nicolas Vacelet defenses his Ph.D thesis, title "*Evaluation de notations formelles de spécifications par système de contraintes*", supervisors: Pr B. Legeard and F. Bouquet, on December 1st 2004

9.2. Habilitation thesis

Olga Kouchnarenko defenses her thesis, title "*Raffiner pour vérifier des propriétés de systèmes finis et infinis*", on November, 25th 2004.

9.3. Awards

The **2003 AFIT Award** has been obtained by Mathieu Turuani (4 award winners) for his Ph. D. dissertation "*Sécurite des Protocoles Cryptographiques : Décidabilité et Complexité*".
The **2003 SPECIF Award** and the **2004 Ph. D. Award of the newspaper *Le Monde*** have been obtained by Véronique Cortier for her Ph. D. dissertation "*Vérification automatique des protocoles cryptograhiques*".

9.4. Committees

V. Cortier is a member of the 2004 SPECIF committee to award the best Ph. D. dissertations in theoretical computer science.
B. Legeard is referee for the thesis of Elena Zinovieva-Leroux, Université de Rennes-IRISA (Projet Vertecs) - December 2004.
M. Rusinowitch is a member of the AFIT committee to award the best Ph. D. dissertations in theoretical computer science of the year. *M. Rusinowitch* is referee for the thesis of Pascal Fontaine (Liège), Liana Bozga (Grenoble), and Vincent Vanackere (Marseille).

9.5. Seminars, workshops, and conferences

Besides conference talks mentioned in the publication list, we have given the following talks.
Y. Chevalier: Seminar of the Université libre de Bruxelles; Seminar at Verimag (Grenoble); Seminar at IRISA (Rennes). All the three talks were entitled *Vérification de protocoles cryptographiques*.
V. Cortier: Seminar SPACES at Loria, January, 1st, entitled *Automatic verification of cryptographic protocols*; Seminar of the École Doctorale of Marseille, October, 8th, entitled *Protocoles cryptographiques : comment assurer leur fiabilité ?* Seminar MIM of the École Normale Supérieure de Lyon, October, 19th, entitled *Vérifier les protocoles cryptographiques*.
P.-C. Héam: Seminar at the ENS de Cachan, LSV, entitled *Semi-commutations et automates*, on April 29th, 2004.
A. Imine: Talk entitled *Deductive Verification of Distributed Groupware Systems*, Nancy-Saarbrücken Workshop on Logic, Deduction and Applications, Saarbrücken, 25-26 Nov. 2004.
B. Legeard: Invited Talk at the Microsoft Resarch Summer Institute on Trends in Testing: Theory, Techniques and Tools, University of Washington, USA, entitled *Controlling Test Case Explosion in Test Generation from Formal Models*, August 2004.
S. Ranise: Series of seminars on *building and combining decision procedures* at the University of Milan (February–June, 2004). Seminar at the Istituto Trentino di Cultura (ITC) "***haRVey**: Combining Boolean Solving, Decision Procedures, and Quantifier Instantiations*".
M. Rusinowitch: Invited conference at 18th IFIP - Theoretical Computer Science, World Computer Congress, Toulouse 2004; Seminars at Universities of Namur (FUNDP), Créteil (LACL), Limoges (LACO).
L. Vigneron: invited talk at the Unif workshop entitled *Automated verification of security protocols*.
C. Zarba: May 24, 2004, Nancy, Theoretical Computer Science Seminar at LORIA, *A quantifier elimination algorithm for a fragment of set theory involving the cardinality operator*; June 17–18, 2004, Nancy, Nancy-Saarbrücken Workshop on Logic, Proofs, and Programs, Talk on June 17 on *Combining container-based data structures with non-stably infinite theories*, (Based on a joint work with Silvio Ranise and Christophe Ringeissen); November 18, Besançon, *Combining decision procedures for sorted theories*, (Based on a joint work with Cesare Tinelli).

10. Bibliography

Major publications by the team in recent years

[1] A. ARMANDO, S. RANISE, M. RUSINOWITCH. *Uniform Derivation of Decision Procedures by Superposition*, in "Proceedings of Annual Conference of the European Association for Computer Science Logic (CSL'01), Paris, France", L. FRIBOURG (editor)., Lecture Notes in Computer Science, vol. 2142, Springer, September 2001, p. 513–528.

[2] A. BOUHOULA, M. RUSINOWITCH. *Observational Proofs by Rewriting*, in "Theoretical Computer Science", vol. 275, n° 1–2, 2002, p. 675–698.

[3] Y. CHEVALIER, L. VIGNERON. *Automated Unbounded Verification of Security Protocols*, in "14th International Conference on Computer Aided Verification, CAV'2002, Copenhagen (Denmark)", E. BRINKSMA, K. GULDSTRAND LARSEN (editors)., Lecture Notes in Computer Science, vol. 2404, Springer, July 2002, p. 324–337, http://www.loria.fr/~vigneron/Work/papers/ChevalierV-CAV02.ps.gz.

[4] 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.

[5] 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, nº 9, 2002, p. 1189–1218.

[6] 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.

[7] 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.

Books and Monographs

[8] D. BASIN, M. RUSINOWITCH (editors). *Automated Reasoning: Second International Joint Conference, IJCAR 2004*, Lecture Notes in Computer Science, vol. 3097, Springer, Cork (Ireland), July 2004.

[9] D. KAPUR, L. VIGNERON (editors). *Special issue on First-Order Theorem Proving of the Journal of Automated Reasoning*, To appear., Kluwer, 2004.

Articles in referred journals and book chapters

[10] T. ABBES, A. BOUHOULA, M. RUSINOWITCH. *On the Fly Pattern Matching for Intrusion Detection with Snort*, in "Annals of telecommunications", vol. 59, nº 9-10, September-October 2004, p. 941-967.

[11] S. ANANTHARAMAN, P. NARENDRAN, M. RUSINOWITCH. *Unification Modulo ACUI Plus Distributivity Axioms*, in "Journal of Automated Reasoning", vol. 33, nº 1, December 2004, 28 pages.

[12] 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, nº 10, 2004, p. 915–948.

[13] 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, nº 2, August 2004, p. 143–157.

[14] Y. CHEVALIER, L. VIGNERON. *Strategy for Verifying Security Protocols with Unbounded Message Size*, in "Journal of Automated Software Engineering", vol. 11, nº 2, April 2004, p. 141-166.

[15] S. COLIN, B. LEGEARD, F. PEUREUX. *Preamble Computation in Automated Test Case Generation using Constraint Logic Programming*, in "The Journal of Software Testing, Verification and Reliability", Selected papers from the 2003 UK-Test Workshop, vol. 14, nº 3, 2004, p. 213–235.

[16] H. COMON-LUNDH, V. CORTIER. *Security properties: two agents are sufficient*, in "Science of Computer Programming", vol. 50, nº 1-3, March 2004, p. 51-71.

[17] H. COMON-LUNDH, V. CORTIER. *Tree automata with one memory, set constraints and cryptographic protocols*, in "Theoretical Computer Science", vol. To appear, 2004.

[18] V. CORTIER. *Vérifier les protocoles cryptographiques*, in "Technique et Science Informatique", vol. To appear, 2004.

[19] 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, vol. To appear, 2004.

[20] A. IMINE, M. RUSINOWITCH, G. OSTER, P. MOLLI. *An Algebraic Framework for Designing Operational Transformation Algorithms*, in "Theoretical Computer Science", to appear, 2005.

[21] B. LEGEARD, F. PEUREUX, M. UTTING. *Controlling Test Case Explosion in Test Generation from B Formal Models*, in "The Journal of Software Testing, Verification and Reliability", to appear, vol. 14, n° 2, 2004.

[22] C. G. ZARBA, D. CANTONE, J. T. SCHWARTZ. *A decision procedure for a sublanguage of set theory involving monotone, additive, and multiplicative functions. I. The Two-Level Case.*, in "Journal of Automated Reasoning", To appear, 2004.

Publications in Conferences and Workshops

[23] M. ABADI, V. CORTIER. *Deciding knowledge in security protocols under equational theories*, in "Proc. 31st Int. Coll. Automata, Languages, and Programming (ICALP'2004), Turku, Finland", D. SANNELLA (editor)., Lecture Notes in Computer Science, Springer, April 2004.

[24] T. ABBES, A. BOUHOULA, M. RUSINOWITCH. *Protocol Analysis in Intrusion Detection Using Decision Tree*, in "Int. Conference on Information Technology: Coding and Computing (ITCC'04), Las Vegas, Nevada", vol. 1, IEEE Computer Society, April 2004, p. 404–408.

[25] T. ABBES, M. RUSINOWITCH. *Fast Multipattern Matching for Intrusion Detection*, in "13th Annual EICAR Conference (European Institute for Computer Anti-Virus Research) CD-rom: Best Paper Proceedings", U. E. GATTIKER (editor)., EICAR Best Paper Proceedings CD-rom (ISBN: 87-987271-6-8), 22 pages., May 2004.

[26] Y. BOICHUT, P.-C. HEAM, O. KOUCHNARENKO, F. OEHL. *Improvements on the Genet and Klay Technique to Automatically Verify Security Protocols*, in "Proc. Int. Workshop on Automated Verification of Infinite-State Systems (AVIS'2004), joint to ETAPS'04, Barcelona, Spain", The final version will be published in ENTCS, Elsevier, 2004, p. 1–11.

[27] F. BOUQUET, F. LEBEAU, B. LEGEARD. *Test case and Test driver generation for automotive embedded systems*, in "5th Int. Conf. on Software Testing, ICS-Test 2004, Düsseldorf, Germany", April 2004, p. 37–53.

[28] F. BOUQUET, B. LEGEARD, F. PEUREUX, E. TORREBORRE. *Mastering Test Generation from Smart Card Software Formal Models*, in "Post proceedings of the Int. Workshop on Construction and Analysis of Safe, Secure and Interoperable Smart devices (CASSIS'04), Marseille, France", LNCS, Selected papers from the CASSIS'04 workshop. To appear, Springer, March 2004.

[29] F. BOUQUET, B. LEGEARD, M. UTTING, N. VACELET. *Faster Analysis of Formal Specification*, in "6th Int. Conf. on Formal Engineering Methods (ICFEM'04), Seattle, USA", LNCS, To appear, Springer-Verlag, November 2004.

[30] Y. CHEVALIER. *A Simple Constraint Combination Procedure for Cryptographic Protocols with Xor*, in "18th Int. Workshop on Unification, Cork, Ireland", M. KOHLHASE (editor)., Long version available as INRIA Research Report RR-5224, July 2004.

[31] Y. CHEVALIER, L. COMPAGNA, J. CUELLAR, P. HANKES DRIELSMA, J. MANTOVANI, S. MÖDERSHEIM, L. VIGNERON. *A High Level Protocol Specification Language for Industrial Security-Sensitive Protocols*, in "Proceedings of Workshop on Specification and Automated Processing of Security Requirements (SAPS), Linz, Austria", September 2004.

[32] Y. CHEVALIER, R. KÜSTERS, M. RUSINOWITCH, M. TURUANI. *Deciding the Security of Protocols with Commuting Public Key Encryption.*, in "Workshop on Automated Reasoning for Security Protocol Analysis - ARSPA'2004, Cork, Ireland", Electronic Notes in Theoretical Computer Science - ENTCS, Jul 2004.

[33] Y. CHEVALIER, L. VIGNERON. *Rule-based Programs describing Internet Security Protocols*, in "5th Int. Workshop on Rule-Based Programming (RULE), Aachen, Germany", S. ABDENNADHER, C. RINGEISSEN (editors)., June 2004.

[34] S. COLIN, F. LEBEAU, B. LEGEARD. *Génération de tests à partir de statecharts fondée sur le calcul de comportements*, in "Congrès Approches Formelles dans l'Assistance au Développement de Logiciels, AFADL'04, Besançon, France", J. JULLIAND (editor)., June 2004, p. 153–167.

[35] J.-F. COUCHOT. *Vérification d'invariant par superposition*, in "MAnifestation de JEunes Chercheurs STIC (MAJECSTIC'04), Calais, France", A paraître, October 2004.

[36] J.-F. COUCHOT, D. DÉHARBE, A. GIORGETTI, S. RANISE. *Barvey: Vérification automatique de consistance de machines abstraites B*, in "Sessions Outils, Congrès Approches Formelles dans l'Assistance au Développement de Logiciels, AFADL'04, Besançon, France", J. JULLIAND (editor)., June 2004, p. 369–372.

[37] J.-F. COUCHOT, F. DADEAU, D. DÉHARBE, A. GIORGETTI, S. RANISE. *Proving and Debugging Set-Based Specifications*, in "ENTCS, proceedings of the Sixth Brazilian Workshop on Formal Methods (WMF'03), Campina Grande, Brazil", A. CAVALCANTI, P. MACHADO (editors)., vol. 95, May 2004, p. 189–208.

[38] J.-F. COUCHOT, A. GIORGETTI. *Analyse d'atteignabilité déductive*, in "Congrès Approches Formelles dans l'Assistance au Développement de Logiciels, AFADL'04, Besançon, France", J. JULLIAND (editor)., June 2004, p. 269–283.

[39] 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.

[40] A. IMINE, D. DÉHARBE, S. RANISE. *Abstraction-Driven Verification of Array Programs*, in "Proceedings of the 7th International Conference on Artificial Intelligence and Symbolic Computation (AISC'04), Linz, Austria", LNCS, n° 3249, Springer, September 2004, p. 271–275.

[41] A. IMINE, G. MOLLI, M. RUSINOWITCH. *Deductive Verification of Distributed Groupware Systems*, in "10th International Conference on Algebraic Methodology And Software Technology, AMAST'04, Stirling, Scotland", C. RATTRAY, S. MAHARAJ (editors)., Lecture Notes in Computer Science, vol. 3116, Springer,

July 2004, p. 226-240.

[42] B. LEGEARD, F. BOUQUET, F. LEBEAU. *Automated Test Generation and Execution for Automative Embedded Software*, in "INCOSE 2004, Annual Int. Symp., 4th European Systems Engineering Conference", June 2004, p. 252–268.

[43] B. LEGEARD, N. KOSMATOV, F. PEUREUX, M. UTTING. *Boundary Coverage Criteria for Test Generation from Formal Models*, in "Proc. of the 15th Int. Symp. on Software Reliability Engineering (ISSRE'04), Saint-Malo, France", To appear, IEEE Computer Society Press, November 2004.

[44] G. OSTER, P. MOLLI, H. SKAF-MOLLI, A. IMINE. *Un modèle sûr et générique pour la synchronisation de données divergentes*, in "Premières Journées Francophones: Mobilité et Ubiquité - UbiMob'04, Nice, France", ISBN: 2.85428.653.7, Cépaduès-Editions, June 2004, p. 98-106.

[45] S. RANISE, C. RINGEISSEN, D.-K. TRAN. *Nelson-Oppen, Shostak and the Extended Canonizer: A Family Picture with a Newborn*, in "First International Colloquium on Theoretical Aspects of Computing - ICTAC 2004, Guiyang, Chine", K. ARAKI, Z. LIU (editors)., Lecture Notes in Computer Science, To appear in post-event proceedings., Springer, September 2004.

[46] M. RUSINOWITCH. *A Decidable Analysis of Security Protocols*, in "18th IFIP World Computer Congress on Theoretical Computer Science - TCS'2004, Toulouse, France", J.-J. LÉVY, E. MAYR, J. MITCHELL (editors)., Kluwer Academic Publishers, August 2004.

[47] J. SANTOS SANTIAGO. *Analyse automatique de protocoles avec AtSe*, in "Congrès Approches Formelles dans l'Assistance au Développement de Logiciels, AFADL'04, Besançon, France", J. JULLIAND (editor)., Présentation système, June 2004.

[48] C. TINELLI, C. G. ZARBA. *Combining Decision Procedures for Sorted Theories*, in "Logics in Artificial Intelligence, 9th European Conference, JELIA 2004, Lisbon, Portugal", J. J. ALFERES, J. A. LEITE (editors)., Lecture Notes in Computer Science, vol. 3229, Springer, 2004, p. 641–653.

[49] L. VIGNERON. *Automatic Verification of Security Protocols*, in "18th Int. Workshop on Unification, Cork, Ireland", M. KOHLHASE (editor)., Invited talk, July 2004.

[50] C. G. ZARBA. *A quantifier elimination algorithm for a fragment of set theory involving the cardinality operator*, in "18th International Workshop on Unification", M. KOHLHASE (editor)., 2004.

Internal Reports

[51] T. ABBES, M. RUSINOWITCH, A. HALOI. *Network Traffic Classification for Intrusion Detection*, Research Report, n° A04-R-225, LORIA, Nancy (France), June 2004.

[52] V. CORTIER, B. WARINSCHI. *Computationally Sound, Automated Proofs for Security Protocols*, Research Report, n° RR-5341, INRIA, October 2004, http://www.inria.fr/rrrt/rr-5341.html.

[53] A. IMINE, G. MOLLI, M. RUSINOWITCH. *Achieving Convergence with Operational Transformation in Distributed Groupware Systems*, Research Report, n° RR-5188, INRIA, 2004, http://www.inria.fr/rrrt/rr-

5188.html.

[54] C. G. ZARBA, D. CANTONE, J. T. SCHWARTZ. *A decision procedure for a sublanguage of set theory involving monotone, additive, and multiplicative functions.*, Technical report, nº RR-5267, INRIA, 2004, http://www.inria.fr/rrrt/rr-5267.html.

[55] C. G. ZARBA. *C-tableaux*, Technical report, nº RR-5229, INRIA, 2004, http://www.inria.fr/rrrt/rr-5229.html.

Bibliography in notes

[56] 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.

[57] 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.

[58] 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.

[59] J.-F. COUCHOT, D. DÉHARBE, A. GIORGETTI, S. RANISE. *Scalable Automated Proving and Debugging of Set-Based Specifications*, in "Journal of the Brazilian Computer Society", ISSN 0104-6500, vol. 9, nº 2, November 2003, p. 17–36, http://lifc.univ-fcomte.fr/~couchot/pubs/cdgr04.ps.gz.

[60] 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.

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

[62] M. HIND. *Pointer Analysis: Haven't We Solved This Problem Yet?*, in "2001 ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE'01), Snowbird, UT", 2001.

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