IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2013

Project-Team CARTE

Theoretical adverse computations, and safety

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

RESEARCH CENTER
Nancy - Grand Est

THEME
Security and Confidentiality

Table of contents

1. Members **1**
2. Overall Objectives **1**
2.1. Introduction 1
2.2. Highlights of the Year 2
3. Research Program **2**
3.1. Computer Virology 2
3.2. Computation over continuous structures 2
3.3. Rewriting 3
4. Application Domains **3**
4.1. Computer Virology 3
4.1.1. The theoretical track. 3
4.1.2. The virus detection track 3
4.1.3. The virus protection track 4
4.1.4. The experimentation track 4
4.2. Computations and Dynamical Systems 4
4.2.1. Continuous computation theories 4
4.2.2. Analysis and verification of adversary systems 4
5. Software and Platforms **5**
5.1. Morphus/MMDEX 5
5.2. TraceSurfer 5
5.3. CROCUS 5
6. New Results **5**
6.1. Computation and Dynamical Systems 5
6.2. Computability, Complexity and Topology 6
6.2.1. Complexity of real functionals 6
6.2.2. Higher-order complexity 6
6.2.3. Irreversible computable functions 6
6.3. Implicit Computational Complexity 7
6.4. Computer Virology 7
6.5. Graph rewriting 8
7. Bilateral Contracts and Grants with Industry **8**
8. Partnerships and Cooperations **8**
8.1. National Initiatives 8
8.1.1. ANR 8
8.1.2. PEPS 8
8.2. European Initiatives 8
8.3. International Initiatives 9
8.4. International Research Visitors 9
8.4.1. Visits of International Scientists 9
8.4.2. Visits to International Teams 9
9. Dissemination **9**
9.1. Scientific Animation 9
9.1.1. Conference organization and program committee 9
9.1.2. Talks 10
9.1.3. Referees 10
9.1.4. Others 10
9.2. Teaching - Supervision - Juries 10
9.2.1. Teaching 10
9.2.2. Supervision 11

9.2.3. Juries 11
9.3. Popularization 12
10. Bibliography **12**

Project-Team CARTE

Keywords: Formal Methods, Security, Virology, Complexity, Model Of Computation, Real Numbers, Dynamical System, Rewriting

Creation of the Project-Team: 2009 January 01.

1. Members

Research Scientists

Isabelle Gnaedig [Inria, Researcher]
Mathieu Hoyrup [Inria, Researcher]
Simon Perdrix [CNRS, Researcher, from Sep 2013]

Faculty Members

Emmanuel Jeandel [Team leader (from July 2013), Univ. Lorraine, Professor, HdR]
Guillaume Bonfante [Univ. Lorraine, Associate Professor, HdR]
Emmanuel Hainry [Univ. Lorraine, Associate Professor]
Jean-Yves Marion [Team leader (until July 2013), Univ. Lorraine, Professor, HdR]
Romain Péchoux [Univ. Lorraine, Associate Professor]

Engineer

Fabrice Sabatier [Inria, granted by FP7 FI WARE project (until September 2013) then by ADT Inria (from October 2013)]

PhD Students

Joan Calvet [Univ. Lorraine, until Aug 2013]
Hugo Férée [Univ. Lorraine]
Hubert Godfroy [Inria, half-funded by DGA, since Oct 2013]
Thanh Dinh Ta [Inria, Cordi-S]
Aurélien Thierry [Inria, Cordi-S]

Post-Doctoral Fellow

Bruno Bauwens [Univ. Lorraine, from Feb 2013]

Visiting Scientists

Neil Jones [Professor, from Mar 2013 until Apr 2013]
Subramanian Kumbakonam Govindarajan [Professor, Feb 2013]

Administrative Assistants

Véronique Constant [Inria]
Martine Kuhlmann [CNRS]
Delphine Hubert [Univ. Lorraine]

Others

Claude-Antoine Loudenot [Intern, Mines Nancy, from Jul 2013 until Sep 2013]
François Menet [Intern, Mines Nancy, from Jul 2013 until Sep 2013]

2. Overall Objectives

2.1. Introduction

The aim of the CARTE research team is to take into account adversity in computations, which is implied by actors whose behaviors are unknown or unclear. We call this notion adversary computation.

The project combines two approaches. The first one is the analysis of the behavior of systems, using tools coming from Continuous Computation Theory. The second approach is to build defenses with tools coming from logic, rewriting and, more generally, from Programming Theory.

The activities of the CARTE team are organized around two research actions:

- Computation over Continuous Structures
- Computer Virology.

2.2. Highlights of the Year

Our team made remarkable progress into the difference between "real world" systems and artefacts due to exact (infinite) precision computations. Olivier Bournez, Daniel Graça and Emmanuel Hainry succeeded in proving an equivalence between robustness and computability: Robust dynamical systems have computable dynamical properties [12], a strong evidence that "real world" systems will not exhibit undecidability properties.

Another highlight of the year is a paper by Hugo Férée, Mathieu Hoyrup and Walid Gomaa, accepted in LICS 2013 [19] that provides a systematic approach to define and analyse the complexity of algorithms acting on infinite precision numbers (infinite words).

3. Research Program

3.1. Computer Virology

From a historical point of view, the first official virus appeared in 1983 on Vax-PDP 11. At the very same time, a series of papers was published which always remains a reference in computer virology: Thompson [74], Cohen [45] and Adleman [34]. The literature which explains and discusses practical issues is quite extensive [49], [51]. However, there are only a few theoretical/scientific studies, which attempt to give a model of computer viruses.

A virus is essentially a self-replicating program inside an adversary environment. Self-replication has a solid background based on works on fixed point in λ-calculus and on studies of von Neumann [78]. More precisely we establish in [41] that Kleene's second recursion theorem [63] is the cornerstone from which viruses and infection scenarios can be defined and classified. The bottom line of a virus behavior is

1. a virus infects programs by modifying them,
2. a virus copies itself and can mutate,
3. it spreads throughout a system.

The above scientific foundation justifies our position to use the word virus as a generic word for self-replicating malwares. There is yet a difference. A malware has a payload, and virus may not have one. For example, worms are an autonous self-replicating malware and so fall into our definition. In fact, the current malware taxonomy (virus, worms, trojans, ...) is unclear and subject to debate.

3.2. Computation over continuous structures

Classical recursion theory deals with computability over discrete structures (natural numbers, finite symbolic words). There is a growing community of researchers working on the extension of this theory to continuous structures arising in mathematics. One goal is to give foundations of numerical analysis, by studying the limitations of machines in terms of computability or complexity, when computing with real numbers. Classical questions are : if a function $f : \mathbb{R} \rightarrow \mathbb{R}$ is computable in some sense, are its roots computable? in which time? Another goal is to investigate the possibility of designing new computation paradigms, transcending the usual discrete-time, discrete-space computer model initiated by the Turing machine that is at the base of modern computers.

While the notion of a computable function over discrete data is captured by the model of Turing machines, the situation is more delicate when the data are continuous, and several non-equivalent models exist. In this case, let us mention computable analysis, which relates computability to topology [48], [77]; the Blum-Shub-Smale model (BSS), where the real numbers are treated as elementary entities [40]; the General Purpose Analog Computer (GPAC) introduced by Shannon [72]with continuous time.

3.3. Rewriting

The rewriting paradigm is now widely used for specifying, modelizing, programming and proving. It allows to easily express deduction systems in a declarative way, and to express complex relations on infinite sets of states in a finite way, provided they are countable. Programming languages and environments with a rewriting based semantics have been developed ; see ASF+SDF [42], MAUDE [44], and TOM [69].

For basic rewriting, many techniques have been developed to prove properties of rewrite systems like confluence, completeness, consistency or various notions of termination. Proof methods have also been proposed for extensions of rewriting such as equational extensions, consisting of rewriting modulo a set of axioms, conditional extensions where rules are applied under certain conditions only, typed extensions, where rules are applied only if there is a type correspondence between the rule and the term to be rewritten, and constrained extensions, where rules are enriched by formulas to be satisfied [36], [47], [73].

An interesting aspect of the rewriting paradigm is that it allows automatable or semi-automatable correctness proofs for systems or programs: the properties of rewriting systems as those cited above are translatable to the deduction systems or programs they formalize and the proof techniques may directly apply to them.

Another interesting aspect is that it allows characteristics or properties of the modelled systems to be expressed as equational theorems, often automatically provable using the rewriting mechanism itself or induction techniques based on completion [46]. Note that the rewriting and the completion mechanisms also enable transformation and simplification of formal systems or programs.

Applications of rewriting-based proofs to computer security are various. Approaches using rule-based specifications have recently been proposed for detection of computer viruses [75], [76]. For several years, in our team, we have also been working in this direction. We already have proposed an approach using rewriting techniques to abstract program behaviors for detecting suspicious or malicious programs [37].

4. Application Domains

4.1. Computer Virology

4.1.1. The theoretical track.

It is rightful to wonder why there is only a few fundamental studies on computer viruses while it is one of the important flaws in software engineering. The lack of theoretical studies explains maybe the weakness in the anticipation of computer diseases and the difficulty to improve defenses. For these reasons, we do think that it is worth exploring fundamental aspects, and in particular self-reproducing behaviors.

4.1.2. The virus detection track

The crucial question is how to detect viruses or self-replicating malwares. Cohen demonstrated that this question is undecidable. The anti-virus heuristics are based on two methods. The first one consists in searching for virus signatures. A signature is a regular expression, which identifies a family of viruses. There are obvious defects. For example, an unknown virus will not be detected, like ones related to a 0-day exploit. We strongly suggest to have a look at the independent audit [50] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [52], this kind of methods is not yet really implemented. Moreover, the large number of false-positive implies this is barely usable. To end this short survey, intrusion detection encompasses virus detection. However, unlike computer virology, which has a solid scientific foundation as we have seen, the IDS notion of “malwares” with respect to some security policy is not well defined. The interested reader may consult [70].

4.1.3. The virus protection track

The aim is to define security policies in order to prevent malware propagation. For this, we need (i) to define what is a computer in different programming languages and setting, (ii) to take into consideration resources like time and space. We think that formal methods like rewriting, type theory, logic, or formal languages, should help to define the notion of a formal immune system, which defines a certified protection.

4.1.4. The experimentation track

This study on computer virology leads us to propose and construct a “high security lab” in which experiments can be done in respect with the French law. This project of “high security lab” in one of the main project of the CPER 2007-2013.

4.2. Computations and Dynamical Systems

4.2.1. Continuous computation theories

Understanding computation theories for continuous systems leads to studying hardness of verification and control of these systems. This has been used to discuss problems in fields as diverse as verification (see e.g. [35]), control theory (see e.g. [43]), neural networks (see e.g. [71]), and so on. We are interested in the formal decidability of properties of dynamical systems, such as reachability [62], the Skolem-Pisot problem [39], the computability of the ω-limit set [61]. Those problems are analogous to verification of safety properties. Contrary to computability theory, complexity theory over continuous spaces is underdeveloped and not well understood. A central issue is the choice of the representation of objects by discrete data and its effects on the induced complexity notions. As for computability, it is well known that a representation is gauged by the topology it induces. However more structure is needed to capture the complexity notions: topologically equivalent representations may induce different classes of polynomial-time computable objects, e.g. Developing a sound complexity theory over continuous structures would enable us to make abstract computability results more applicable by analysing the corresponding complexity issues. We think that the preliminary step towards such a theory is the development of higher-order complexity, which we are currently carrying out. In contrast with the discrete setting, it is of utmost importance to compare the various models of computation over the reals, as well as their associated complexity theories. In particular, we focus on the General Purpose Analog Computer of Claude Shannon [72], on recursive analysis [77], on the algebraic approach [68] and on computability in a probabilistic context [64]. A crucial point for future investigations is to fill the gap between continuous and discrete computational models. This is one deep motivation of our work on computation theories for continuous systems.

4.2.2. Analysis and verification of adversary systems

The other research direction on dynamical systems we are interested in is the study of properties of adversary systems or programs, i.e. of systems whose behavior is unknown or indistinct, or which do not have classical expected properties. We would like to offer proof and verification tools, to guarantee the correctness of such systems. On one hand, we are interested in continuous and hybrid systems. In a mathematical sense, a hybrid system can be seen as a dynamical system, whose transition function does not satisfy the classical regularity hypotheses, like continuity, or continuity of its derivative. The properties to be verified are often expressed as reachability properties. For example, a safety property is often equivalent to (non-)reachability of a subset of unsure states from an initial configuration, or to stability (with its numerous variants like asymptotic stability, local stability, mortality, etc ...). Thus we will essentially focus on verification of these properties in various classes of dynamical systems.

We are also interested in rewriting techniques, used to describe dynamic systems, in particular in the adversary context. As they were initially developed in the context of automated deduction, the rewriting proof techniques, although now numerous, are not yet adapted to the complex framework of modelization and programming. An important stake in the domain is then to enrich them to provide realistic validation tools, both in providing finer rewriting formalisms and their associated proof techniques, and in developing new validation concepts in the adversary case, i.e. when usual properties of the systems like, for example, termination are not verified.

For several years, we have been developing specific procedures for property proofs of rewriting, for the sake of programming, in particular with an inductive technique, already applied with success to termination under strategies [53], [54], [55], to weak termination [56], sufficient completeness [57] and probabilistic termination [59]. The last three results take place in the context of adversary computations, since they allow for proving that even a divergent program, in the sense where it does not terminate, can give the expected results. A common mechanism has been extracted from the above works, providing a generic inductive proof framework for properties of reduction relations, which can be parametrized by the property to be proved [58], [60]. Provided program code can be translated into rule-based specifications, this approach can be applied to correctness proof of software in a larger context. A crucial element of safety and security of software systems is the problem of resources. We are working in the field of Implicit Computational Complexity. Interpretation based methods like Quasi-interpretations (QI) or sup-interpretations, are the approach we have been developing these last years [65], [66], [67]. Implicit complexity is an approach to the analysis of the resources that are used by a program. Its tools come essentially from proof theory. The aim is to compile a program while certifying its complexity.

5. Software and Platforms

5.1. Morphus/MMDEX

MMDEX is a virus detector based on morphological analysis. It is composed of our own disassembler tool, on a graph transformer and a specific tree-automaton implementation. The tool is used in the EU-Fiware project and by some other partners (e.g. DAVFI project).
Written in C, 20k lines.
APP License, IDDN.FR.001.300033.000.R.P.2009.000.10000, 2009.

5.2. TraceSurfer

TraceSurfer is a self-modifying code analyzer coming with an IDA add-on. It works as a wave-builder. In the analysis of self-modifying programs, one basic task is indeed to separate parts of the code which are self-modifying into successive layers, called waves. TraceSurfer extracts waves from traces of program executions. Doing so drastically simplifies program verification.
Written in C, 5k lines.
http://code.google.com/p/tartetatintools/

5.3. CROCUS

CROCUS is a program interpretation synthetizer. Given a first order program (possibly written in OCAML), it outputs a quasi-interpretation based on max, addition and product. It is based on a random algorithm. The interpretation is actually a certificate for the program's complexity. Users are non academics (some artists).
Written in Java, 5k lines.

6. New Results

6.1. Computation and Dynamical Systems

In [12], we analyzed the power of dynamical system that are robust to infinitesimal perturbations. While previous works on this question were limited to very specific kinds of systems such as piecewise constant derivative systems, we obtained results for a quite general class of systems: the main hypothesis being smoothness (which is already a prerequisite in systems that perform analog computation). We show that if a system is robust, then the language it recognizes is computable, and the converse: all computable languages can be recognized by a robust smooth system. Those results are true for discrete-time as well as continuous-time dynamical systems on bounded or unbounded domains.

We investigated in [23], [15], [33] the isomorphism (conjugacy) problem for dynamical systems. While the decidability in the one-dimensional case is a long-standing open problem, we characterize its exact complexity [23] in higher dimensions. Our result suggest that the isomorphism problem is easier than the factoring and embedding problem (decide if one dynamical system is a subsystem of another). A traditional approach to prove two dynamical systems are not isomorphic is to prove that they have different dynamical invariants. We characterised in terms of complexity and computability classes different well known dynamic invariants (periodic points, Turing degrees) in [23], [33].

While Turing machines are usually used for computing, it is an interesting model of dynamical systems, which looks very much like two-dimensional piecewise-affine maps. We investigated dynmacial invariants (entropy and Lyapunov exponents) for Turing machines, and proved quite surprisingly that they are computable. Essentially this means that Turing machines that do interesting computations must do it so slowly that this cannot be seen in their dynamics. This work will be presented in STACS 2014

6.2. Computability, Complexity and Topology

6.2.1. Complexity of real functionals

Computability and topology are closely related as computability assumptions impose topological restrictions: on a topological space, computable functions are continuous and continuous functions are computable relative to some oracle. In the same way, complexity assumptions as bounds on the computation time impose analytical restrictions, but in a way that is not understood yet. For functions from the real numbers to the real numbers, it is known that polynomial-time computable functions correspond to functions with a polynomial modulus of continuity. However for functions on other spaces no such correspondence is known. We investigate the particular case of norms on the space of continuous real functions defined on the unit interval. We introduce analytical characteristics of a norm, namely its dependency on points and the concept of *relevant points*, and use them to characterize the polynomial-time computable norms. This work was presented at LICS 2013 [19]. A full version including other results on non-deterministic complexity classes is currently submitted [28].

6.2.2. Higher-order complexity

While computability theory is well-developed and understood on large classes of topological spaces, complexity theory in analysis is still in its infancy. We argue that the usual way of representing mathematical objects by functions from finite strings to finite strings (order 1 functions) is not appropriate for general spaces. We show that as soon as the space becomes large in a topological sense, it cannot be represented by order 1 functions in a way that respects complexity notions, so we propose to represent objects using higher order functions over finite strings. However higher order complexity theory is not well-understood. The only known class to date is BFF, the class of Basic Feasible Functionals, which does not enjoy nice properties: some intuitively feasible functionals do not belong to the class. We develop a new way of carrying out complexity theory at higher order types, using an adaptation of game semantics. A preliminary version of this work was presented at CCA 2013 [26].

6.2.3. Irreversible computable functions

As mentioned before, computable functions must be continuous. It gives a simple way of proving that some operator is not computable by showing that it is discontinuous. We recall that a function f is computable if there is a *single* oracle Turing machine M that on each x given by an oracle, computes $f(x)$. The following weaker notion is also interesting: a function f *preserves computability* if for each computable x, $f(x)$ is computable. Preservation of computability no more implies continuity, so there is no topological argument to show that some operator does not preserve computability. We develop a strong notion of discontinuity and prove a general result stating that this notion of discontinuity prevents preservation of computability. We apply this result to solve an open problem about the non-computability of the ergodic decomposition. We show that many classical constructions in computability theory are instances of our result. Hence we exhibit deeper connections between computability and topology. The work has been accepted at STACS 2014 [22]. A partial result was published in [13].

6.3. Implicit Computational Complexity

In the setting of non-interference and implicit computational complexity, Emmanuel Hainry, Jean-Yves Marion, and Romain Péchoux presented a characterization of FPSPACE in a language with a fork/wait mechanism [20]. The language used in this work is a classical imperative language with while loops complemented with a mechanism to launch new processes through forks. The fork instruction is heavily inspired by C's fork/wait construction for Unix operating systems, which anchors this work in a down-to-earth setting. Using a type system that enforces a data-ramification on variables, they show that all programs that can be typed and are terminating compute an FPSPACE function, that with a natural evaluation strategy, they indeed use only polynomial space, and conversely that this type system is complete as all FPSPACE functions can be implemented in this language in a typable way.

Emmanuel Hainry and Romain Péchoux also used data-ramification combined with non-interference principles to effectively bound the memory used by object oriented languages in [21]. This work introduces a type system for an object oriented language (derived from java). This type system allows to compute polynomial bounds on the heap and stack used by a typable program, ensuring that if the program halts, it will only use memory under this explicit bound. As the typing procedure is doable in time polynomial in the size of the program, those bounds are easy to obtain, though not tight. Interesting features of this work include inheritance (with overloading and overriding) and, the ability to analyze programs with flow statements controled by objects, contrary to most other works in implicit computational complexity. In [24], Romain Péchoux has shown that the notion of (polynomial) interpretation over term rewrite systems can be adapated on a process language, a variant of the pi-calculus with process recursive definitions. This work shows that the order induced by simulation can be used wrt a given process semantics to infer time and space upper bounds on process resource usage (reduction length, size of sent values, ...).

6.4. Computer Virology

The study on behavioural malware detection has been continued. Guillaume Bonfante, Isabelle Gnaedig and Jean-Yves Marion have been developing an approach detecting suspicious schemes on an abstract representation of the behavior of a program, by abstracting program traces, rewriting given subtraces into abstract symbols representing their functionality. Considering abstract behaviors allows us to be implementation-independent and robust to variants and mutations of malware. Suspicious behaviors are then detected by comparing trace abstractions to reference malicious behaviors.

Model checking is a strong point of our approach: the predefined behavior patterns, used to abstract program traces, are defined by first order temporal logic formulas, as well as the reference suspicious behaviors, given in a signature. The infection problem can then be seen as the satisfaction problem of the formula of the signature by an abstracted trace of the program, which can be checked using existing model checking techniques

The previous work by the team involved abstracting trace automata by rewriting them with respect to a set of predefined behavior patterns defined as a regular language described by a string rewriting system [37], and then, by a term rewriting system [38], which allows to detect information leak.

This work has been finished this year by designing a probabilistic generalization of our approach. Introducing probabilities in our technique allows to express a pertinence degree of detection when analysis of the program results in an incomplete or uncertain program dataflow, or when abstraction cannot be performed reliably. Proposing malware detection with a probabilistic rate is finer and more realistic in practice than giving the binary answer of whether a program is infected or not.

Using a tropical semiring over the reals, they have presented a formalism relying on a weighted term rewriting mechanism, where a weight w, naturally associated to a probability p by the formula: $w = -log(p)$, represents the probability that the realized abstraction be right.

Detection of an abstract behavior has then be defined with respect to a threshold, and a program P exhibits an abstract behavior M if and only if one of its traces admits an abstract form realizing M with a weight not exceeding this threshold.

The weighted abstraction formalism has the advantage of providing a detection algorithm with the same complexity as in the unweighted case, that is linear in the size of the trace automaton [27].

6.5. Graph rewriting

Guillaume Bonfante and Bruno Guillaume provide a new graph rewriting framework adapted to Natural Language Processing. It involves a new form of edge transformation. A new termination technique is also described. The extended paper [17] is accepted for publication in Mathematical Structure in Computer Science.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

We are currently working with the consortium “malware.lu”.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

- The team was a funding parter in ANR Complice (Implicit Computational Complexity, Concurrency and Extraction), ref.: ANR-08-BLANC-0211-01, that ended in april 2013 and whose aim was to extend the results of ICC to other paradigms (process languages, ...) and take benefice of proof extraction techniques in order to synthesize resoure certificates. This ANR should be followed by a new ANR submission (ANR Elica proposal) involving Paris 7 PPS team, Paris 13 LCC team, ENS Lyon Plume team and Bologna Inria team Focus.
- The team is a funding parter in ANR Binsec, whose aim is to fill part of the gap between formal methods over executable code on one side, and binary-level security analyses currently used in the security industry. Two main applicative domains are targeted: vulnerability analysis and virus detection. Two other closely related applications will also be investigated: crash analysis and program deobfuscation.
- Emmanuel Jeandel is a member of ANR Blanche ANR-09-BLAN-0164 (EMC: *Emerging Phenomena in Computation Models*), that ended in April 2013.

8.1.2. PEPS

- Simon Perdrix is a member of a PEPS INS2I “Information et Communication Quantique: Cryptographie et Calcul Quantiques Distribués.“ with partners in Telecom ParisTech and other labs.
- Mathieu Hoyrup is principal investigator of a PEPS INS2I “Approches Topologiques de l’Information et de la Calculabilité”, with Emmanuel Jeandel and Laurent Bienvenu (CNRS, LIAFA).

8.2. European Initiatives

8.2.1. FP7 Projects

8.2.1.1. FI-WARE

Title: Morphus

Type: COOPERATION

Defi: PPP FI: Technology Foundation: Future Internet Core Platform

Instrument: Integrated Project (IP)

Objectif: PPP FI: Technology Foundation:Future Internet Core Platform

Duration: September 2011 - May 2014

Coordinator: Telefonica (Spain)

Other Partners: Thales, SAP, Inria

Inria contact: Olivier Festor

Abstract: See also: http://www.fi-ware.eu/. FI-WARE will deliver a novel service infrastructure, building upon elements (called Generic Enablers) which offer reusable and commonly shared functions making it easier to develop Future Internet Applications in multiple sectors. This infrastructure will bring significant and quantifiable improvements in the performance, reliability and production costs linked to Internet Applications for building a true foundation for the Future Internet.

8.3. International Initiatives

8.3.1. Inria International Partners

8.3.1.1. Informal International Partners

The team has an informal partnership with Pr. James Royer (University of Syracuse) and PhD. Norman Danner (Wesleyan University) on the study of program higher order complexity (an Inria associated team proposal has been submitted on this domain). On the Implicit Computational Complexity part, the team has strong contacts with Universita di Torino (Pr Simona Ronchi Della Rocca), Dundee University(PhD Marco Gaboardi), Universita di Bologna (Pr Simone Martini and PhD Ugo Dal Lago).

8.4. International Research Visitors

8.4.1. Visits of International Scientists

Subramanian Kumbakonam Govindarajan, professor in Universiti Sains Malaysia, was visiting Carte team in february. He works on computational models and Parikh matrices.

Neil Jones, professor in the University of Copenhagen, visited Carte team for one month in March. He is currently working on program transformation and program obfuscation, which have obvious applications to Computer Virology.

8.4.2. Visits to International Teams

Mathieu Hoyrup visited Universidad Andres Bello in Santiago de Chile during february. He worked there with Cristobal Rojas on extending the results [22] from functions to relations.

9. Dissemination

9.1. Scientific Animation

9.1.1. Conference organization and program committee

Mathieu Hoyrup was in the Program Committee of Computability, Complexity and Randomness (CCA) 2013. He is guest editor for the post-proceedings of CCA 2013, special issue of Logical Methods in Computer Science. He has been invited to organize the special session Algorithmic Randomness of the conference Computability in Europe (CiE) 2013.

Simon Perdrix is a member of the Program Committee of the first Workshop on Parallel Quantum Computing (ParQ).

Guillaume Bonfante was in the Program Committee of Malware 2013 and Symposium on Foundations & Practice of Security (FPS) 2013.

The Carte Team has organized a few conferences and workshops in Nancy this year:

- Journées Calculabilités 2013, April
- Analysis, Randomness and Applications (ARA) 2013, June.
- Computability and Complexity in Analysis (CCA) 2013, July.
- Journées Informatique Quantique, October.

9.1.2. Talks

Emmanuel Jeandel gave an invited Talk at the conference (Computability, Complexity and Randomness (CCR) 2013) on entropy of Turing machines. He also presented his work on multidimensional symbolic dynamics in the PIMS Workshop on Automata Theory and Symbolic Dynamics.

Guillaume Bonfante was invited to present his work to workshop "Proof Theory and Rewriting" (Kanazawa, february) and "Journées Francophones d'Investigation Numérique" (Neuchâtel, october).

Guillaume Bonfante, Hugo Férée and Jean-Yves Marion were invited to workshop "Advances in implicit computational complexity" in Shonan Village in november.

Mathieu Hoyrup, Emmanuel Hainry, Emmanuel Jeandel, were invited to present their work to (workshop DySyCo) in Lyon, december 2013.

Emmanuel Hainry presented *Complexité d'ordre supérieur, de l'Analyse Récursive aux Basic Feasible Functionals* at the Séminaire d'algorithmique et de complexité du plateau de Saclay (http://www.lix.polytechnique.fr/~bodirsky/seminaire/) in LIX, Palaiseau in June 2013.

9.1.3. Referees

Emmanuel Hainry reviewed articles for the journal *Computability*, for *SIAM Journal on Computing*, and for the *STACS 2014* conference.

Emmanuel Jeandel reviewed articles for the *LICS 2013* and the *STACS 2014* conference.

Romain Péchoux reviewed articles for the *WORDS 2013* conference

Mathieu Hoyrup reviewed articles for the CiE 2013, STACS 2014, STOC 2014 conferences and the journals Logical Methods in Computer Science and Theory of Computing Systems.

9.1.4. Others

Isabelle Gnaedig is member of the scientific mediation committee of Inria Nancy Grand-Est and researcher social referee at Inria Nancy-Grand Est.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Unless specified otherwise, all teaching is done at Université de Lorraine, France.

Licence :

- Guillaume Bonfante
  - Java, L3, Mines Nancy
- Emmanuel Hainry
  - Operating Systems, 60 hours, L1, IUT Nancy Brabois
  - Algorithms and Programs, 60 hours, L1, IUT Nancy Brabois
  - Object Oriented Programming, 24 hours, L1, IUT Nancy Brabois
  - Databases, 24 hours, L2, IUT Nancy Brabois

  - Complexity, 28 hours, L1, IUT Nancy Brabois
  - Algorithmics, 12 hours, DU PFST (eq. L1), IUT Nancy Brabois
- Emmanuel Jeandel
  - Statistics for Computer Science, 46 hours, L3 Informatique
  - Linear Programming, 46 hours, L3 Informatique
  - Algorithmics and Programming 1, 60 hours, L1 Maths-Info
  - Algorithmics and Programming 4, 30 hours, L3 Informatique
  - Networking, 20 hours, L2 and L3 Informatique
- Romain Péchoux
  - Introduction to OO programming, 55 hours, L3 MIASHS parcours MIAGE.
  - Databases, 42 hours, L3 SG, ISAM-IAE
  - Propositional logic, 35 hours, L1 MIASHS
  - Algorithmic complexity, 30 hours, L3 MIASHS parcours MIAGE, IGA Casablanca, Marocco.

Master

- Guillaume Bonfante
  - Modelling and UML, M1, Mines Nancy
  - Video Games, M1, Mines Nancy
  - Semantics, M1, Mines Nancy
  - Safety of Software, M2, Mines Nancy
- Isabelle Gnaedig
  - Design of Safe Software, Coordination of the module, M2, Telecom-Nancy
  - Rule-based Programming, 20 hours, M2, Telecom-Nancy
- Emmanuel Jeandel
  - Algorithmics and Complexity, M1 Informatique and M1 ENSEM, 60 hours
  - Combinatorial Optimization, M1 Informatique, 30 hours.
- Romain Péchoux
  - Mathematics for computer science, 20 hours, M1 SC
  - Advanced Java, 52 hours, M1 MIAGE
- Simon Perdrix
  - Pépites Algorithmiques — Informatique Quantique, 14 hours, M1/M2, Ecole des Mines de Nancy.

9.2.2. Supervision

PhD : Joan Calvet, Analyse Dynamique de Logiciels Malveillants, Université de Lorraine and Ecole Polytechnique de Montreal, defended August 23rd, supervised by Jean-Yves Marion and José M. Fernandez.

PhD in progress: David Cattanéo, Combinatorial Modelization in Quantum Computation and Generalized Cover Problems, started sept. 2012, Pablo Arrighi (director), Simon Perdrix (co-advisor)

PhD in progress: Hugo Férée, Computational Complexity in Analysis, defense planned in December 2014, Jean-Yves Marion (director) and Mathieu Hoyrup (co-advisor).

PhD in progress: Hubert Godfroy, Semantics of Self-modifying Programs, Jean-Yves Marion

PhD in progress: Jérôme Javelle , Quantum Cryptography: Protocols and Graphs, started Jan. 2011, Pablo Arrighi (director), Mehdi Mhalla (co-advisor), Simon Perdrix (co-advisor)

PhD in progress: Thanh Dinh Ta, Malware Algebraic Modeling and Detection, started Sept. 2010, Jean-Yves Marion (director) and Guillaume Bonfante (co-advisor)

PhD in progress: Aurélien Thierry, Morphological Analysis of Malware, started Oct. 2011 supervised by Jean-Yves Marion.

9.2.3. Juries

Isabelle Gnaedig:

- participation to the Telecom-Nancy admission committee.

Emmanuel Jeandel

- Selection committee for a research assistant position in Nice (MCF 1114).
- Jury of Razvan Barbulescu's PhD Defense on "Algorithmes de logarithmes discrets dans les corps finis", defended in Université de Lorraine, December 5th.

9.3. Popularization

Isabelle Gnaedig is member of the scientific vulgarization committee of Inria Nancy Grand-Est. This committee is a choice and guidance instance helping the direction of the center and the person in charge of popularization events, to elaborate a strategy, to realize events and to help researchers to get involved in various actions aiming at popularizing our research themes, and more generally computer science and mathematics.

This year, in particular, the center participated to organization of mathematics competitions and projects for high school students, to conferences for computer science high school teachers, to the "Fête de la Science", to the "Moments d'invention" exhibition of the "Nancy Renaissance" event, and received several high school classes in various research teams of Inria Nancy Grand-Est. Details can be found at https://iww.inria.fr/NanSciNum/#.UsGBEWTuKY8.

10. Bibliography

Major publications by the team in recent years

[1] G. Bonfante, J.-Y. Marion, J.-Y. Moyen. *Quasi-interpretations a way to control resources*, in "Theoretical Computer Science", May 2011, vol. 412, n^{o} 25, pp. 2776-2796 [*DOI :* 10.1016/j.tcs.2011.02.007], http://hal.inria.fr/hal-00591862/en

[2] O. Bournez, D. Graça, E. Hainry. *Computation with perturbed dynamical systems*, in "Journal of Computer and System Sciences", August 2013, vol. 79, n^{o} 5, pp. 714-724 [*DOI :* 10.1016/j.jcss.2013.01.025], http://hal.inria.fr/hal-00861041

[3] J. Calvet, J. Fernandez, J.-Y. Marion. *The case for in-the-lab botnet experimentation: creating and taking down a 3000-node botnet*, in "Annual Computer Security Applications Conference", Austin, Texas États-Unis, 12 2010, http://hal.inria.fr/inria-00536706/en/

[4] H. Férée, E. Hainry, M. Hoyrup, R. Péchoux. *Interpretation of stream programs: characterizing type 2 polynomial time complexity*, in "21st International Symposium on Algorithms and Computation - ISAAC 2010", Corée, République De Jeju Island, Springer, Dec 2010, http://hal.inria.fr/inria-00518381

[5] H. Férée, M. Hoyrup, W. Gomaa. *On the query complexity of real functionals*, in "LICS - 28th ACM/IEEE Symposium on Logic in Computer Science", New Orleans, United States, January 2013, pp. 103-112 [*DOI :* 10.1109/LICS.2013.15], http://hal.inria.fr/hal-00773653

[6] I. Gnaedig, H. Kirchner. *Proving Weak Properties of Rewriting*, in "Theoretical Computer Science", 2011, vol. 412, pp. 4405-4438 [*DOI :* 10.1016/j.tcs.2011.04.028], http://hal.inria.fr/inria-00592271/en

[7] E. Hainry, J.-Y. Marion, R. Péchoux. *Type-based complexity analysis for fork processes*, in "16th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)", Rome, Italy, F. Pfenning (editor), Lecture Notes in Computer Science, Springer, 2013, vol. 7794, pp. 305-320 [*DOI :* 10.1007/978-3-642-37075-5_20], http://hal.inria.fr/hal-00755450

[8] M. HOYRUP. *The dimension of ergodic random sequences*, in "STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)", Paris, France, C. DÜRR, T. WILKE (editors), LIPIcs, March 2012, vol. 14, pp. 567-576, http://hal.inria.fr/hal-00678161

[9] E. JEANDEL, P. VANIER. *Hardness of Conjugacy, Embedding and Factorization of multidimensional Subshifts of Finite Type*, in "STACS - 30th International Symposium on Theoretical Aspects of Computer Science", Kiel, Germany, N. PORTIER, T. WILKE (editors), Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, February 2013, vol. 20, pp. 490–501 [*DOI :* 10.4230/LIPICS.STACS.2013.490], http://hal.inria.fr/hal-00840384

[10] J.-Y. MARION. *A type system for complexity flow analysis*, in "Twenty-Sixth Annual IEEE Symposium on Logic in Computer Science - LICS 2011", Toronto, Canada, ACM, June 2011, pp. 1–10, http://hal.inria.fr/hal-00591853/en

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] J. CALVET. , *Analyse Dynamique de Logiciels Malveillants*, Université de Lorraine, August 2013, http://hal.inria.fr/tel-00922384

Articles in International Peer-Reviewed Journals

[12] O. BOURNEZ, D. GRAÇA, E. HAINRY. *Computation with perturbed dynamical systems*, in "Journal of Computer and System Sciences", August 2013, vol. 79, n^{o} 5, pp. 714-724 [*DOI :* 10.1016/J.JCSS.2013.01.025], http://hal.inria.fr/hal-00861041

[13] M. HOYRUP. *Computability of the ergodic decomposition*, in "Annals of Pure and Applied Logic", May 2013, vol. 164, n^{o} 5, pp. 542-549 [*DOI :* 10.1016/J.APAL.2012.11.005], http://hal.inria.fr/hal-00746473

[14] E. JEANDEL, G. THEYSSIER. *Subshifts as models for MSO logic*, in "Information and Computation", 2013, n^{o} 225, pp. 1-15 [*DOI :* 10.1016/J.IC.2013.01.003], http://hal.inria.fr/hal-00783099

[15] P. VANIER, E. JEANDEL. *Turing degrees of multidimensional SFTs*, in "Journal of Theoretical Computer Science (TCS)", November 2013, http://dx.doi.org/10.1016/j.tcs.2012.08.027, http://hal.inria.fr/hal-00613165

International Conferences with Proceedings

[16] B. BAUWENS. *Asymmetry of the Kolmogorov complexity of online predicting odd and even bits*, in "STACS - 31th Symposium on Theoretical Aspects of Computer Science - 2014", Lyon, France, March 2014, http://hal.inria.fr/hal-00920894

[17] G. BONFANTE, B. GUILLAUME. *Non-simplifying Graph Rewriting Termination*, in "TERMGRAPH", Rome, Italy, R. ECHAHED, D. PLUMP (editors), 2013, pp. 4-16, http://hal.inria.fr/hal-00921053

[18] G. BONFANTE, J.-Y. MARION, F. SABATIER, A. THIERRY. *Analysis and Diversion of Duqu's Driver*, in "Malware 2013 - 8th International Conference on Malicious and Unwanted Software", Fajardo, Puerto Rico, IEEE, October 2013, http://hal.inria.fr/hal-00925517

[19] H. FÉRÉE, M. HOYRUP, W. GOMAA. *On the query complexity of real functionals*, in "LICS - 28th ACM/IEEE Symposium on Logic in Computer Science", New Orleans, United States, January 2013, pp. 103-112 [*DOI :* 10.1109/LICS.2013.15], http://hal.inria.fr/hal-00773653

[20] E. HAINRY, J.-Y. MARION, R. PÉCHOUX. *Type-based complexity analysis for fork processes*, in "16th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)", Rome, Italy, F. PFENNING (editor), Lecture Notes in Computer Science, Springer, 2013, vol. 7794, pp. 305-320 [*DOI :* 10.1007/978-3-642-37075-5_20], http://hal.inria.fr/hal-00755450

[21] E. HAINRY, R. PÉCHOUX. *Types for controlling heap and stack in Java*, in "Third International Workshop on Foundational and Practical Aspects of Resource Analysis (FOPARA)", Bertinoro, Italy, Ugo Dal Lago and Ricardo Pena, August 2013, http://hal.inria.fr/hal-00910166

[22] M. HOYRUP. *Irreversible computable functions*, in "STACS - 31st Symposium on Theoretical Aspects of Computer Science - 2014", Lyon, France, March 2014, http://hal.inria.fr/hal-00915952

[23] E. JEANDEL, P. VANIER. *Hardness of Conjugacy, Embedding and Factorization of multidimensional Subshifts of Finite Type*, in "STACS - 30th International Symposium on Theoretical Aspects of Computer Science", Kiel, Germany, N. PORTIER, T. WILKE (editors), Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, February 2013, vol. 20, pp. 490–501 [*DOI :* 10.4230/LIPICS.STACS.2013.490], http://hal.inria.fr/hal-00840384

[24] R. PÉCHOUX. *Bounding Reactions in the Pi-calculus using Interpretations*, in "Third International Workshop on Foundational and Practical Aspects of Resource Analysis (FOPARA 2013)", Bertinoro, Italy, Ugo Dal Lago and Ricardo Pena, August 2013, http://hal.inria.fr/hal-00910170

National Conferences with Proceedings

[25] G. BONFANTE, J.-Y. MARION, F. SABATIER, A. THIERRY. *Duqu contre Duqu : Analyse et détournement du driver de Duqu*, in "SSTIC - Symposium sur la sécurité des technologies de l'information et des communications", Rennes, France, June 2013, http://hal.inria.fr/hal-00925184

Conferences without Proceedings

[26] H. FÉRÉE, M. HOYRUP. *Higher-order complexity in analysis*, in "CCA - 10th International Conference on Computability and Complexity in Analysis - 2013", Nancy, France, July 2013, http://hal.inria.fr/hal-00915973

Research Reports

[27] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. , *Behavior Analysis of Malicious Code by Weighted Behavior Abstraction*, March 2013, http://hal.inria.fr/hal-00803412

[28] H. FÉRÉE, W. GOMAA, M. HOYRUP. , *Analytical properties of resource-bounded real functionals*, July 2013, 33 p. , http://hal.inria.fr/hal-00848482

[29] A. ROUSSEAU, A. DARNAUD, B. GOGLIN, C. ACHARIAN, C. LEININGER, C. GODIN, C. HOLIK, C. KIRCHNER, D. RIVES, E. DARQUIE, E. KERRIEN, F. NEYRET, F. MASSEGLIA, F. DUFOUR, G. BERRY, G. DOWEK, H. ROBAK, H. XYPAS, I. ILLINA, I. GNAEDIG, J. JONGWANE, J. EHREL, L. VIENNOT, L. GUION, L. CALDERAN, L. KOVACIC, M. COLLIN, M.-A. ENARD, M.-H. COMTE, M. QUINSON, M. OLIVI, M. GIRAUD, M. DORÉMUS, M. OGOUCHI, M. DROIN, N. LACAUX, N. ROUGIER, N. ROUSSEL,

P. GUITTON, P. PETERLONGO, R.-M. CORNUS, S. VANDERMEERSCH, S. MAHEO, S. LEFEBVRE, S. BOLDO, T. VIÉVILLE, V. POIREL, A. CHABREUIL, A. FISCHER, C. FARGE, C. VADEL, I. ASTIC, J.-P. DUMONT, L. FÉJOZ, P. RAMBERT, P. PARADINAS, S. DE QUATREBARBES, S. LAURENT. , *Médiation Scientifique : une facette de nos métiers de la recherche*, March 2013, 34 p. , http://hal.inria.fr/hal-00804915

Other Publications

[30] A. BALLIER, E. JEANDEL. , *Structuring multi-dimensional subshifts*, 2013, http://hal.inria.fr/hal-00868899

[31] G. BONFANTE, B. GUILLAUME. , *Non-size increasing Graph Rewriting for Natural Language Processing*, 2013, Accepted for publication in MSCS (Mathematical Structures for Computer Science), http://hal.inria.fr/hal-00921038

[32] E. HAINRY, R. PÉCHOUX. , *Type-based heap and stack space analysis in Java*, January 2013, Rapport technique, http://hal.inria.fr/hal-00773141

[33] E. JEANDEL, P. VANIER. , *Characterizations of periods of multidimensional shifts*, March 2013, http://hal.inria.fr/hal-00798336

References in notes

[34] L. ADLEMAN. *An Abstract Theory of Computer Viruses*, in "Advances in Cryptology — CRYPTO'88", Lecture Notes in Computer Science, 1988, vol. 403

[35] E. ASARIN, O. MALER, A. PNUELI. *Reachability analysis of dynamical systems having piecewise-constant derivatives*, in "Theoretical Computer Science", February 1995, vol. 138, n^{o} 1, pp. 35–65

[36] F. BAADER, T. NIPKOW. , *Term rewriting and all that*, Cambridge University PressNew York, NY, USA, 1998

[37] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. *Behavior Abstraction in Malware Analysis*, in "1st International Conference on Runtime Verification", St. Julians, Malte, G. ROSU, O. SOKOLSKY (editors), Lecture Notes in Computer Science, Springer-Verlag, August 2010, vol. 6418, pp. 168-182, http://hal.inria.fr/inria-00536500/en/

[38] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. *Abstraction-based Malware Analysis Using Rewriting and Model Checking*, in "ESORICS", Pisa, Italie, S. FORESTI, M. YUNG (editors), LNCS, Springer, 2012, vol. 7459, pp. 806-823 [*DOI :* 10.1007/978-3-642-33167-1], http://hal.inria.fr/hal-00762252

[39] P. BELL, J.-C. DELVENNE, R. JUNGERS, V. D. BLONDEL. , *The Continuous Skolem-Pisot Problem: On the Complexity of Reachability for Linear Ordinary Differential Equations*, 2008, http://arxiv.org/abs/0809.2189

[40] L. BLUM, M. SHUB, S. SMALE. *On a theory of computation and complexity over the real numbers; NP completeness, recursive functions and universal machines*, in "Bulletin of the American Mathematical Society", July 1989, vol. 21, n^{o} 1, pp. 1–46

[41] G. BONFANTE, M. KACZMAREK, J.-Y. MARION. *On abstract computer virology: from a recursion-theoretic perspective*, in "Journal in Computer Virology", 2006, vol. 1, n^{o} 3-4

[42] M.G.J. VAN DEN. BRAND, A. VAN. DEURSEN, J. HEERING, H.A. DE JONG, M. DE JONGE, T. KUIPERS, P. KLINT, L. MOONEN, P. OLIVIER, J. SCHEERDER, J. VINJU, E. VISSER, J. VISSER. *The ASF+SDF Meta-Environment: a Component-Based Language Development Environment*, in "Compiler Construction (CC '01)", R. WILHELM (editor), Lecture Notes in Computer Science, Springer, 2001, vol. 2027, pp. 365–370

[43] M. S. BRANICKY. *Universal computation and other capabilities of hybrid and continuous dynamical systems*, in "Theoretical Computer Science", 6 February 1995, vol. 138, n^{o} 1, pp. 67–100

[44] M. CLAVEL, F. DURÁN, S. EKER, P. LINCOLN, N. MARTÍ-OLIET, J. MESEGUER, C. TALCOTT. *The Maude 2.0 System*, in "Proceedings of the 14th International Conference on Rewriting Techniques and Applications", R. NIEUWENHUIS (editor), Lecture Notes in Computer Science, Springer, June 2003, vol. 2706, pp. 76-87

[45] F. COHEN. , *Computer Viruses*, University of Southern California, January 1986

[46] H. COMON. *Inductionless Induction*, in "Handbook of Automated Reasoning", A. ROBINSON, A. VORONKOV (editors), Elsevier Science, 2001, vol. I, chap. 14, pp. 913-962

[47] N. DERSHOWITZ, D. PLAISTED. *Rewriting*, in "Handbook of Automated Reasoning", A. ROBINSON, A. VORONKOV (editors), Elsevier Science, 2001, vol. I, chap. 9, pp. 535-610

[48] A. EDALAT, P. SÜNDERHAUF. *A domain-theoretic approach to computability on the real line*, in "Theoretical Computer Science", 1999, vol. 210, n^{o} 1, pp. 73–98

[49] E. FILIOL. , *Computer Viruses: from Theory to Applications*, Springer-Verlag, 2005

[50] E. FILIOL. *Malware Pattern Scanning Schemes Secure Against Black-box Analysis*, in "Journal in Computer Virology", 2006, vol. 2, n^{o} 1, pp. 35-50

[51] E. FILIOL. , *Techniques virales avancées*, Springer, 2007

[52] E. FILIOL, G. JACOB, M. LE LIARD. *Evaluation methodology and theoretical model for antiviral behavioural detection strategies*, in "Journal in Computer Virology", 2007, vol. 3, n^{o} 1, pp. 23-37

[53] O. FISSORE, I. GNAEDIG, H. KIRCHNER. *Termination of rewriting with local strategies*, in "Selected papers of the 4th International Workshop on Strategies in Automated Deduction", M. P. BONACINA, B. GRAMLICH (editors), Electronic Notes in Theoretical Computer Science, Elsevier Science Publishers, 2001, vol. 58

[54] O. FISSORE, I. GNAEDIG, H. KIRCHNER. *CARIBOO : An induction based proof tool for termination with strategies*, in "Proceedings of the Fourth International Conference on Principles and Practice of Declarative Programming", Pittsburgh (USA), ACM Press, October 2002, pp. 62–73

[55] O. FISSORE, I. GNAEDIG, H. KIRCHNER. *Outermost ground termination*, in "Proceedings of the Fourth International Workshop on Rewriting Logic and Its Applications", Pisa, Italy, Electronic Notes in Theoretical Computer Science, Elsevier Science Publishers, September 2002, vol. 71

[56] O. FISSORE, I. GNAEDIG, H. KIRCHNER. *A proof of weak termination providing the right way to terminate*, in "First International Colloquium on Theoretical Aspect of Computing", Guiyang, China, Lecture Notes in Computer Science, Springer, September 2004, vol. 3407, pp. 356-371

[57] I. GNAEDIG, H. KIRCHNER. *Computing Constructor Forms with Non Terminating Rewrite Programs*, in "Proceedings of the Eighth ACM-SIGPLAN International Symposium on Principles and Practice of Declarative Programming", Venice, Italy, ACM Press, July 2006, pp. 121–132

[58] I. GNAEDIG, H. KIRCHNER. *Termination of Rewriting under Strategies*, in "ACM Transactions on Computational Logic", 2009, vol. 10, n^{o} 2, pp. 1-52, http://hal.inria.fr/inria-00182432/en/

[59] I. GNAEDIG. *Induction for Positive Almost Sure Termination*, in "Proceedings of the 9th ACM-SIGPLAN International Symposium on Principles and Practice of Declarative Programming - PPDP 2007", Wroclaw, Pologne, ACM, 2007, pp. 167-177, http://hal.inria.fr/inria-00182435/en/

[60] I. GNAEDIG, H. KIRCHNER. *Narrowing, Abstraction and Constraints for Proving Properties of Reduction Relations*, in "Rewriting, Computation and Proof - Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday", Paris, France, H. COMON, C. KIRCHNER, H. KIRCHNER (editors), Lecture Notes in Computer Science, Springer, 2007, vol. 4600, pp. 44-67, http://hal.inria.fr/inria-00182434/en/

[61] E. HAINRY. *Computing omega-limit Sets in Linear Dynamical Systems*, in "Unconventional Computation", Autriche Vienne, C. S. CALUDE, J. F. COSTA, R. FREUND, M. OSWALD, G. ROZENBERG (editors), Springer, 2008, vol. 5204, pp. 83–95, http://hal.inria.fr/inria-00250111/en/

[62] E. HAINRY. *Reachability in linear dynamical systems*, in "Computability in Europe Logic and Theory of Algorithms", Grèce Athènes, A. BECKMANN, C. DIMITRACOPOULOS, B. LÖWE (editors), Springer, 2008, vol. 5028, pp. 241-250, http://hal.inria.fr/inria-00202674/en/

[63] S. KLEENE. , *Introduction to Metamathematics*, Van Nostrand, 1952

[64] K.-I. KO. , *Complexity Theory of Real Functions*, Birkhäuser, 1991

[65] J.-Y. MARION. , *Complexité implicite des calculs, de la théorie à la pratique*, Université Nancy 2, 2000, Habilitation à diriger les recherches

[66] J.-Y. MARION, J.-Y. MOYEN. *Efficient first order functional program interpreter with time bound certifications*, in "Logic for Programming and Automated Reasoning, 7th International Conference, LPAR 2000, Reunion Island, France", M. PARIGOT, A. VORONKOV (editors), Lecture Notes in Computer Science, Springer, Nov 2000, vol. 1955, pp. 25–42

[67] J.-Y. MARION, R. PÉCHOUX. *Resource Analysis by Sup-interpretation*, in "FLOPS", Lecture Notes in Computer Science, Springer, 2006, vol. 3945, pp. 163–176

[68] C. MOORE. *Recursion Theory on the Reals and Continuous-Time Computation*, in "Theor. Comput. Sci.", 1996, vol. 162, n^{o} 1, pp. 23-44

[69] P.-E. MOREAU, C. RINGEISSEN, M. VITTEK. *A Pattern Matching Compiler for Multiple Target Languages*, in "12th Conference on Compiler Construction, Warsaw (Poland)", G. HEDIN (editor), LNCS, Springer-Verlag, May 2003, vol. 2622, pp. 61–76, http://www.loria.fr/~moreau/Papers/MoreauRV-CC2003.ps.gz

[70] B. MORIN, L. MÉ. *Intrusion detection and virology: an analysis of differences, similarities and complementariness*, in "Journal in Computer Virology", 2007, vol. 3, n^{o} 1, pp. 33-49

[71] P. ORPONEN. *A Survey of Continuous-Time Computation Theory*, in "Advances in Algorithms, Languages, and Complexity", D.-Z. DU, K.-I. KO (editors), Kluwer Academic Publishers, 1997, pp. 209-224, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.1991

[72] C. E. SHANNON. *Mathematical Theory of the Differential Analyser*, in "Journal of Mathematics and Physics MIT", 1941, vol. 20, pp. 337-354

[73] TERESE. , *Term Rewriting Systems*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2003, n^{o} 55

[74] K. THOMPSON. *Reflections on Trusting Trust*, in "Communication of the ACM", august 1984, vol. 27, pp. 761–763, Also appears in ACM Turing Award Lectures: The First Twenty Years 1965-1985

[75] M. WEBSTER, G. MALCOLM. *Detection of metamorphic computer viruses using algebraic specification*, in "Journal in Computer Virology", 2006, vol. 2, n^{o} 3, pp. 149-161

[76] M. WEBSTER, G. MALCOLM. *Detection of metamorphic and virtualization-based malware using algebraic specification*, in "Journal in Computer Virology", 2009, vol. 5, n^{o} 3, pp. 221-245

[77] K. WEIHRAUCH. , *Computable Analysis*, Springer, 2000

[78] J. VON NEUMANN. , *Theory of Self-Reproducing Automata*, University of Illinois PressUrbana, Illinois, 1966, edited and completed by A.W.Burks