IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2014

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

Project-Team CARTE

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

Creation of the Project-Team: 2009 January 01.

1. Members

Research Scientists

Isabelle Gnaedig [Inria, Researcher]
Mathieu Hoyrup [Inria, Researcher]
Simon Perdrix [CNRS, Researcher]
Pablo Arrighi [Délégation Inria, from Mar. 2014 until Aug. 2014]

Faculty Members

Emmanuel Jeandel [Team leader, Univ. Lorraine, Professor, HdR]
Guillaume Bonfante [Univ. Lorraine, Associate Professor, HdR]
Emmanuel Hainry [Univ. Lorraine, Associate Professor]
Jean-Yves Marion [Univ. Lorraine, Professor, HdR]
Romain Péchoux [Univ. Lorraine, Associate Professor]

Engineer

Fabrice Sabatier [Inria, granted by FP7 FI WARE project]

PhD Students

Aurélien Thierry [Inria, expected date of defense: March 2015]
Hugo Férée [Univ. Lorraine]
Hubert Godfroy [Inria]
Thanh Dinh Ta [Univ. Lorraine, expected date of defense: March 2015]

Post-Doctoral Fellow

Bruno Bauwens [Univ. Lorraine, until Jan. 2014]

Visiting Scientists

Robin David [CEA, from Nov. 2014]
Mizuhito Ogawa [Professor, Japan Advanced Institute of Science and Technology, from Oct. 2014 until Nov. 2014]
Luis Cristobal Rojas [Assistant Professor Universidad Andres Bello, from Jul. 2014 until Sep. 2014]

Administrative Assistants

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

Others

Mathieu Aria [DGA, M2 ENSTA, from June 2014 until Aug. 2014]
Nidhal Hamrit [Inria, M2 Telecom ParisTech, from Sep. 2014]

2. Overall Objectives

2.1. Overall Objectives

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.

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 same time, a series of papers was published which always remains a reference in computer virology: Thompson [71], Cohen [39] and Adleman [28]. The literature which explains and discusses practical issues is quite extensive [44], [46]. 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 [75]. More precisely we establish in [35] that Kleene's second recursion theorem [59] 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, a worm is an autonous self-replicating malware and so falls 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 $f : \mathbb{R} \to \mathbb{R}$ 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 [43], [74]; the Blum-Shub-Smale model (BSS), where the real numbers are treated as elementary entities [34]; the General Purpose Analog Computer (GPAC) introduced by Shannon [69]with continuous time.

3.3. Rewriting

The rewriting paradigm is now widely used for specifying, modelizing, programming and proving. It allows one 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 [36], MAUDE [38], and TOM [66].

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 [30], [42], [70].

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 [41]. 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 [72], [73]. For several years, in our team, we have also been working in this direction. We already proposed an approach using rewriting techniques to abstract program behaviors for detecting suspicious or malicious programs [31], [32].

4. Application Domains

4.1. Computer Virology

4.1.1. The theoretical track.

It is rightful to wonder why there are 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 [45] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [47], 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 [67].

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.

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., [29]), control theory (see e.g., [37]), neural networks (see e.g., [68]), and so on. We are interested in the formal decidability of properties of dynamical systems, such as reachability [58], the Skolem-Pisot problem [33], the computability of the ω-limit set [57]. 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 [69], on recursive analysis [74], on the algebraic approach [65] and on Markov computability [60]. 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 [48], [49], [50], to weak termination [51], sufficient completeness [53] and probabilistic termination [55]. 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 [54], [56]. 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 [62], [63], [64]. 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. New 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. DynamicTracer

DynamicTracer is a new tool with a public web interface which provides run trace of executable files. The trace is obtained by recording a dynamic execution in a safe environment. The trace contain instruction addresses, instruction opcodes and other optional informations.
Written in C++, 2.5k lines.
http://www.lhs.loria.fr/wp/?page_id=96

5.3. CoDisasm

Codisasm is a new disassembly program which support self-modifying code and code overlapping. Up to our knowledge, this is the first to cope both aspects of program obfuscation. The tool is based on the notion of wave developed in the group.
Written in C, 3k lines.

6. New Results

6.1. Highlights of the Year

Our team made remarkable progress into the understanding of complexity of higher-order functionals. While a robust class of computable functionals exists at any finite type built from $\mathbb{N}$ and $\rightarrow$ (the Kleene-Kreisel functionals), no satisfying complexity classes had been defined so far, except the class BFF of Basic Feasible Functionals. However that class is not a complexity class in the usual sense and does not offer the possibility to define space complexity or non-deterministic time complexity. In his PhD Hugo Férée has developed a non-trivial notion of size for higher-order functionals using game semantics and he has defined a notion of polynomial-time computable functional including BFF but behaving more satisfactorily in several ways. A paper in preparation will gather these results.

6.2. Malware Detection and Program Analysis

- **Complexity Information Flow in a Multi-threaded Imperative Language.** Program resource analysis using tiering based type system has been extended to analyze the time consumed by multi-threaded imperative programs with a shared global memory, which delineates a class of safe multi-threaded programs. In this work presented at TAMC'14 (Theory and Applications of Models of Computation) [22] Jean-Yves Marion and Romain Péchoux have demonstrated that a safe multi-threaded program runs in polynomial time if (i) it is strongly terminating w.r.t. a non-deterministic scheduling policy or (ii) it terminates w.r.t. a deterministic and quiet scheduling policy. As a consequence, we obtain a characterization of the set of polynomial time functions. As far as we know,this is the first characterization by a type system of polynomial time multi-threaded programs
- **A Categorical Treatment of Malicious Behavioral Obfuscation.** In this work presented at TAMC'14 (Theory and Applications of Models of Computation) [23] Romain Péchoux and Thanh Dinh Ta consider malicious behavioral obfuscation through the use of a new abstract model for process and kernel interactions based on monoidal categories. In this model, program observations are considered to be finite lists of system call invocations. In a first step, the authors have shown how malicious behaviors can be obfuscated by simulating the observations of benign programs. In a second step, they have shown how to generate such malicious behaviors through a technique called path replaying and they have extended the class of captured malwares by using some algorithmic transformations on morphisms graphical representation.
- **Malware Message Classification by Dynamic Analysis.** Guillaume Bonfante, Jean-Yves Marion and Thanh Dinh Ta presented to FPS in 2014 a new approach in malware retro-engineering. Usually, either communications, or code is analyzed. Here, the authors take a hybrid perspective. They showed how malware communication can be seen under a language perspective. They tested their idea on real malware and, for instance, showed that the botnet Zeus uses FTP as an underlying network support.
- **Supertagging with Constraints.** The parsing in Natural Language Processing is usually done by statistical analysis. Formal approaches are much more challenging, usually involving hard problems. Guillaume Bonfante, Bruno Guillaume, Mathieu Morey, and Guy Perrier [24] propose a new stream algorithm which discriminates tags in sentences.

6.3. Computability and Complexity

- **Genericity of semi-computable objects.** One of the main goals of computability theory is to understand and classify the algorithmic content of infinite objects, which can be expressed as the difficulty of computing them or as their ability to help solving problems. In establishing this classification one is often led to separate classes of algorithmic complexity and the construction of counter-examples is usually a hard task that requires the use of advanced technics (among which the so-called priority method with finite injury). The difficulty in such a construction is that the constructed object should satisfy two types of requirements going in opposite directions: it should lack algorithmic content but at the same time should be constructible in some way. In other words, these objects live somewhere between *generic* objects (objects with no structure) and *computable* objects (the most constructible objects). While computability theory provides formal notions of genericity, these ones are always incompatible with computability.

  We introduce a new notion of genericity which has two advantages: it is close to plain genericity, and we prove that it is compatible with semi-computability (for a property, being semi-decidable is a semi-computability notion while being decidable is a plain computability notion). The latter result has important consequences: many ad hoc existing constructions are subsumed by this result and then unified, new results can be obtained whenever the new notion of genericity captures the sought properties, and the result clarifies the role of topology in computability theory.

  This work is the sequel of the STACS 2013 paper [19] and is currently submitted [26].

- **Analytical properties of resource-bounded real functionals.** In [14] Hugo Férée, Walid Gomaa and Mathieu Hoyrup extend the results of [52] to non-deterministic complexity. More precisely, we introduce the analytical concepts of essential point and sufficient set for norms over continuous functions and use them to characterize the class of norms that are computable in non-deterministic polynomial time.
- **Call-by-value, call-by-name and the vectorial behaviour of the algebraic λ-calculus.** In this article published in LMCS (Logical Methods in Computer Science) [12], Ali Assaf, Alejandro Díaz-Caro, Simon Perdrix, Christine Tasson and Benoît Valiron examine the relationship between the algebraic lambda-calculus, a fragment of the differential lambda-calculus and the linear-algebraic lambda-calculus, a candidate lambda-calculus for quantum computation. Both calculi are algebraic: each one is equipped with an additive and a scalar-multiplicative structure, and their set of terms is closed under linear combinations. However, the two languages were built using different approaches: the former is a call-by-name language whereas the latter is call-by-value; the former considers algebraic equalities whereas the latter approaches them through rewrite rules. In this paper, they analyse how these different approaches relate to one another. To this end, four canonical languages based on each of the possible choices are proposed: call-by-name versus call-by-value, algebraic equality versus algebraic rewriting. The various languages are simulating each other. Due to subtle interaction between beta-reduction and algebraic rewriting, to make the languages consistent some additional hypotheses such as confluence or normalisation might be required.
- **Real or Natural numbers interpretations and their effect on complexity.** Guillaume Bonfante, Florian Deloup and Antoine Henrot [13] have shown how deep results in algebraic geometry may be read in a complexity perspective. They show that real numbers though they are not well founded can be used as natural numbers are for program interpretations. The argument is based on Positivstellensatz, a major result proved by Stengle.
- **Information carried by programs about the objects they compute.** In computability theory and computable analysis, finite programs can compute infinite objects. Presenting a computable object via any program for it, provides at least as much information as presenting the object itself, written on an infinite tape. What additional information do programs provide? We characterize this additional information to be any upper bound on the Kolmogorov complexity of the object, i.e., it gives an upper bound on size of a shortest program computing the object.

  This problem can be formalized using the two classical models of computation of Markov-computability [61] and Type-2 computability [74], which are the most famous and studied ways of computing with infinite objects. Many celebrated results comparing these models have been developed in the 50's (theorems by Rice, Rice-Shapiro, Kreisel-Lacombe-Schoenfiled/Ceitin, Friedberg) but a complete understanding of their precise relationship has never been obtained. Our results fill this void, identifying the exact relationship between the two models. In particular this relationship enables us to obtain several results characterizing the computational and topological structure of Markov-semidecidable properties.

  This work, made in collaboration with Cristóbal Rojas (Santiago) during his visit as an Inria "Chercheur Invité", has been accepted in STACS 2015 [20].
- **Causal Graph Dynamics.** Causal Graph Dynamics extend Cellular Automata to arbitrary, bounded-degree, time-varying graphs. The whole graph evolves in discrete time steps, and this global evolution is required to have a number of physics-like symmetries: shift-invariance (it acts everywhere the same) and causality (information has a bounded speed of propagation). Pablo Arrighi, Emmanuel Jeandel, Simon Martiel (I3S, Univ. Nice-Sophia Antipolis), and Simon Perdrix are investigating the properties of this model. In particular a work on the reversibility of causal graph dynamics has just been submitted in January 2015.
- **The Parameterized Complexity of Domination-type Problems and Application to Linear Codes.** In this article presented at TAMC'14 (Theory and Applications of Models of Computation) [17], David Cattanéo and Simon Perdrix study the parameterized complexity of domination-type problems. (σ,ρ)-domination is a general and unifying framework introduced by Telle: given

$\sigma, \rho \subseteq \mathbb{N}$, a set D of vertices of a graph G is (σ, ρ)-dominating if for any $v \in D$, $|N(v) \cap D| \in \sigma$ and for any $v \notin D$, $|N(v) \cap D| \in \rho$. The main result is that for any σ and ρ recursive sets, deciding whether there exists a (σ, ρ)-dominating set of size k, or of size at most k, are both in W[2]. This general statement is optimal in the sense that several particular instances of (σ, ρ)-domination are W[2]-complete (e.g., DOMINATING SET). This result is also extended to a class of domination-type problems which do not fall into the (σ, ρ)-domination framework, including CONNECTED DOMINATING SET and the problem of the minimal distance of a linear code over a finite field.

To prove the W[2]-membership of the domination-type problems the authors extend the Turing-way to parameterized complexity by introducing a new kind of non-deterministic Turing machine with the ability to perform 'blind' transitions, i.e., transitions which do not depend on the content of the tapes.

- **Quantum Circuits for the Unitary Permutation Problem**. In this paper [18] presented at DCM'14 (New Development in Computational models) and at the Workshop on Quantum Metrology, Interaction, and Causal Structure 2014 (invited talk), Stefano Facchni and Simon Perdrix consider the *Unitary Permutation* problem which consists, given n quantum gates $U_1, ..., U_n$ and a permutation σ of $\{1, ..., n\}$, in applying the quantum gates in the order specified by σ, i.e., in performing $U_{\sigma(n)} \circ ... \circ U_{\sigma(1)}$.

  This problem has been introduced and investigated in [40] where two models of computations are considered. The first is the (standard) model of query complexity: the complexity measure is the number of calls to any of the quantum gates U_i in a quantum circuit which solves the problem. The second model is roughly speaking a model for higher order quantum computation, where quantum gates can be treated as objects of second order. In both model the existing bounds are improved, in particular the upper and lower bounds for the standard quantum circuit model are established by pointing out connections with the *permutation as substring* problem introduced by Karp.

7. Partnerships and Cooperations

7.1. Regional Initiatives

7.1.1. Région Lorraine- Université de Lorraine

Simon Perdrix is the principal investigator of the project *measurement-based quantum computing* funded by Région Lorraine and Université de Lorraine.

7.2. National Initiatives

7.2.1. ANR

- The team is a funding partner in ANR Elica (2014-2019), "Elargir les idées logistiques pour l'analyse de complexité". The Carte team is reknown for its expertise in implicit computational complexity.
- The team is a funding partner in ANR Binsec (2013-2017), whose aim is to fill part of the gap between formal methods over executable code, 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.

7.3. European Initiatives

7.3.1. FP7 & H2020 Projects

7.3.1.1. 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: 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.

7.4. International Initiatives

7.4.1. Informal International Partners

- Submission of an Inria associate team proposal THOR (complexity Theory at Higher ORder) in collaboration with Syracuse University, Wesleyan University (Royer, Danner, Ramyaa Ramyaa) and Egypt-Japan University (Walid Gomaa).

7.5. International Research Visitors

7.5.1. Visits of International Scientists

- Cristóbal Rojas (Univ. Andres Bello, Chili) was Inria "Chercheur Invité" for 3 months from July to September 2014. The collaboration led to the paper [20] accepted at STACS 2015.
- Visit of Marco Gaboardi, full researcher at Dundee University, for one week in March 2014.

7.5.2. Short Visits to International Teams

- Romain Péchoux, two one-week visits to Dundee University in March and August 2014.
- Simon Perdrix, visit to the quantum group, Oxford University Computing Laboratory, 1 week in October 2014.
- Simon Perdrix, visit to the Tsinghua University, Beijing, 1 week in December 2014.

8. Dissemination

8.1. Promoting Scientific Activities

8.1.1. Scientific events organisation

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

- Journées SDA2 and FRAC 2014, April.
- JFIN (Journée Française d'investigation numérique), October.
- Botconf'14 (The botnet fighting conference - Second edition), December.

Simon Perdrix organised JIQ'14 (Journées Informatique quantique) in Lyon in November.

8.1.2. Scientific events selection

8.1.2.1. Conference program committee membership

- Guillaume Bonfante was in the Program Committee of the conference FPS (Foundations and Practice of Security) and the following workshops: Termgraph (Vienna), Caesar (Rennes), PPREW (New Orleans).
- Mathieu Hoyrup was in the Program Committee of Computability and Complexity in Analysis (CCA) 2014.
- Romain Péchoux was in the Program Committee of DICE 2014 (Developpements in Implicit Computational complExity).
- Simon Perdrix was in the Program Committee of STACS 2014 (Symposium on Theoretical Aspects of Computer Science) and QPL 2014 (Quantum Physics and Logic).

8.1.2.2. Reviewing activities

- Mathieu Hoyrup reviewed articles for LICS 2014, CiE 2014.
- Emmanuel Jeandel reviewed articles for STACS 2015.
- Romain Péchoux reviewed articles for DICE 2014 (Developpements in Implicit Computational complExity), ISMVL 2015 (International Symposium on Multi-Valued Logics), RTA-TLCA 014 (Rewriting Techniques and Applications - Typed Lambda-Calculi and Applications).
- Simon Perdrix reviewed articles for CONCUR'14.

8.1.3. Journal

8.1.3.1. Editorial board membership

- Emmanuel Jeandel is in the editorial board of RAIRO-ITA.

8.1.3.2. Reviewing activities

- Mathieu Hoyrup reviewed articles for *Theory of Computing Systems*, *Logical Methods in Computer Science*, *Mathematical Structures in Computer Science*.
- Emmanuel Jeandel reviewed articles for *Journal of Computer and System Sciences* and for *Ergodic Theory and Dynamical Systems*.
- Simon Perdrix reviewed articles for *Information Processing Letters*, *Quantum Information & Computation*.

8.1.4. Invited Talks

- Jean-Yves Marion gave an invited talk at the Journées Francophones de l'Investigation Numérique (JFIN).
- Emmanuel Jeandel gave an invited talk at the Journées Montoises.
- Simon Perdrix gave an invited talk at the Workshop on Quantum Metrology, Interaction, and Causal Structure 2014 (Tsinghua University, Beijing) and an invited talk at the Chinese Academy of Sciences.

8.1.5. Collective Responsibilities

Isabelle Gnaedig is:

- member of the scientific mediation committee of Inria Nancy - Grand Est,
- researcher social referee at Inria Nancy - Grand Est.

Simon Perdrix is responsible of GT IQ (groupe de travail Informatique quantique) at the CNRS GdR IM (groupe de recherche Informatique Mathématique).

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Licence :

- Guillaume Bonfante
  - Java, L3, Mines Nancy
- Emmanuel Hainry
  - Operating Systems, 30h, L1, IUT Nancy Brabois
  - Algorithmics, 40h, L1, IUT Nancy Brabois
  - Dynamic Web, 60h, L1, IUT Nancy Brabois
  - Databases, 30h, L1, IUT Nancy Brabois
  - Object Oriented Languages, 12h, L2, IUT Nancy Brabois
  - Complexity, 30h, L2, IUT Nancy Brabois
- Mathieu Hoyrup
  - Programmation en JAVA, 56h, DUT d'informatique 1ère année, IUT Charlemagne, France
- Emmanuel Jeandel
  - Algorithmics and Programming 1, 60h, L1 Maths-Info
  - Algorithmics and Programming 4, 30h, L3 Informatique
  - Modelling Using Graph Theory, 30h, L3 Informatique
  - Networking, 15h, L3 Informatique
- Romain Péchoux
  - Programmation orientée objet, 37,5h, L3, Université de Lorraine, France
  - Programmation orientée objet, 33,5h, L2, Université de Lorraine, France
  - Outils logiques pour l'informatique, 25h, L1, Université de Lorraine, France
  - Bases de données, 42h, L3, Université de Lorraine, France
  - Algorithmic complexity, 30h, L3 MIASHS parcours MIAGE, IGA Casablanca, Marocco.
- Simon Perdrix
  - Structure de données, 72h, DUT d'informatique 1ère année, IUT Charlemagne, France

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, 20h, M2, Telecom-Nancy
- Emmanuel Jeandel
  - Algorithmics and Complexity, M1 Informatique and M1 ENSEM, 30h
  - Combinatorial Optimization, M1 Informatique, 36h.

- Romain Péchoux
  - Mathematics for computer science, 30h, M1 SC
  - Advanced Java, 52,5h, M1 MIAGE
- Simon Perdrix
  - Pépites Algorithmiques — Informatique Quantique, 6h, M1/M2, Ecole des Mines de Nancy.

8.2.2. Supervision

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 : Hugo Férée, Complexité d'ordre supérieur et analyse récursive, Université de Lorraine, 10 December 2014, Jean-Yves Marion and Mathieu Hoyrup.

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

PhD: Jérôme Javelle, Quantum Cryptography: Protocols and Graphs, June 2nd 2014, 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.

8.2.3. Juries

Isabelle Gnaedig was a:

- member of the Inria hiring committee for young researchers,
- member of the Telecom-Nancy engineering school admission committee.

Emmanuel Jeandel was in the:

- Selection committee for a research assistant position in Nice (MCF 1313).
- Selection committee for a professor position in Marseille (PR 174).
- Jury of Jérôme Javelle's PhD Defense on "Crytographie Quantique: Protocoles et Graphes", defended in Université de Grenoble, June 2nd.
- Jury of Benjamin Hellouin de Menibus's PhD Defense on "Asymptotic Behaviour of Cellular Automata: Computation and Randomness", defended in Aix-Marseille Université, September 26th.
- Jury of Bastien Le Gloannec's PhD Defense on "Coloriage du plan discret par jeux de tuiles déterministes", defended in Université d'Orléans, December 12th.

8.3. 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.

9. 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, pp. 141-150, 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", Republic of Korea, Jeju Island, Springer, Dec 2010, pp. 291-303

[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. *Irreversible computable functions*, in "STACS - 31st Symposium on Theoretical Aspects of Computer Science - 2014", Lyon, France, March 2014, pp. 362-373, https://hal.inria.fr/hal-00915952

[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] H. FÉRÉE. *Higher order complexity and computable analysis*, Université de Lorraine, December 2014, https://tel.archives-ouvertes.fr/tel-01098839

Articles in International Peer-Reviewed Journals

[12] A. ASSAF, A. DÍAZ-CARO, S. PERDRIX, C. TASSON, B. VALIRON. *Call-by-value, call-by-name and the vectorial behaviour of the algebraic λ-calculus*, in "Logical Methods in Computer Science", December 2014, vol. 10:4, n^{o} 8, 40 p. [*DOI :* 10.2168/LMCS-10(4:8)2014], https://hal.inria.fr/hal-01097602

[13] G. BONFANTE, F. DELOUP, A. HENROT. *Real or Natural numbers interpretations and their effect on complexity*, in "Theoretical Computer Science", 2015, 23 p. , https://hal.archives-ouvertes.fr/hal-01093579

[14] H. FÉRÉE, W. GOMAA, M. HOYRUP. *Analytical properties of resource-bounded real functionals*, in "Journal of Complexity", October 2014, vol. 30, n° 5, 33 p. [*DOI :* 10.1016/J.JCO.2014.02.008], https://hal.inria.fr/hal-00848482

International Conferences with Proceedings

[15] 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, https://hal.inria.fr/hal-00920894

[16] G. BONFANTE, J.-Y. MARION, T. DINH TA. *Malware Message Classification by Dynamic Analysis*, in "The 7th International Symposium on Foundations and Practice of Security", Montreal, Canada, Springer, November 2014, vol. 8930, 16 p. , https://hal.inria.fr/hal-01099692

[17] D. CATTANÉO, S. PERDRIX. *The Parameterized Complexity of Domination-type Problems and Application to Linear Codes*, in "Theory and Applications of Models of Computation", Chennai, India, Lecture Notes in Computer Science, April 2014, vol. 8402, pp. 86-103 [*DOI :* 10.1007/978-3-319-06089-7_7], https://hal.archives-ouvertes.fr/hal-00944653

[18] S. FACCHINI, S. PERDRIX. *Quantum Circuits for the Unitary Permutation Problem*, in "TAMC 2015", Singapore, May 2015, https://hal.inria.fr/hal-00994182

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

[20] M. HOYRUP, C. ROJAS. *On the information carried by programs about the objects they compute*, in "STACS15", Munich, Germany, March 2015, https://hal.inria.fr/hal-01067618

[21] E. JEANDEL. *Computability of the entropy of one-tape Turing Machines*, in "STACS - Symposium on Theoretical Aspects of Computer Science", Lyon, France, E. MAYR, N. PORTIER (editors), LIPCS, March 2014, vol. 25, pp. 421-432, First version [*DOI :* 10.4230/LIPICS.STACS.2014.421], https://hal.inria.fr/hal-00785232

[22] J.-Y. MARION, R. PÉCHOUX. *Complexity Information Flow in a Multi-threaded Imperative Language*, in "TAMC 2014", Chennai, India, T. V. GOPAL, M. AGRAWAL, A. LI, S. B. COOPER (editors), Theory and Applications of Models of Computation, Springer, April 2014, pp. 124 - 140 [*DOI :* 10.1007/978-3-319-06089-7_9], https://hal.inria.fr/hal-01084043

[23] R. PÉCHOUX, T. DINH TA. *A Categorical Treatment of Malicious Behavioral Obfuscation*, in "TAMC 2014", Chennai, India, T. V. GOPAL, M. AGRAWAL, A. LI, S. B. COOPER (editors), Theory and Applications of Models of Computation., Springer, April 2014, pp. 280 - 299 [*DOI :* 10.1007/978-3-319-06089-7_20], https://hal.inria.fr/hal-01084041

Scientific Books (or Scientific Book chapters)

[24] G. BONFANTE, B. GUILLAUME, M. MOREY, G. PERRIER. *Supertagging with Constraints*, in "Constraint and Language", P. BLACHE, H. CHRISTIANSEN, V. DAHL, D. DUCHIER, J. VILLADSEN (editors), Cambridge Scholar Publishing, 2014, https://hal.inria.fr/hal-01097999

Other Publications

[25] P. GUILLON, E. JEANDEL. *Infinite Communication Complexity*, September 2014, First Version. Written from the Computer Science POV, https://hal.inria.fr/hal-01108690

[26] M. HOYRUP. *Genericity of weakly computable objects*, December 2014, https://hal.inria.fr/hal-01095864

[27] E. JEANDEL. *Some Notes about Subshifts on Groups*, January 2015, https://hal.inria.fr/hal-01110211

References in notes

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

[29] 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

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

[31] 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/

[32] 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

[33] 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

[34] 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

[35] 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

[36] 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

[37] 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

[38] 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

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

[40] T. COLNAGHI, G. M. D'ARIANO, S. FACCHINI, P. PERINOTTI. *Quantum computation with programmable connections between gates*, in "Physics Letters A", 2012, vol. 376, n^{o} 45, pp. 2940 - 2943, http://dx.doi.org/10.1016/j.physleta.2012.08.028

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

[42] 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

[43] 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

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

[45] 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

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

[47] 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

[48] 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

[49] 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

[50] 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

[51] 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

[52] 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

[53] 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

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

[55] 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/

[56] 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/

[57] 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/

[58] 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/

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

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

[61] B. A. KUSHNER. *The Constructive Mathematics of A. A. Markov*, in "The American Mathematical Monthly", 2006, vol. 113, n° 6, pp. 559-566, http://www.jstor.org/stable/27641983

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

[63] 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

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

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

[66] 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

[67] 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

[68] 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

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

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

[71] 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

[72] 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

[73] 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

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

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