



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team carte

*Theoretical adverse computations, and
safety*

Nancy - Grand Est

Theme : Programs, Verification and Proofs

Activity
R *eport*

2010

Table of contents

1. Team	1
2. Overall Objectives	1
2.1. Introduction	1
2.2. Highlights	1
3. Scientific Foundations	2
3.1. Computer Virology	2
3.2. Computation over continuous structures	2
3.3. Rewriting	2
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.	3
4.1.4. The experimentation track.	3
4.2. Computations and Dynamical Systems	4
4.2.1. Continuous computation theories	4
4.2.2. Analysis and verification of adversary systems	5
5. Software	5
5.1. Morphus/MMDEX	5
5.2. PYMS	5
5.3. TraceSurfer	5
5.4. Crème Brûlée	6
6. New Results	6
6.1. Termination and complexity with polynomial interpretation	6
6.2. Characterization of space complexity classes by KBO	6
6.3. Toward a theory of algorithms	6
6.4. Complexity of stream programs	6
6.5. A constructive version of Birkhoff's ergodic theorem	6
6.6. Computability of invariant measures	6
6.7. Setting up a large-scale malware experiment	6
6.8. Robust computations with dynamical systems	7
6.9. Computer virology: behavioral analysis	7
6.10. Computer virology: self-modifying programs	8
6.11. Type systems for complexity flow analysis	8
7. Contracts and Grants with Industry	8
8. Other Grants and Activities	8
8.1. Regional Initiatives	8
8.2. National Initiatives	8
8.3. European Initiatives	9
8.4. International Initiatives	9
9. Dissemination	9
9.1. Animation of the scientific community	9
9.2. Teaching	11
10. Bibliography	11

1. Team

Research Scientists

Isabelle Gnaedig [Junior Researcher]

Mathieu Hoyrup [Junior Researcher]

Faculty Members

Jean-Yves Marion [Team leader, Vice head of LORIA, Professor, Nancy-University, INPL, ENSMN, HdR]

Guillaume Bonfante [Associate Professor, Nancy-University, INPL, ENSMN]

Emmanuel Hainry [Associate Professor, Nancy-University, IUT Nancy Brabois, UHP]

Romain Péchoux [Associate Professor, Nancy-University, UFR MI, Université Nancy 2]

Technical Staff

Matthieu Kaczmarek [Senior engineer INRIA until August 2010]

Wadie Guizani [Engineer, INRIA until August 2010]

PhD Students

Philippe Beaucamps [BDI CNRS-DGA (since October 2007) until August 2010 – ATER UHP since September 2010, defense planned in 2011]

Joan Calvet [Ministère (since October 2009), defense planned in 2012, joint PhD with Ecole Polytechnique de Montréal]

Octave Boussaton [ACET contract, defense on February 16, 2010]

Daniel Reynaud [BDI CNRS-Région, defense on October 15, 2010 - PostDoc at U. Berkeley since then]

Stéphane Wloka [FEDER (and Master thesis till august)]

Thanh Dinh Ta [CORDI (and Master thesis till august)]

Administrative Assistants

Chantal Llorens [janvier]

Valérie Genre

Others

Hugo Férée [internship M1 from ENS Lyon]

Jan Degrieck [internship M1 from ENSMN]

Eric Théron [Master thesis]

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, and we think that their combination will be fruitful. The first one is the analysis of the behavior of a wide-scale system, using tools coming from Continuous Computation Theory. The second approach is to build defenses with tools coming rather from logic, rewriting and, more generally, from Programming Theory.

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

- Computer Virology.
- Computation over Continuous Structures

2.2. Highlights

High security Laboratory (LHS). The opening of LHS was officially made on July 1st, 2010. It is the first academic scientific platform in France to make large-scale experiments on computer security. Jean-Yves Marion is the head of LHS and all Carte's member are participating. This is the major achievement of this year.

Jean-Yves Marion got an Award “outstanding contribution” at Malware 2010.

Matthieu Kaczmarek received the first prize for his PhD in February 2010 of Région Lorraine at Metz.

3. Scientific Foundations

3.1. Computer Virology

From a historical point of view, the first official virus appeared in 1983 on Vax-PDP 11. In the very same time, a series of papers was published which always remain a reference in computer virology: Thompson [77], Cohen [45] and Adleman [35].

The literature which explains and discusses practical issues is quite extensive, see for example Ludwig’s book [66] or Szor’s one [75] and all web sites...But, we think that the best references are both books of Filiol [49] (English translation [50]) and [52]. 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 [81]. More precisely we establish in [40] that Kleene’s second recursion theorem [64] 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. Virus spread 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 autonomous 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 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, in terms of computability or complexity, of machines 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 and underlying the modern computers.

While the notion of a computable function over discrete data is captured, according to the Church-Turing thesis, by the model of Turing machines, the situation is more delicate when the data are continuous, and several non-equivalent models exist. We mention computable analysis, which relates computability to topology [48], [80]; the Blum-Shub-Smale model (BSS), where the real numbers are treated as elementary entities [39]; the General Purpose Analog Computer (GPAC) introduced by Shannon [74] where the time is continuous.

3.3. Rewriting

Rewriting has reached some maturity and the rewriting paradigm is now widely used for specifying, modeling, programming and proving. It allows for easily expressing deduction systems in a declarative way, for expressing complex relations on infinite sets of states in a finite way, provided they are countable. Programming languages and environments have been developed, which have a rewriting based semantics. Let us cite ASF+SDF [41], MAUDE [44], and TOM [71].

For basic rewriting, many techniques have been developed to prove properties of rewrite systems like confluence, completeness, consistency or various notions of termination. In a weaker proportion, proof methods have also been proposed for extensions of rewriting like 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 [37], [47], [76].

An interesting aspect of the rewriting paradigm is that it allows automatable or semi-automatable correctness proofs for systems or programs. Indeed, 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 modeled 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. Let us mention recent work using rule-based specifications for detection of computer viruses [78], [79].

4. Application Domains

4.1. Computer Virology

Nowadays, our thoughts lead us to define four different research tracks, that we are describing below.

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 [51] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [53], 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 [72].

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” is 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. [36]), control theory (see e.g. [42]), neural networks (see e.g. [73]), and so on.

We are interested in the formal decidability of properties of dynamical systems, such as reachability [63], the Skolem-Pisot problem [38], the computability of the ω -limit set [62]. Those problems are analogous to verification of safety properties.

Due to the difficulty of their analysis, the study of dynamical systems is often impossible without computer simulations. Nevertheless those simulations are often heuristic and due to round-off errors, what is observed on the screen is not guaranteed to reflect the actual behavior of the original mathematical system. Computable analysis has the advantage of getting rid of the truncation problems, integrating the management of errors to the computation. We then use this theory to investigate the possibility to compute characteristics of dynamical systems that are fundamental objects in the mathematical theory, such as attractors or invariant measures. Being asymptotic objects, they might not be always computable: for instance it has been proved in [43] that some Julia sets (see Figure 1) cannot be computed at all, i.e. there is no program that would plot such sets up to any resolution.

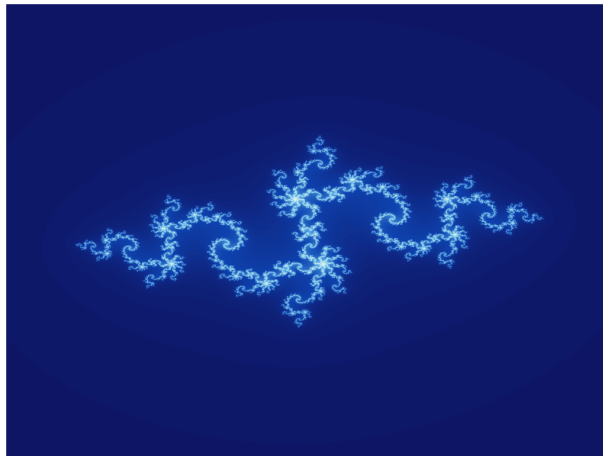


Figure 1. A Julia set: the set of points $z \in \mathbb{C}$ that do not go to ∞ when iterating $z \mapsto z^2 + c$ (here $c = -0.835 - 0.2321i$).

In [12] we prove that there exist computable systems for which the statistical long-term behavior (technically the invariant measures) is not computable.

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 [74], on recursive analysis [80], on the algebraic approach [70] and on computability in a probabilistic context [65].

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 by 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 [54], [55], [56], to weak termination [57], sufficient completeness [59] and probabilistic termination [58].

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 [60], [61]. 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 five years, see [67], [68], [69]. 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

5.1. Morphus/MMDEX

An anti-virus software based on morphological analysis, Dépôt APP du logiciel MMDEX, 2009, IDDN.FR.001.300033.000.R.P.2009.000.10000

5.2. PYMS

Online disassembler. <http>

5.3. TraceSurfer

A self-modifying code analyzer coming with an IDA add-on. <http>

5.4. Crème Brûlée

Crème Brûlée is an experimental Javascript dynamic instrumentation engine. [http](#)

6. New Results

6.1. Termination and complexity with polynomial interpretation

Guillaume Bonfante and Florian Deloup have shown that the characterization of complexity classes based on interpretations over the integers hold when using interpretations over the reals numbers in [16].

6.2. Characterization of space complexity classes by KBO

The ordering KBO has been introduced in the 70's by Knuth and Bendix. Generally speaking any multiply recursive function can be computed by a program with a KBO proof of termination. Guillaume Bonfante and Georg Moser have shown in [18] how to restrain KBO to get LINSPEACE, PSPACE and ESPACE.

6.3. Toward a theory of algorithms

Comparing characterizations of complexity classes is undecidable in general. In [15], Guillaume Bonfante shows how transformation of programs can be used to characterize algorithms and not functions.

6.4. Complexity of stream programs

Participants: Hugo Férée, Emmanuel Hainry, Mathieu Hoyrup, Romain Péchoux.

Characterization of polytime computable functions over streams using interpretations, [23]. A notion of polynomial interpretation of streams (which are second-order objects) is proposed, and polytime computable functions over streams are characterized as those functions which can be computed by a stream program admitting a polynomial interpretation.

6.5. A constructive version of Birkhoff's ergodic theorem

Participant: Mathieu Hoyrup.

A new constructive proof of Birkhoff's ergodic theorem, with as an application a strengthening of former results on random elements: in ergodic systems, random elements eventually reach *effective closed sets* of positive measure (while it was only known for a more restricted class of sets). The paper [34] is currently submitted.

6.6. Computability of invariant measures

Participant: Mathieu Hoyrup.

The statistical properties of a dynamical system are captured by its *invariant measures*. Given a computable system (a system whose evolution law can be implemented on a computer), we study the problem of computing its invariant measures. We first show that large classes of systems have computable "interesting" invariant measures. Secondly, we construct a computable system having no computable invariant measure. The paper [12] will appear in 2011.

6.7. Setting up a large-scale malware experiment

Participants: Joan Calvet, Wadie Guizani, Matthieu Kaczmarek, Jean-Yves Marion.

Botnets, that is networks of infected computers, constitute a serious security problem. A lot of effort has been invested towards understanding them better, while developing and learning how to deploy effective countermeasures against them. Their study via various analysis, modeling and experimental methods are integral parts of the development cycle of any such botnet mitigation schemes. It also constitutes a vital part of the process of understanding present threats and predicting future ones. Currently, the most popular of these techniques are “in-the-wild” botnet studies, where researchers interact directly with real-world botnets. This approach brings a lot of problems, including scientific validity, ethical and legal issues. Consequently, we developed an alternative approach employing “in the lab” experiments involving at-scale emulated botnets [21]. We have thus implemented an experiment in which we emulated a close to 3000-node, fully-featured version of the Waledac botnet, complete with a reproduced command and control (C&C) infrastructure [22]. By observing the load characteristics and yield (rate of spamming) of such a botnet, we have drawn interesting conclusions about its real-world operations and design decisions made by its creators. In particular we found that the choices made in term of cryptography by the botmasters was not so bad as we first thought during our in-the-wild study of the botnet: their primary purpose was the botnet performance, it became clear when we were setting up our own version of the botnet.

Furthermore, we conducted experiments where we launched sybil attacks against the botnet [20]. We were able to verify that such an attack is, in the case of Waledac, viable. However, we were able to determine that mounting such an attack is not so simple: high resource consumption can cause havoc and partially neutralize the attack. Finally, we were able to repeat the attack with varying parameters, in an attempt to optimize it. We thus found the optimal parameters for the attack. The merits of this experimental approach is underlined by the fact that it is very difficult to obtain these results by employing other methods.

6.8. Robust computations with dynamical systems

Participant: Emmanuel Hainry.

Olivier Bournez, Daniel Graça and Emmanuel Hainry have studied the link between undecidability and robustness in dynamical systems. Indeed, undecidability occurs very easily in dynamical systems. However there exist good decision algorithms that work for most systems that are not pathological. They argue that this decidability trait may be related to their robustness to infinitesimal noise. They have proved that in smooth dynamical systems, robustness implies decidability of the reachability problem and that the perturbed reachability problem is $P_{i_0}^1$ -complete[19].

6.9. Computer virology: behavioral analysis

Participants: Philippe Beaucamps, Isabelle Gnaedig, Jean-Yves Marion.

We have further developed our study on behavioral malware detection. We have developed an approach detecting suspicious schemes on an abstract representation of the behavior of a program. Our technique works by abstracting program traces, rewriting given subtraces into abstract symbols representing their functionality. Suspicious behaviors are then detected by comparing trace abstractions to reference malicious behaviors.

Sets of execution traces are represented by trace automata. Abstraction is then computed by rewriting trace automata, with respect to a set of predefined behavior patterns defined as a regular language described by a string rewriting system. Finally, the abstracted trace automaton is intersected with a malware database, composed of abstract signatures representing known malicious behaviors. We have shown that detection works in quadratic time with respect to the size of both trace automaton and signature database.

Traces are captured dynamically by code instrumentation, which allows us to handle packed or self-modifying malware. The expressive power of abstraction allows us to handle general suspicious behaviors rather than specific malware code and then, to detect malware mutations [14]

An implementation has been developed, validating our approach on known malwares like Allapple, Virut and Agent.

6.10. Computer virology: self-modifying programs

Participants: Isabelle Gnaedig, Matthieu Kaczmarek, Jean-Yves Marion, Daniel Reynaud, Stéphane Wloka.

We are currently studying self-modifying programs. This programming method is used in different contexts like JIT, packers but also to hide malware. We develop a representation of self-modifying programs by waves. Each wave corresponds to a maximal non-modifying part of a program. We built a tool to automatically extract waves, Tracesurfer, by using code instrumentation. We presented this tool at SSTIC (Symposium sur la Sécurité des Technologies de l'Information et de la Communication) [27]

We have proposed a new approach for studying self-modifying code in packed programs. We have given an original semantics-based technique to simplify dynamic code analysis by flattening, allowing to get a program reconstitution freed from self-modification from a packed program execution trace. It works by data flow analysis of execution traces, based on x86 instruction semantics. Flattening distinguishes traced instructions related to packing features from other ones and portrays a static view of the program, given by a time evolution of its assembly code: the flat view. Flattening has been formalized to a model based on Hybrid-GOTO, a small imperative language, having the main usual instructions. It introduces a functional view of flattening using the notions of self-modification kernel and of instruction evolution.

The approach can be very useful, because once the flattened view computed, it allows classical malware analysis tools to apply on the program, which is not possible directly on packed programs [25].

6.11. Type systems for complexity flow analysis

Participant: Jean-Yves Marion.

Jean-Yves Marion investigates intrinsic characterizations of well-known complexity classes. He develops a type system for imperative programming language to bound program runtimes. The type system is based on secure information flow typing, which has been widely studied. Intuitively, program variables are classified as either low or high. He wishes to prevent information from flowing from low variables to high variables. He proves non-interference results, which basically say that values of high variables are independent from values of low variables. Thus, he establishes a closed relation with ramified recursion, which is used in implicit computational complexity.

He also deals with declassification in order to delineate a broad class of programs. Finally, He provides a characterization of the class of polynomial time functions.

This result was presented at DICE 2010 workshop and at PICS Concerto meeting as invited talks.

7. Contracts and Grants with Industry

7.1. Contracts with Industry

We have no contract with industry. However, we have several relationships with industrial partners like Thales and Netasq and established a lot of others contacts.

8. Other Grants and Activities

8.1. Regional Initiatives

- Virus (see last year)

8.2. National Initiatives

- ANR Complice

- Project CyS of GIS 3SGS on smartphone forensics.

8.3. European Initiatives

- **EA Cristal** (France-Italy)
- **EA ComputR** (France-Portugal)

8.4. International Initiatives

- **ARC CaCO3** (France-Egypt)

9. Dissemination

9.1. Animation of the scientific community

A chapter in a Handbook on IA on “Informatique théorique: décidabilité et complexité”. Jean-Yves Marion co-coordinated this chapter [30]. The contributors are Olivier Bournez, Gilles Dowek, Rémi Gilleron, Serge Grigorieff (Co-coordinator), Simon Perdrix, and Sophie Tison. This 50 page long chapter covers some selected parts of theoretical computer science, mainly from the computability and complexity point of view. The handbook on IA is edited by Pierre Marquis, Odile Papini and Henri Prade.

Guillaume Bonfante:

- talk at the ANR-Complice (in may) meeting about course-of-value recursion and complexity classes.
- contribution to the work [17] whose aim is to compute the semantics of a sentence (in a natural language) from its syntactical analysis.

Joan Calvet was involved in the following events:

- “Modern Malware Reverse Engineering” EkoParty Conference, Buenos Aires (Argentina), September 2010 ([http](#)) The training was targeted for security researchers, exploit writers or reverse engineers looking to learn about the common techniques, tips and tools for analyzing current complex malware.
- “The Waledac botnet - its original characteristics, its weaknesses and its end in February 2010!” Association de Sécurité de l’Information du Montréal Métropolitain, Juin 2010, ([http](#)) The purpose of this presentation was to present the Waledac malware family and describe its botnet, plus some attacks that were launched against it.

Emmanuel Hainry:

- 35th international symposium on Mathematical Foundations of Computer Science - MFCS 2010, in Brno (Czech Republic)[[http](#)]
- Visit at Universidade do Algarve, Faro, Portugal in october to work with Daniel Graça on the question of decidability in dynamical systems.

Mathieu Hoyrup participated in the following events:

- organization, together with Cristóbal Rojas, of a workshop named “Computability in ergodic theorems” inside the event **Dynamics and Computation**, Luminy, February 2010.
- workshop **Logical Approaches to Barriers in Computing and Complexity**, Greifswald (Germany), February 2010. The title of the talk was “Randomness and the ergodic decomposition”.
- **5th Conference on Logic, Computability and Randomness**, South Bend (Indiana, US), May 2010.
- **Franco-Russian workshop on Algorithms, complexity and applications**, Moscow, June 2010. The title of the talk was “Randomness, computability and the ergodic decomposition”.

Mathieu Hoyrup is now in charge of the organization of the SSS seminar (temporary website: [here](#)).

Jean-Yves Marion:

- co-chairman of STACS 2010 and the main organizer. He is editor of a special issue of TOCS.
- in the program committee of DICE 2010 and FOPARA 2010.
- co-chair of Malware 2010 and the main organizer. He was one of the Panel-talker at Malware on cloud computing security.
- invited in Torino and gave a talk at PICS Concerto workshop.
- talk at INRIA centre Lille Nord Europe for the local members and local industrial partners of agence nationale de la sécurité des systèmes d'information (ANSII).
- selected to participate to the symposium franco-japonais au frontier de la science. This invitation was declined.
- participated to the habilitation thesis committee of Pierre Valarcher, U. Paris 12.
- participated to the PhD committee of Romain Demangeon (ENS Lyon) and Jonathan Rouzaud-Cornabas (U. Orleans).
- one of the visiting committee members of AERES at Caen (Greyc).
- together with Matthieu Kaczmarek, Jean-Yves Marion wrote a paper on computer viruses in Pour la Science.

Emmanuel Hainry, Mathieu Hoyrup, Jean-Yves Marion and Romain Péchoux participated to the **E-JUST Workshop on Computer Science Research**, Alexandria, October 2010. They gave the following talks:

- Emmanuel Hainry: “Robust computations with dynamical systems”
- Mathieu Hoyrup: “Decidability over infinite sequences”
- Jean-Yves Marion: “Ramified information flow”
- Romain Péchoux: “Interpretation of stream programs”

Invited researchers:

- **John Case** (University of Delaware) was invited 2 weeks in July by Jean-Yves Marion.
- Florian Deloup from the department of mathematics of the University Paul Sabatier - Toulouse III was invited for one week in february. Guillaume Bonfante and Florian Deloup initiated some new research on the topological theory of finite state automata.
- Walid Gomaa, from the University of Alexandria, during 3 weeks in July and one week in October. We work on computational complexity over the real numbers and more generally over mathematical structures from analysis. We submitted an Imhotep project (the franco-egyptian Partenariat Hubert Curien) to continue this collaboration in 2011.
- Daniel Graça, Universidade do Algarve, Faro, Portugal, came one week in May.

Diffusion in the media.

- Interview for Interstice: [http](#).
- Interview for France Culture (10/10/2010) [http](#).
- Press on the new results : Setting up ... with Joan Calvet. [atelier.net lemondeinformatique.fr](#)
- and an interview on RTV Suisse.
- Press on LHS opening: La Croix, Science et Vie Junior, le Figaro, les Échos, la Tribune, France Info, Micro Hebdo.

9.2. Teaching

Guillaume Bonfante is giving the following courses at Ecole des Mines:

- theory of computing for second year students,
- security of information systems for third year students,
- model driven architecture for second year students,
- java programming for first year students.

Isabelle Gnaedig is coordinator of the course on “Design of Safe Software” at ESIAL, 3rd year. In this context, she also gave courses and supervised practical works on “Rule-based Programming”. She is also part of the ESIAL admission committee.

Emmanuel Hainry is teaching courses on operating systems, algorithmics, object programming, functional programming and databases at IUT Nancy-Brabois.

Jean-Yves Marion is giving the following courses:

- Computer security for third year students,
- Theoretical computer science for second year students,
- Logic for computer scientists for second year students.

Jean-Yves Marion gave a course (3 hours) at Science Po, Paris on computer security.

Romain Péchoux teaches at Université Nancy 2 the following courses:

- Préparation au c2i en L1,
- Langage de programmation orienté objet en L3 MIAGE,
- Java avancé en M1 MIAGE,
- Base de données en L3 IAE,
- UML en formation continue.

10. Bibliography

Major publications by the team in recent years

- [1] G. BONFANTE, M. KACZMAREK, J.-Y. MARION. *Architecture of a morphological malware detector*, in "Journal in Computer Virology", 2009, vol. 5, n^o 3, p. 263-270 [DOI : 10.1007/s11416-008-0102-4], <http://hal.inria.fr/inria-00330022/en/>.
- [2] J. CALVET, J. M. 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/>.
- [3] 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>.
- [4] M. GABOARDI, J.-Y. MARION, S. RONCHI DELLA ROCCA. *A Logical Account of PSPACE*, in "Symposium on Principles of Programming Languages - POPL'08", États-Unis d'Amérique San Francisco, ACM, 2008, vol. 43, p. 121-131, <http://hal.archives-ouvertes.fr/hal-00342323/en/>.

- [5] P. GACS, M. HOYRUP, C. ROJAS. *Randomness on Computable Probability Spaces - A Dynamical Point of View*, in "Proceedings of the 26th Annual Symposium on the Theoretical Aspects of Computer Science STACS 2009", Freiburg Allemagne, S. ALBERS, J.-Y. MARION (editors), IBFI Schloss Dagstuhl, February 2009, p. 469-480, <http://hal.inria.fr/inria-00360519/en/>.
- [6] I. GNAEDIG, H. KIRCHNER. *Termination of Rewriting under Strategies*, in "ACM Transactions on Computational Logic", 2009, vol. 10, n^o 2, p. 1-52 [DOI : 10.1145/1462179.1462182], <http://hal.inria.fr/inria-00182432/en/>.
- [7] E. HAINRY. *Reachability in linear dynamical systems*, in "Computability in Europe Logic and Theory of Algorithms Lecture Notes in Computer Sciences", Grèce Athènes, A. BECKMANN, C. DIMITRACOPOULOS, B. LÖWE (editors), Springer, 2008, vol. 5028, p. 241-250, <http://hal.inria.fr/inria-00202674/en/>.
- [8] M. HOYRUP, C. ROJAS. *Applications of Effective Probability Theory to Martin-Löf Randomness*, in "36th International Colloquium on Automata, Languages and Programming - ICALP 2009", Grèce Rhodes, S. ALBERS, A. MARCHETTI-SPACCAMELA, Y. MATIAS, S. NIKOLETSEAS, W. THOMAS (editors), Springer Berlin / Heidelberg, July 2009, vol. 5555, p. 549-561, <http://hal.archives-ouvertes.fr/hal-00425560/en/>.
- [9] J.-Y. MARION, R. PÉCHOUX. *Sup-interpretations, a semantic method for static analysis of program resources*, in "ACM Trans. Comput. Logic", 2009, vol. 10, n^o 4, p. 1-31, <http://doi.acm.org/10.1145/1555746.1555751>.
- [10] J.-Y. MARION, R. PÉCHOUX. *Analyzing the Implicit Computational Complexity of object-oriented programs*, in "Annual Conference on Foundations of Software Technology and Theoretical Computer Science - FSTTCS 2008", Inde Bangalore, R. HARIHARAN, M. MUKUND, V. VINAY (editors), Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 License: Creative Commons-NC-ND, 2008, <http://hal.inria.fr/inria-00332550/en/>.

Publications of the year

Articles in International Peer-Reviewed Journal

- [11] S. GALATOLO, M. HOYRUP, C. ROJAS. *Effective symbolic dynamics, random points, statistical behavior, complexity and entropy*, in "Information and Computation", 2010, vol. 208, n^o 1, p. 23-41, <http://hal.inria.fr/inria-00517382>.
- [12] S. GALATOLO, M. HOYRUP, C. ROJAS. *Dynamics and abstract computability: computing invariant measures*, in "Discrete and Continuous Dynamical Systems: Series A", 01 2011, vol. 29, p. 193-212, <http://hal.inria.fr/inria-00517367>.

Invited Conferences

- [13] S. GALATOLO, M. HOYRUP, C. ROJAS. *Computing the speed of convergence of ergodic averages and pseudorandom points in computable dynamical systems*, in "Computability and Complexity in Analysis (CCA)", Chine Zhenjiang, Xizhong Zheng and Ning Zhong, 2010, vol. 24, p. 7-18, <http://hal.inria.fr/inria-00517372>.

International Peer-Reviewed Conference/Proceedings

- [14] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. *Behavior Abstraction in Malware Analysis*, in "1st International Conference on Runtime Verification", St. Julians Malte, O. S. GRIGORE ROSU (editor), Lecture Notes in Computer Science, Springer-Verlag, 08 2010, vol. 6418, p. 168-182, <http://hal.inria.fr/inria-00536500/en/>.

- [15] G. BONFANTE. *Observation of implicit complexity by non confluence*, in "International Workshop on Developments in Implicit Computational Complexity - DICE 2010", Paphos Chypre, 03 2010, <http://hal.inria.fr/inria-00535539/en/>.
- [16] G. BONFANTE, F. DELOUP. *Complexity invariance of real interpretations*, in "7th Annual Conference on Theory and Applications of Models of Computation - TAMC 2010", Prague Tchèque, République, J. F. JAN KRATOCHVI, P. KOLMAN (editors), Lecture Notes in Computer Science, Springer, 03 2010, vol. 6108, p. 139-150, <http://hal.inria.fr/inria-00535784/en/>.
- [17] G. BONFANTE, B. GUILLAUME, M. MOREY, G. PERRIER. *Réécriture de graphes de dépendances pour l'interface syntaxe-sémantique*, in "Traitement Automatique des Langues Naturelles - TALN 2010", Montréal Canada, 07 2010, <http://hal.inria.fr/inria-00526437/en/>.
- [18] G. BONFANTE, G. MOSER. *Characterising Space Complexity Classes via Knuth-Bendix Orders*, in "17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning - LPAR", Yogyakarta Indonésie, C. G. FERMÜLLER, A. VORONKOV (editors), Lecture Notes in Computer Science, Springer, 10 2010, vol. 6397, p. 142–156, <http://hal.inria.fr/inria-00535541/en/>.
- [19] O. BOURNEZ, D. GRAÇA, E. HAINRY. *Robust computations with dynamical systems*, in "35th international symposium on Mathematical Foundations of Computer Science - MFCS 2010", Tchèque, République Brno, Springer-Verlag, 2010, vol. 6281, p. 198-208, <http://hal.inria.fr/inria-00522029>.
- [20] P.-M. BUREAU, J. CALVET, J. M. FERNANDEZ, J.-Y. MARION. *Large-scale Malware Experiments: Why, How, and So What?*, in "Virus Bulletin 2010", Vancouver Canada, 09 2010, <http://hal.inria.fr/inria-00536717/en/>.
- [21] J. CALVET, J. M. FERNANDEZ, J.-Y. MARION. *Isolated virtualised clusters: testbeds for high-risk security experimentation and training*, in "3rd Workshop on Cyber Security Experimentation and Test (CSET '10)", Washington DC États-Unis, 08 2010, <http://hal.inria.fr/inria-00536712/en/>.
- [22] J. CALVET, J. M. 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/>.
- [23] 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>.
- [24] M. GABOARDI, R. PÉCHOUX. *Global and local space properties of stream programs*, in "First International Workshop on Foundational and Practical Aspects of Resource Analysis - FOPARA 2009", Eindhoven Pays-Bas, M. VAN EEKELLEN, O. SHKARAVSKA (editors), Lecture notes in computer science, Springer, 06 2010, vol. 6324, p. 51–66, <http://hal.inria.fr/inria-00539835/en/>.
- [25] I. GNAEDIG, M. KACZMAREK, D. REYNAUD, S. WLOKA. *Unconditional self-modifying code elimination with dynamic compiler optimizations*, in "Proceedings of the 5th International Conference on Malicious and Unwanted Software", Nancy France, F. C. C. OSORIO (editor), IEEE, 08 2010, vol. CFP1059F-PRT, <http://hal.inria.fr/inria-00538376/en/>.

- [26] W. GOMAA. *Polynomial Time Computation in the Context of Recursive Analysis*, in "Foundational and Practical Aspects of Resource Analysis - FOPARA 2009", Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2010, vol. 6324, p. 146-162, <http://www.springerlink.com/content/wx71303642076674/>.

National Peer-Reviewed Conference/Proceedings

- [27] W. GUIZANI, J.-Y. MARION, D. REYNAUD. *Analyse de Programmes par Traçage*, in "SSTIC", Rennes France, 06 2010, p. 125-138, <http://hal.inria.fr/inria-00549418/en/>.

Workshops without Proceedings

- [28] P.-M. BUREAU, J. CALVET. *Understanding Swizzor's Obfuscation Scheme*, in "REcon", Montréal Canada, 07 2010, <http://hal.inria.fr/inria-00536718/en/>.

- [29] J. CALVET. *Tripoux: Reverse Engineering of malware packers for dummies!*, in "DeepSec 2010", Vienna Autriche, 11 2010, <http://hal.inria.fr/inria-00536727/en/>.

Scientific Books (or Scientific Book chapters)

- [30] O. BOURNEZ, G. DOWEK, R. GILLERON, S. GRIGORIEFF, J.-Y. MARION, S. PERDRIX, S. TISON. *Décidabilité et Complexité*, in "IA Handbook", O. P. PIERRE MARQUIS, H. PRADE. (editors), Cépaduès, 01 2010, p. 1-63, <http://hal.inria.fr/inria-00549416/en/>.

Research Reports

- [31] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. *Abstraction by Term Rewriting for Malware Behavior Analysis - Extended Version*, CARTE - INRIA Lorraine - LORIA - CNRS : UMR7503 - INRIA - Université Henri Poincaré - Nancy I - Université Nancy II - Institut National Polytechnique de Lorraine, 12 2010, <http://hal.inria.fr/inria-00547884/en/>.

Scientific Popularization

- [32] L. BIENVENU, M. HOYRUP. *Une brève introduction à la théorie effective de l'aléatoire*, in "Gazette des Mathématiciens", Jan 2010, vol. 123, p. 35-47, <http://hal.inria.fr/inria-00449022>.

- [33] J.-Y. MARION, M. KACZMAREK. *Boulevard du cybercrime*, in "Dossier Pour la Science", Jan 2010, n^o 66, <http://hal.inria.fr/inria-00506082>.

Other Publications

- [34] L. BIENVENU, A. DAY, M. HOYRUP, I. MEZHROV, A. SHEN. *A constructive version of Birkhoff's ergodic theorem for Martin-Löf random points*, 2010, Submitted, <http://arxiv.org/abs/1007.5249>.

References in notes

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

- [36] 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, p. 35-65.

-
- [37] F. BAADER, T. NIPKOW. *Term rewriting and all that*, Cambridge University Press, New York, NY, USA, 1998.
- [38] 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>.
- [39] 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, p. 1–46.
- [40] 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.
- [41] 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, p. 365–370.
- [42] 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, p. 67–100.
- [43] M. BRAVERMAN, M. YAMPOLSKY. *Constructing non-computable Julia sets*, in "STOC", D. S. JOHNSON, U. FEIGE (editors), ACM, 2007, p. 709-716.
- [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, p. 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, p. 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, p. 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, p. 73–98.
- [49] E. FILIOL. *Les virus informatiques: théorie, pratique et applications*, Springer-Verlag France, 2004, Translation.
- [50] E. FILIOL. *Computer Viruses: from Theory to Applications*, Springer-Verlag, 2005.
- [51] E. FILIOL. *Malware Pattern Scanning Schemes Secure Against Black-box Analysis*, in "Journal in Computer Virology", 2006, vol. 2, n^o 1, p. 35-50.
- [52] E. FILIOL. *Techniques virales avancées*, Springer, 2007.

- [53] 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, p. 23-37.
- [54] 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.
- [55] 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, p. 62–73.
- [56] 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.
- [57] 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, p. 356-371.
- [58] I. GNAEDIG. *Induction for Positive Almost Sure Termination*, in "Proceedings of the Ninth ACM-SIGPLAN International Symposium on Principles and Practice of Declarative Programming", Wroclaw, Poland, ACM Press, July 2007, p. 167–177.
- [59] 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, p. 121–132.
- [60] I. GNAEDIG, H. KIRCHNER. *Termination of Rewriting under Strategies*, in "ACM Transactions on Computational Logic", 2009, vol. 10, n^o 2, p. 1-52, <http://hal.inria.fr/inria-00182432/en/>.
- [61] 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, p. 44-67, <http://hal.inria.fr/inria-00182434/en/>.
- [62] 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, p. 83–95, <http://hal.inria.fr/inria-00250111/en/>.
- [63] 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, p. 241-250, <http://hal.inria.fr/inria-00202674/en/>.
- [64] S. KLEENE. *Introduction to Metamathematics*, Van Nostrand, 1952.
- [65] K.-I. KO. *Complexity Theory of Real Functions*, Birkhäuser, 1991.
- [66] M. LUDWIG. *The Giant Black Book of Computer Viruses*, American Eagle Publications, 1998.

- [67] J.-Y. MARION. *Complexité implicite des calculs, de la théorie à la pratique*, Université Nancy 2, 2000, Habilitation à diriger les recherches.
- [68] 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, p. 25–42.
- [69] J.-Y. MARION, R. PÉCHOUX. *Resource Analysis by Sup-interpretation*, in "FLOPS", Lecture Notes in Computer Science, Springer, 2006, vol. 3945, p. 163–176.
- [70] C. MOORE. *Recursion Theory on the Reals and Continuous-Time Computation*, in "Theor. Comput. Sci.", 1996, vol. 162, n^o 1, p. 23-44.
- [71] 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, p. 61–76, <http://www.loria.fr/~moreau/Papers/MoreauRV-CC2003.ps.gz>.
- [72] B. MORIN, L. MÉ. *Intrusion detection and virology: an analysis of differences, similarities and complementarity*, in "Journal in Computer Virology", 2007, vol. 3, n^o 1, p. 33-49.
- [73] 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, p. 209-224, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.1991>.
- [74] C. E. SHANNON. *Mathematical Theory of the Differential Analyser*, in "Journal of Mathematics and Physics MIT", 1941, vol. 20, p. 337-354.
- [75] P. SZOR. *The Art of Computer Virus Research and Defense*, Addison-Wesley Professional, 2005.
- [76] TERESE. *Term Rewriting Systems*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2003, n^o 55.
- [77] K. THOMPSON. *Reflections on Trusting Trust*, in "Communication of the ACM", august 1984, vol. 27, p. 761–763, Also appears in ACM Turing Award Lectures: The First Twenty Years 1965-1985.
- [78] M. WEBSTER, G. MALCOLM. *Detection of metamorphic computer viruses using algebraic specification*, in "Journal in Computer Virology", 2006, vol. 2, n^o 3, p. 149-161.
- [79] 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, p. 221-245.
- [80] K. WEIHRAUCH. *Computable Analysis*, Springer, 2000.
- [81] J. VON NEUMANN. *Theory of Self-Reproducing Automata*, University of Illinois Press, Urbana, Illinois, 1966, edited and completed by A.W.Burks.