INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team CARTE

Theoretical Adverse Computations, and Safety

Nancy - Grand Est

THEME SYM

Activity Report

2008

Table of contents

1. Team

Research Scientist

Olivier Bournez [CR INRIA, Professor at Polytechnique school since September 2008, HdR]
Johanne Cohen [CR CNRS, member of PRISM since September 2008]
Isabelle Gnaedig [CR INRIA]

Faculty Member

Jean-Yves Marion [Professor, INPL, Team Leader, HdR]
Guillaume Bonfante [Assistant Professor, École des Mines de Nancy, en détachement INRIA jusqu'au 31/08]
Emmanuel Hainry [Assistant Professor, IUT Nancy Brabois, UHP depuis 1/09/2008; ATER jusqu'au 31/08/2008]

Technical Staff

Wadie Guizani [Engineer,INRIA associate engineer program (since 09/2008)]

PhD Student

Philippe Beaucamps [BDI CNRS-DGA (since October 1st 2007), defense planned in 2010]
Octave Boussaton [Bourse Région]
Matthieu Kaczmarek [BDI CNRS-Région, defense December 3rd,2008]
Marco Gaboardi [co-tutelle avec U. Turin, defense Dec. 12th, 2007]
Daniel Reynaud [BDI CNRS-Région (since October 1st), defense planned in 2010]

Post-Doctoral Fellow

Walid Gomaa [INRIA postdoctoral fellow (since september 10th)]
Mathieu Hoyrup [INPL-Région Lorraine postdoctoral fellow (since september 15th)]
Romain Péchoux [Post-doc since September 2008 at Dublin]

Administrative Assistant

Chantal Llorens

2. Overall Objectives

2.1. Introduction

Keywords: *Algorithmic*, *Algorithms*, *Complex Systems*, *Complexity*, *Computability*, *Computational Models*, *Formal Systems*, *Logic*, *Malwares*, *Networks*, *Program Properties*, *Resource Analysis*, *Termination*, *Virus*.

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 both Continuous Computation Theory and Game 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 three research actions:

- Computations by Dynamical Systems
- Robust and Distributed Algorithms, Algorithmic Game Theory
- Computer Virology.

2.2. Highlights of the year

- Paper at POPL conference [23] on light linear logic and polynomial space problems
- Survey in the Journal TOCL [14] on sup-interpreation methods to analyse program resources
- Development of morphological analysis to detect malware [10]
- New results on decidability of dynamic systems [25]
- A first method for proving termination of priority rewriting [35]
- New Results in algorithmic game theory [16]
- Polygraphs have been recognized as a good model candidate for implicit complexity and sup-interpretation methods have demonstrate their abilities to determine resopurce upper bound.

3. Scientific Foundations

3.1. Introduction

We survey the different fields, which underline the scientific basis of CARTE, enhancing the aspects around adverse computations.

3.2. Continuous Computation Theory

Today's classical computability and complexity theory deals with discrete time and space models of computation. However, discrete time models of machines working on a continuous space have been considered: see e.g. *Blum Shub and Smale* machines [43], or recursive analysis [91]. Models of machines working with continuous time and space can also be considered: see e.g. the General Purpose Analog Computer of Claude Shannon [87].

Continuous models of computation lead to particular continuous dynamical systems. More generally, continuous time dynamical systems arise as soon as one attempts to model systems that evolve over a continuous space with a continuous time. They can even emerge as natural descriptions of discrete time or space systems. Utilizing continuous time systems is a common approach in fields such as biology, physics or chemistry, when a huge population of agents (molecules, individuals, ...) is abstracted into real quantities such as proportions or thermodynamic data.

Computation theory of continuous dynamical systems allow us to understand both the hardness of questions related to continuous dynamical systems and the computational power of continuous analog models of computations.

A survey on continuous-time computation theory, with discussions of relations between both approaches, co-authored by Olivier Bournez and Manuel Campagnolo, can be found in [46].

3.3. Rewriting

Rewriting has reached some maturity and the rewriting paradigm is now widely used for specifying, modelizing, 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* [50], *Maude* [52], and *Tom* [80].

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 [40], [55], [89].

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 modelized systems to be expressed as equational theorems, often automatically provable using the rewriting mechanism itself or induction techniques based on completion [54]. Note that the rewriting and the completion mechanisms also enable transformation and simplification of formal systems or programs.

3.4. Algorithmic Game Theory

Game theory aims at discussing situations of competition between rational players [86]. After the seminal works of Emile Borel and John von Neumann, one key events was the publication in 1944 of the book [95] by John von Neumann and Oskar Morgenstern. Game theory then spent a long period in the doldrums. Much effort was devoted at that time towards the mathematics of two-person, zero-sum games.

For general games, the key concept of Nash equilibrium was proposed in the early 50s by John Nash in [83], but it was not until the early 70s that it was fully realized what a powerful tool Nash has provided in formulating this concept. This is now a central concept in economics, biology, sociology and psychology to discuss general situations of competition, as attested for example by several Nobel prizes of economics.

Algorithmic game theory differs from game theory by taking into account algorithmic and complexity aspects. Indeed, historically main developpments of classical game theory have been realized in a mathematical context, without true considerations on effectivness of constructions.

Game theory and algorithmic game theory have large domains of applications in theoretical computer science: it has been used to understand complexity of computing equilibria [77], the loss of performance due to individual behavior in distributed algorithmics [38], the design of incentive mechanisms [84], the problems related to the pricing of services in some protocols [56]...

3.5. Computer Virology

From an 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 [90], Cohen [53] and Adleman [37].

The literature which explains and discusses practical issues is quite extensive, see for example Ludwig's book [73] or Szor's one [88] and all web sites...But, we think that the best references are both books of Filiol [57] (English translation [58]) and [60]. 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 [94]. More precisely we establish in [44] that Kleene's second recursion theorem [72] is the cornerstone from which viruses and infection scenarios can be defined and classified. The bottom line of a virus is behavior is

1. A virus infect programs by modifying them
2. A virus copies itself and can mutate
3. Virus 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 are yet a difference. A malware has a payload, and virus may not have one.) For example, worms are an autonous self-replicating malwares and so fall into our definition. In fact, the current malware taxonomy (virus, worms, trojans, ...) is unclear and subject to debate.

4. Application Domains

4.1. Computations by Dynamical Systems

Keywords: *Analog Models of Computations*, *Complexity*, *Computability*.

4.1.1. Continuous time computation theories

Continuous time systems arise as soon as one attempts to model systems that evolve over a continuous space with a continuous time. They can even emerge as natural descriptions of discrete time or space systems. Utilizing continuous time systems is a common approach in fields such as biology, physics or chemistry, when a huge population of agents (molecules, individuals, ...) is abstracted into real quantities such as proportions or thermodynamic data [71], [82].

Understanding computation theories for continuous systems leads to understand hardness of verification and control of these systems. This has been used to discuss problems in fields as diverse as verification (see e.g. [39]), control theory (see e.g. [51]), VLSI design (see e.g. [78]), neural networks (see e.g. [85])

Some systems carrying out computations are intrinsically continuous. This is the case, for example, of some more or less futuristic models such as quantum models of computations, or models based on space-time curves, but also of some mechanical [87] or electronic [45] machines.

Understanding computation theories for continuous systems leads, in this context, to characterize the computational power of these machines. This has been used to discuss, for example, the power of models such as the General Purpose Analog Computer of Shannon [87], Formal Neural Network models [85].

Since continuous time systems are conducive to modeling huge populations, one might speculate that they will have a prominent role in analyzing massively parallel systems such as the Internet. This is one deep motivation of our work on computation theories for continuous systems.

4.1.2. 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.

One 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 inductive techniques, already applied with success to termination under strategies [62], [63], [64], to weak termination [65], sufficient completeness [69] and probabilistic termination [68].

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.

We especially intend to continue to study the termination property, and its impact on other properties, since it seems to be central in several research directions of the project:

- Until now it was very used for deciding properties like confluence, completeness or consistency. Its weakening or its absence, very frequent in real cases, throws us into the adversary context.
- It is a tool for the analysis of resources.

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 [74], [75], [76]. 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.

4.2. Robust and Distributed Algorithms, Algorithmic Game Theory

Keywords: *Algorithmic Game Theory*, *Distributed Systems*, *Models for Dynamics*.

One of the problems related to distributed algorithmics corresponds to the minimization of resources (time of transit, quality of services) in problems of transiting information (routing problems, group telecommunications) in telecommunication networks.

Each type of network gives rise to natural constraints on models. For example, a network is generally modeled by a graph. The material and physical constraints on each component of the network (routers, communication media, topology, etc ...) result in different models. One natural objective is then to build algorithms to solve those types of problems on various models. One can also constraint solutions to offer certain guarantees: for example the property of self-stabilization, which expresses that the system must end in a correct state whatever its initial state is; or certain guarantees of robustness: even in the presence of a small proportion of Byzantine actors, the final result will remain correct; even in the presence of rational actors with divergent interests, the final result will remain acceptable.

Algorithms of traditional distributed algorithmics were designed with the strong assumption that the interest of each actor does not differ from the interest of the group. For example, in a routing problem, classical distributed algorithms do not take into account the economic interests of the various autonomous systems, and only try to minimize criteria such as shortest distances, completely ignoring the economical consequences of decisions for involved agents.

If one wants to have more realistic models, and take into account the way the different agents behave, one gets more complex models.

However, today, one gets models which are hard to analyse. For example,

- Models of dynamism are missing: e.g., how to model a negotiation in a distributed auction mechanism for the access to a telecommunications service,
- only few methods are known to guarantee that the equilibrium reached by such systems remains in some domains that could be qualified as safe or reasonable,
- there is almost no method discussing the speed of convergence, when there is convergence,
- only a little is known about the time and space resources necessary to establish some techniques to guarantee correct behavior.

Thus, it is important to reconsider the algorithms of the theory of distributed algorithmics, under the angle of the competitive interests that involved agents can have (Adversary computation). This requires to include/understand well how to reason on these types of models.

4.3. Computer Virology

Keywords: *Virus*, *attack*, *defense*, *formal grammars*, *propagation*, *recursion theory*, *self-replication*, *worms*.

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

4.3.1. *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.

4.3.2. *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 [59] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [61], 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 interesting reader may consult [81].

4.3.3. *The virus protection track.*

The aim is to define security policies in order to prevent malaware propagation. For this, we need to (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.

4.3.4. *The experimentation track.*

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

5. Software

5.1. CARIBOO

Keywords: *Abstraction*, *Induction*, *Narrowing*, *Strategy*, *Termination*.

Participant: Isabelle Gnaedig [correspondant].

In the context of our study of rule-based program proof and validation, we develop and distribute CARIBOO (http://protheo.loria.fr/softwares/cariboo/), an environment dedicated to specific termination proofs under strategies like the innermost, the outermost or local strategies.

Written in Elan and Java, it has a reflexive aspect, since Elan is itself a rule-based language. CARIBOO was partially developed in the Toundra QSL project, and reinforced in the framework of the Modulogic ACI [66], [67].

5.2. CROCUS

Keywords: *Implicit computational complexity*, *Polynomial inequation*, *Resource analysis*.

Participants: Guillaume Bonfante [correspondant], Jean-Yves Marion, Romain Péchoux.

The CROCUS software aims at synthesizing quasi-interpretations. It takes programs as input and returns the corresponding quasi-interpretation. Doing this, it can guarantee some bounds on the memory used along computations by the input program. The currently analyzed programs are written in a subset of the CAML language, more precisely a first-order functional language subset of CAML. The synthesis procedure has been reconsidered, it is more robust and efficient.

6. New Results

6.1. Analog Computations

Keywords: *Analog Computations*, *Complexity*, *Computability*.

Participants: Olivier Bournez, Daniel Graça, Emmanuel Hainry.

There are many ways to model analog computers. Unifying those models is therefore a crucial matter as opposed to the discrete case, as there is no property stating that those models are equivalent. It is possible to distinguish two groups in these models: on one side, continuous time models; on the other side, discrete-time models working on continuous structures as a model derived from Turing machines. The first group contains in particular some sets of functions defined by Moore in [79]. The main representative of the second group are the real computable functions and a subclass of this set: the set of elementary computable functions.

There are few comparisons between classes of functions from the first group and from the second group, and in particular, there were almost no result of equality. Those different models can however be described using a common formalism: continuous time dynamical systems.

It is known that Euler's Gamma function is computable according to computable analysis, while it cannot be generated by Shannon's General Purpose Analog Computer (GPAC). This example has often been used in order to argue that the GPAC is less powerful than digital computations.

However, as we demonstrated in [48], when computability with GPACs is considered in the framework of recursive analysis, we obtain two equivalent models of analog computation. Since GPACs are equivalent to polynomial continuous-time dynamical systems, then we show that all real computable functions can be defined by such models.

As all computable functions can be computed using polynomial dynamical systems as proven in [48], the problems comparable with the halting problem for dynamical systems are undecidable. We are trying to exhibit the frontier between dynamical systems where the canonical questions are decidable and those where those questions are undecidable.

We have proven that the point-to-point reachability problem is decidable in linear dynamical systems whereas it is undecidable for quadratic dynamical systems in [25]. Also, in [24], we have proven that the set of ultimately reachable points becomes computable for those linear dynamical systems is computable. Those results bring applications for verification of some properties of safety for systems that can be expressed as linear dynamical systems.

6.2. Analysis and verification of adversary systems

Keywords: *Completeness*, *Induction*, *Narrowing*, *Priority*, *Rewriting*, *Strategy*, *Termination*, *Weak Property*.

Participant: Isabelle Gnaedig.

For sevaral years, we have been developing a method for proving properties of rewriting by explicit induction. The proof principle consists, for a given term rewriting system, in establishing on the ground term algebra that every term has the property to be proved, assuming this property for every smaller term. For that, we develop proof trees representing the possible derivations from any term using the term rewriting system. Two mechanisms are used, namely abstraction, introducing variables that represent canonical forms for the given property and schematizing rewriting of subterms until these forms, and narrowing, schematizing rewriting in different ways according to the possible ground instances of the terms studied. These two mechanisms deal with constraints used to ensure the inductive process, and to control the growth of the proof trees.

This year, we have focalized on priority rewriting, which is a model of computation often used in rule-based programming as well as in functional programming. Introducing priorities on rules in rewriting increases their expressive power and helps to limit computations. This rewriting paradigm consists in giving a priority order to the set of rules to be applied. Applying a rule to a term at a given position is then allowed only if no subterm under this position can be reduced to a form that would be reducible with a rule of higher priority. Termination of priority rewriting is important to guarantee that programs give a result. Althoug priority rewriting is often used for programming, until now there was no specific technique for proving its termination.

We have applied our inductive approach to termination of priority rewriting under the innermost strategy, which is a decidable relation. Restrict to the innermost case is reasonable since this strategy is often present in the programming contexts using priority rewriting. The priority mechanism has been isolated in our termination technique: it localizes in the narrowing relation used to develop the proof trees. A specific narrowing relation has been proposed, to modelize priority rewriting on ground terms. We have given a lifting lemma, ensuring the correctness of the modelization, and proposed how to deal with the above cited constraints in the specific case of priority rewriting. We then have applied our approach to security policies, recently studied in [92], [93]. This work is being submitted to a conference [35].

We also have reconsidered two previous works, proposing respectively proof procedures for weak termination [65] and weak sufficient completeness [69]. Our approach allows to ensure that rule-base programs give a result, even if they are not terminating in the usual sense. As it is constructive - the computing branches giving a result can be deduced from the proof trees - this technique liberates from the usual and stronger properties of termination and completeness.

We have extracted the common mechanisms of the two procedures, inherent to the inductive process and to the development of the proof trees for weak properties. As strong properties are to be verified on every rewriting branch from any term, the proof trees have to modelize all rewriting branches of any ground term. For proving weak properties instead, for every ground term, it is enough to exhibit one branch of the proof tree ensuring the property. We have explicited how to generate the relevant branches. We then have proposed a generic procedure instantiable by the own particular features of the two considered properties, which is itself an instance of the general scheme of [70], generalizing our approach to properties of reduction relations on $\mathcal{F}$-algebras. This work is being submitted to a journal [36].

6.3. Implicit Computational Complexity

Keywords: *complexity*, *light linear logic*, *quasi-interpretation*, *ramification*, *resource upper bound*, *sup-interpretation*, *tiering*.

Participants: Jean-Yves Marion, Guillaume Bonfante, Romain Péchoux.

The goal of implicit computational complexity is to give ontogenetic model of computational complexity. We follow two lines of research. The first line is more theoretical and is related to the initial ramified recursion theory due to Leivant and Marion and to light linear logic due to Girard. The second is more practical and is related to interpretation methods, quasi-interpretation and sup-interpretation, in order to provide an upper bound on some computational resources, which are necessary for a program execution. This approach seems to have some practical interests, and we develop a software Crocus that automatically infer complexity upper bounds of functional programs.

The theoretical line. In [23] and [12], Marco Gaboardi, Jean-Yves Marion and Simona Ronchi provided the first characterization of polynomial space problems by a light logic called STA_B. Then, we also characterize the set of NP-problems. From a practical point of view, such results can be seen as a type system which guarantees the complexity of a program.

Guillaume Bonfante, Reinhard Kahle, Jean-Yves Marion and Isabel Oitavem have proposed the first characterization of the classes NC^k in terms of impliit computational complexity in [22]. The characterization uses no explicit bounds in the recursion and also no tiering discipline is needed. It is based on three recursion schemes, one corresponds to time (time iteration), one to space allocation (explicit structural recursion) and one to internal computations (mutual in place recursion). This is, to our knowledge, the first exact characterization of the NC^k by function algebra over inÞnite domains in implicit complexity.

The practical line. Quasi-interpretations and sup-interpretations are a general method to analyse a program and to determine in a static way an upper bound on the amount of ressource (memory, time), which is necessary to execute a program. We published a survey paper in [14]. We have worked on sup-interpretations and quasi-interpretations with different perspectives. First, we showed that these methods are flexible enough to characterize the small parallel complexity classes NC^k, see [26]. Second, we extended the class of first order functional program that we can analyse by these methods, see [28]. This paper gives also some method to prove program termination, which is related to the previous section. Lastly, we apply these methods to programming object languages. In [27], Jean-Yves Marion and Romain Péchoux extended the notion of sup-interpretations to a fragment of Java. So we determine an upper bound on the space complexity on a java like program.

6.4. Models of dynamics in Networks

Keywords: *algorithmic game theory*, *dynamics*, *graphs*, *networks*, *telecommunications*.

Participants: Olivier Bournez, Johanne Cohen, Octave Boussaton.

We considered a model of interdomain routing proposed by a partner from SOGEA project. We proved that the model has no pure Nash equilibria, even for 4 nodes. Proof of convergence of the fictious player dynamics for the corresponding network has been established for some specific cases.

We reviewed the different models of dynamism in literature in game theory, in particular models from evolutionary game theory. We presented some ways to use them to realize distributed computations in [49]. Considered models are particular continuous time models, and hence are also covered by the survey [47].

Octave Boussaton is currently working on the theory of learning equilibria, in particular in Wardrop routing networks. The proof of the convergence of a specific learning strategy has been established for some networks. The result has been presented in [41], and should soon lead to a submission.

erover, we presented both a game theoretic and a distributed algorithmic approach for the transit price negotiation problem in the interdomain routing framework. We analyzed the behavior of providers on a specific scenario, mainly by considering the simple but not simplistic case of one source and one destination. The analysis of the centralized transit price negotiation problem shows that the only one non cooperative equilibrium is when the lowest cost provider takes all the market. The perspective of the game being repeated makes cooperation possible while maintaining higher prices. Then, we considered the system under a distributed framework. Indeed, in reality the nodes only have a local view of the game including the topology and thus the nature and the length of the possible routes. We simulated the behavior of the distributed system under a simple price adjustment strategy and analyzed whether it matches the theoretical results or not. This work is published in [42].

6.5. Computer virology

Keywords: *Malware*, *art of war*, *detection*, *learning*, *self-reproduction*.

Participants: Guillaume Bonfante, Matthieu Kaczmarek, Jean-Yves Marion, Daniel Reynaud, Philippe Beaucamps.

We work on the construction of new anti-viral methods. We have presented in [10] and in [20], [21] a method to detect malware based on the static analysis of program data flow graphs. We dub this approach *morphological detection*. We try to show that it is possible to recognize a malware by identifying its data flow graphs. We are currently trying to experiment our ideas and to validate them. For this, we use various tree automata techniques. This is a part of the detection track.

We work on some proof of concepts of attack. Daniel Reynaud [34] developped an attack in which a malware use the Graphics Processing Units. Philippe Beaucamps, Daniel Reynaud, Jean-Yves Marion and Eric Filiol [18] designed an attack on Firefox which allows to attack a voting process.

We are also fully implied in the construction of the high security lab (LHS).

And because there is no special drawer to put this result, Jérôme Besombes and Jean-Yves Marion have published in [9] an algorithm to infer categorial grammar (Lambek-Calculus) from positive (structured) examples.

7. Other Grants and Activities

7.1. Regional Actions

- CARTE is part of the “Sécurité et Sûreté des Système (SSS)” theme of the “contrat de plan État-Région”. Olivier Bournez is the head of the research operation TATA. Jean-Yves Marion is the co-head of the research operation LHS.
- Jean-Yves Marion is the head of the high security lab (laboratoire de haute sécurtité - LHS). Carte members are fully involved in this project.

7.2. National Actions

7.2.1. Agence Nationale de la Recherche (ANR) project SOGEA

The three-year ARA Sécurité, Systèmes embarqués et Intelligence ambiante SOGEA began on September 2006. It deals with models of dynamics in algorithmic game theory, and applications in networks. It involves some members of CARTE, and some members of the group Parallelism and some members of the group Complexity in Orsay, as well as some members of PRISM Laboratory in Versailles. The head of the project is Olivier Bournez.

7.2.2. Agence Nationale de la Recherche (ANR) project Virus

The three-year “Action de Recherche Amont” “Sécurité, Systèmes embarqués et Intelligence ambiante" "Virus" began on September 2006. It deals with the theory of computer viruses. It involves some members of CARTE and the group of Éric Filiol at the ESAT-Rennes, a group leader in computer virology. The head of the Project is Jean-Yves Marion.

7.2.3. Agence Nationale de la Recherche (ANR) project COMPLICE

The three-year "COMPLICE" will begin on January 2009. It deals with implicit computational complexity.

7.3. European Actions

- Some members of the project are involved in Computatibiliy in Europe Network. The head of the French Node is Olivier Bournez.

7.4. Visits and invitations of researchers

- José Fernandez from the Ecole polytechnique of Montreal was invited by Jean-Yves Marion

8. Dissemination

8.1. Activities within the scientific community

- Guillaume Bonfante: member of the engineering part of the Comipers hiring committee at LORIA.
- Olivier Bournez:
  - member of the board of directors of AFIT, the french chapter of the EATCS,
  - elected member (SGEN-CFDT) of the board of directors of INRIA,

  - member (titulaire since 2006) of the hiring committee (CS) at the University of Metz, section 27, since September 2004,
  - member (titulaire since 2007) of the hiring committee (CS) at INPL, section 27, since September 2004,
  - member (suppléant since 2004) of the hiring committee (CS) at INPL, section 27, since September 2001,
  - member of the “Comité Espace Transfert” of LORIA,
  - member of the list of experts of ANR and of FP7.
- Johanne Cohen:
  - member of the hiring committee (CS) at INPL, section 27, since September 2003,
  - member of the hiring committee (CS) at University of Nancy I, section 27.
- Jean-Yves Marion:
  - member of the steering committee of the International workshop on Logic and Computational Complexity (LCC/ICC),
  - member of the hiring committee (CS) at the University of Metz, section 27, since September 2004,
  - member of the hiring committee at INPL (Professors and Lecturers), section 27, since June 2004, and President since 2007.
  - elected to the scientific council of INPL in July 2003 and member of the board,
  - expert for DSPT 9, and of the PEDR committe.
  - member of the board of GIS 3SGS,
  - member of the “équipe de direction du Loria” and associate director of Loria
  - member of the invited professor part of the Comipers hiring committee at LORIA, since 2007.
  - member of CNU
  - members of the "jury du film de chercheur, Nancy"

8.2. Workshop and conference organisation

- Guillaume Bonfante, Jean-Yves Marion and Matthieu Kaczmarek co-organized TCV’08 at Loria in May 08.
- Jean-Yves Marion is co-chair of Stacs in February 2009

8.3. Program Committees

- Guillaume Bonfante and Jean-Yves Marion: program comittee of the workshop Theory of computer viruses (TCV’08).

8.4. Teaching

- Olivier Bournez teached “Algorithmic and Complexity” in Master 1st year of Nancy I University, and “Algorithmic Verification” in Master 2nd year of Nancy I University.
- Johanne Cohen teached “Distributed Algorithmic ” in Master 1st year of Nancy I University, and in 2nd year of ESIAL. She also teached “Algorithmique des télécommunications” in master 2nd year of Versailles University.

- 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”.
- Jean-Yves Marion is in charge of the option “Ingénierie des systèmes informatiques” at École des Mines de Nancy and teaches, a half service, in second and third year of ENSMN.
- Guillaume Bonfante is teaching (full service) at the "Ecole des Mines de Nancy".

8.5. Academic Supervision

- Johanne Cohen is co-supervising the thesis work of Octave Boussaton with Dominique Barth (University of Versailles).
- Jean-Yves Marion and Eric Filiol (ESAT) is supervising the thesis work of Philippe Beaucamps from November 2007.
- Jean-Yves Marion and Eric Filiol (ESAT) are supervising the thesis work of Daniel Reynaud from November 2007.
- Jean-Yves Marion and Guillaume Bonfante are co-supervising the thesis work of Matthieu Kaczmarek from September 2005. Defense on December 3rd 2008.

8.6. Thesis and admission committees

- Isabelle Gnaedig: ESIAL admission committee.
- Jean-Yves Marion: examiner for Oana Andrei and Radu Kopetz.

8.7. Participation to colloquia, seminars, invitations

- Guillaume Bonfante:
  - invited as regular visitor for one week by the CMA (Centro de Matemàtica e Applicações) at the Universidade Nova de Lisboa.
  - invited for a master course seminar at the Universidade Nova de Lisboa (Oct. 24th).
- Jean-Yves Marion:
  - invited at the Scientific Coffe on November 15th at Nancy.
  - invited at Turin, June 2008.
  - invited for a conferene at program proof and complexity conference (PCC) at Oslo in August 2008.

9. 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", 2008, http://hal.inria.fr/inria-00330022/en/.

[2] G. BONFANTE, R. KAHLE, J.-Y. MARION, I. OITAVEM. *Recursion Schemata for NCk*, in "CSL", 2008, p. 49-63.

[3] J. COHEN, A. DASGUPTA, S. GHOSH, S. TIXEUIL. *An Exercise in Selfish Stabilization*, in "ACM Transactions on Autonomous and Adaptive Systems (TAAS)", 2008, http://hal.inria.fr/inria-00335919/en/.

[4] M. GABOARDI, J.-Y. MARION, S. RONCHI DELLA ROCCA. *A logical account of pspace*, in "POPL", 2008, p. 121-131.

[5] 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, vol. 4600, Springer, 2007, p. 44-67, http://hal.inria.fr/inria-00182434/en/.

[6] 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), vol. 5028, Springer, 2008, p. 241-250, http://hal.inria.fr/inria-00202674/en/.

[7] J.-Y. MARION, R. PÉCHOUX. *Sup-interpretations, a semantic method for static analysis of program resources*, in "TOCL (accepted)", 2008.

[8] 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, IARCS, the Indian Association for Research in Computing Science, 2008, http://hal.inria.fr/inria-00332550/en/.

Year Publications

Articles in International Peer-Reviewed Journal

[9] J. BESOMBES, J.-Y. MARION. *Learning discrete categorial grammars from structures*, in "RAIRO - Theoretical Informatics and Applications", vol. 42, 2008, p. 165-182, http://hal.archives-ouvertes.fr/hal-00342331/en/.

[10] G. BONFANTE, M. KACZMAREK, J.-Y. MARION. *Architecture of a Morphological Malware Detector*, in "Journal in Computer Virology", 2008, http://hal.inria.fr/inria-00330022/en/.

[11] J. COHEN, A. DASGUPTA, S. GHOSH, S. TIXEUIL. *An Exercise in Selfish Stabilization*, in "ACM Transactions on Autonomous and Adaptive Systems (TAAS)", 2008, http://hal.inria.fr/inria-00335919/en/.

[12] M. GABOARDI, J.-Y. MARION, S. RONCHI DELLA ROCCA. *Soft Linear Logic and Polynomial Complexity Classes*, in "Electronic Notes in Theoretical Computer Science", vol. 205, 2008, p. 67-87, http://hal.archives-ouvertes.fr/hal-00342320/en/.

[13] I. GNAEDIG, H. KIRCHNER. *Termination of Rewriting under Strategies*, in "ACM Transactions on Computational Logic (accepted)", 2008, http://hal.inria.fr/inria-00182432/en/.

[14] J.-Y. MARION, R. PÉCHOUX. *Sup-interpretations, a semantic method for static analysis of program resources*, in "TOCL (accepted)", 2008.

Invited Conferences

[15] O. BOURNEZ, J. CHALOPIN, J. COHEN. *Playing With Population Protocols*, in "International Workshop on The Complexity of Simple Programs, Irlande Cork", 2008, http://hal.inria.fr/inria-00330344/en/.

International Peer-Reviewed Conference/Proceedings

[16] D. BARTH, O. BOURNEZ, O. BOUSSATON, J. COHEN. *Distributed Learning of Wardrop Equilibria*, in "7th International Conference on Unconventional Computation - UC 2008) Lecture Notes in Computer Science, Autriche Vienne", vol. 5204, 2008, p. 19–32, http://hal.inria.fr/inria-00308002/en/.

[17] P. BEAUCAMPS. *Extended Recursion-Based Formalization of Virus Mutation*, in "EICAR'08 Proceedings of EICAR'08, France Laval", 2008, http://hal.inria.fr/inria-00338062/en/.

[18] P. BEAUCAMPS, D. REYNAUD, J.-Y. MARION, E. FILIOL. *On the Use of Internet Voting on Compromised Computers*, in "4th International Conference on Information Warfare and Security", 2009.

[19] G. BONFANTE, Y. GUIRAUD. *Intensional properties of polygraphs*, in "4th International Workshop on Computing with Terms and Graphs - TERMGRAPH 2007 Electronic Notes in Theoretical Computer Science, Portugal Braga", vol. 203(1):65-77, 2008, http://hal.inria.fr/inria-00129391/en/.

[20] G. BONFANTE, M. KACZMAREK, J.-Y. MARION. *An implementation of morphological malware detection*, in "EICAR, France Laval", 2008, p. 49–62, http://hal.inria.fr/inria-00335590/en/.

[21] G. BONFANTE, M. KACZMAREK, J.-Y. MARION. *Morphological Detection of Malware*, in "International Conference on Malicious and Unwanted Software MALWARE 2008, Etats-Unis d'Amérique Alexendria VA", IEEE, Fernando C. Colon Osorio, 2008, http://hal.inria.fr/inria-00330021/en/.

[22] G. BONFANTE, R. KAHLE, J.-Y. MARION, I. OITAVEM. *Recursion Schemata for NCk*, in "22nd International Workshop, CSL 2008, 17th Annual Conference of the EACSL, Bertinoro, Italy, September 16-19, 2008. Proceedings, Italie Bertinoro", M. KAMINSKI, S. MARTINI (editors), vol. 5213, Springer, 2008, p. 49-63, http://hal.archives-ouvertes.fr/hal-00342366/en/.

[23] M. GABOARDI, J.-Y. MARION, S. RONCHI DELLA ROCCA. *A Logical Account of PSPACE*, in "Symposium on Principles of Programming Languages POPL, Etats-Unis d'Amérique San Francisco", 2008, p. 121-131, http://hal.archives-ouvertes.fr/hal-00342323/en/.

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

[25] 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), vol. 5028, Springer, 2008, p. 241-250, http://hal.inria.fr/inria-00202674/en/.

[26] J.-Y. MARION, R. PÉCHOUX. *A Characterization of NCk by First Order Functional Programs*, in "5th International Conference on Theory and Applications of Models of Computation - TAMC 2008 Theory and Applications of Models of Computation Lecture Notes in Computer Science, Chine Xian", Z. D. MANINDRA AGRAWAL, A. LI (editors), vol. 4978, Springer Berlin / Heidelberg, Xidian University, 2008, p. 136-147, http://hal.inria.fr/inria-00332390/en/.

[27] 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, IARCS, the Indian Association for Research in Computing Science, 2008, http://hal.inria.fr/inria-00332550/en/.

[28] J.-Y. MARION, R. PÉCHOUX. *Characterizations of Polynomial Complexity Classes with a Better Intensionality*, in "Proceedings of the 10th international ACM SIGPLAN conference on Principles and Practice of Declarative Programming - PPDP 2008, Espagne Valencia", ACM, Universidad Polytechnica, 2008, p. 79-88, http://hal.inria.fr/inria-00332389/en/.

National Peer-Reviewed Conference/Proceedings

[29] K. DESCHINKEL, J. COHEN. *Arrangement linéaire minimal des graphes d'intervalles*, in "congrès de la Société Francaise de Recherche Opérationnelle et d'Aide à la décision, France Clermont Ferrand", 2008, http://hal.inria.fr/inria-00309041/en/.

Workshops without Proceedings

[30] P. BEAUCAMPS, D. REYNAUD. *Malicious Firefox Extensions*, in "Symposium sur la sécurité des techniques d'information et de communication, France Rennes", 2008, http://hal.inria.fr/inria-00338077/en/.

[31] O. BOURNEZ, P. CHASSAING, J. COHEN, L. GERIN, X. KOEGLER. *On the Convergence of a Population Protocol When Population Goes to Infinity*, in "Physics and Computations, Worshop of Unconventional Computation - UC 2008, Autriche Vienne", 2008, http://hal.inria.fr/inria-00308018/en/.

[32] D. REYNAUD, J.-Y. MARION. *Obfuscation by Interpretation*, in "3rd International Workshop on the Theory of Computer Viruses - TCV 08, France Nancy", 2008, http://hal.inria.fr/inria-00337523/en/.

[33] D. REYNAUD. *A Survey on Virtual Machines for Malware Analysis*, in "3rd International Workshop on the Theory of Computer Viruses - TCV 08, France Nancy", 2008, http://hal.inria.fr/inria-00337520/en/.

[34] D. REYNAUD. *GPU Powered Malware*, in "Ruxcon, Australie Sydney", 2008, http://hal.inria.fr/inria-00332539/en/.

Research Reports

[35] I. GNAEDIG. *Termination of Priority Rewriting*, Rapport de recherche, 2008, http://hal.inria.fr/inria-00243131/en/.

[36] I. GNAEDIG, H. KIRCHNER. *Proving weak properties of rewriting*, Rapport de recherche, 2008, http://hal.inria.fr/inria-00338181/en/.

References in notes

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

[38] E. ANSHELEVICH, A. DASGUPTA, E. TARDOS, T. WEXLER. *Near-optimal network design with selfish agents*, in "Proceedings of the thirty-fifth annual ACM symposium on Theory of computing", ACM Press, 2003, p. 511–520.

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

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

[41] D. BARTH, O. BOURNEZ, O. BOUSSATON, J. COHEN. *Convergences et dynamiques du routage dans les réseaux*, in "Journées Pôle ResCom, Toulouse France", 2007, http://hal.inria.fr/inria-00182739/en/.

[42] D. BARTH, J. COHEN, L. ECHABBI, C. HAMLAOUI. *Transit prices negotiation: Combined repeated game and distributed algorithmic approach*, in "First EuroFGI International Conference on Network Control and Optimization - NET-COOP 2007 Network Control and Optimization First EuroFGI International Conference, NET-COOP 2007, Avignon, France, June 5-7, 2007. Proceedings Lecture Notes in Computer Science, Avignon France", T. CHAHED, B. TUFFIN (editors), Lecture Notes in Computer Science, ISBN 978-3-540-72708-8, vol. 4465, Springer Berlin / Heidelberg, 2007, p. 266-275, http://hal.inria.fr/inria-00180914/en/.

[43] 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", vol. 21, n^{o} 1, July 1989, p. 1–46.

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

[45] O. BOURNEZ. *Modèles Continus. Calculs. Algorithmique Distribuée*, HDR, Institut National Polytechnique de Lorraine, 7 Décembre 2006.

[46] O. BOURNEZ, M. CAMPAGNOLO. *A Survey on Continuous Time Computation*, in "New Computational Paradigms", B. COOPER (editor), Springer-Verlag, 2008.

[47] O. BOURNEZ, M. L. CAMPAGNOLO. *A Survey on Continuous Time Computation*, in "New Computational Paradigms", B. COOPER (editor), Springer, 2007, http://hal.inria.fr/inria-00102948/en/.

[48] O. BOURNEZ, M. L. CAMPAGNOLO, D. S. GRAÇA, E. HAINRY. *Polynomial differential equation compute all real computable functions*, in "Journal of Complexity", vol. 23, 2007, p. 317-335, http://hal.inria.fr/inria-00102947/en/.

[49] O. BOURNEZ, E. HAINRY. *On the Computational Capabilities of Several Models*, in "5th International Conference on Machines, Computations and Universality - MCU 2007 Machines, Computations and Universality 5th International Conference, MCU 2007, Orléans, France, September 10-13, 2007. Proceedings Lecture Notes in Computer Science, Orléans France", J. DURAND-LOSE, M. MARGENSTERN (editors), Lecture Notes in Computer Science, ISBN 978-3-540-74592-1, vol. 4664, Springer Berlin / Heidelberg, 2007, p. 12-23, http://hal.inria.fr/inria-00182738/en/.

[50] 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, vol. 2027, Springer, 2001, p. 365–370.

[51] M. S. BRANICKY. *Universal computation and other capabilities of hybrid and continuous dynamical systems*, in "Theoretical Computer Science", vol. 138, n° 1, 6 February 1995, p. 67–100.

[52] 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, vol. 2706, Springer, June 2003, p. 76-87.

[53] F. COHEN. *Computer Viruses*, Ph. D. Thesis, University of Southern California, January 1986.

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

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

[56] J. FEIGENBAUM, C. H. PAPADIMITRIOU, S. SHENKER. *Sharing the Cost of Multicast Transmissions*, in "Journal of Computer and System Sciences", vol. 63, n° 1, 2001, p. 21–41.

[57] E. FILIOL. *Les virus informatiques: théorie, pratique et applications*, Translation, Springer-Verlag France, 2004.

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

[59] E. FILIOL. *Malware Pattern Scanning Schemes Secure Against Black-box Analysis*, in "Journal in Computer Virology", vol. 2, n° 1, 2006, p. 35-50.

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

[61] E. FILIOL, G. JACOB, M. L. LIARD. *Evaluation methodology and theoretical model for antiviral behavioural detection strategies*, in "Journal in Computer Virology", vol. 3, n° 1, 2007, p. 23-37.

[62] 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, vol. 58, Elsevier Science Publishers, 2001.

[63] 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.

[64] 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, vol. 71, Elsevier Science Publishers, September 2002.

[65] 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, vol. 3407, Springer, September 2004, p. 356-371.

[66] O. FISSORE, I. GNAEDIG, H. KIRCHNER. *CARIBOO, a termination proof tool for rewriting-based programming languages with strategies, Version 1.0*, August 2004, Free GPL Licence, APP registration IDDN.FR.001.170013.000.R.P.2005.000.10600.

[67] O. FISSORE, I. GNAEDIG, H. KIRCHNER, L. MOUSSA. *CARIBOO, a termination proof tool for rewriting-based programming languages with strategies, Version 1.1*, December 2005, http://protheo.loria.fr/softwares/cariboo/, Free GPL Licence, APP registration IDDN.FR.001.170013.000.S.P.2005.000.10600.

[68] 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.

[69] 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.

[70] 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, vol. 4600, Springer, 2007, p. 44-67, http://hal.inria.fr/inria-00182434/en/.

[71] M. W. HIRSCH, S. SMALE, R. DEVANEY. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, Elsevier Academic Press, 2003.

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

[73] M. LUDWIG. *The Giant Black Book of Computer Viruses*, American Eagle Publications, 1998.

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

[75] 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, vol. 1955, Springer, Nov 2000, p. 25–42.

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

[77] R. MCKELVEY, A. MCLENNAN. *Computation of equilibria in finite games*, in "Handbook of Computational Economics", Elsevier, 1996, citeseer.ist.psu.edu/mckelvey96computation.html.

[78] J. W. MILLS, B. HIMEBAUGH, A. ALLRED, D. BULWINKLE, N. DECKARD, N. GOPALAKRISHNAN, J. MILLER, T. MILLER, K. NAGAI, J. NAKAMURA, B. OLOLOWEYE, R. VLAS, P. WHITENER, M. YE, C. ZHANG. *Extended Analog Computers: A Unifying Paradigm for VLSI, Plastic and Colloidal Computing Systems*, in "Workshop on Unique Chips and Systems (UCAS-1). Held in conjunction with IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS05), Austin, Texas", March 2005.

[79] C. MOORE. *Dynamical recognizers: real-time language recognition by analog computers*, in "Theoretical Computer Science", vol. 201, n^{o} 1–2, 6 July 1998, p. 99–136.

[80] 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, vol. 2622, Springer-Verlag, May 2003, p. 61–76.

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

[82] J. D. MURRAY. *Mathematical Biology. I: An Introduction*, in "Biomathematics", vol. 17, Springer Verlag, 2002.

[83] J. F. NASH. *Equilibrium points in n-person games*, in "Proc. of the National Academy of Sciences", vol. 36, 1950, p. 48-49.

[84] N. NISAN, A. RONEN. *Algorithmic mechanism design (extended abstract)*, in "Proceedings of the thirty-first annual ACM symposium on Theory of computing", ACM Press, 1999, p. 129–140.

[85] 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, citeseer.ist.psu.edu/orponen97survey.html.

[86] M. J. OSBOURNE, A. RUBINSTEIN. *A Course in Game Theory*, MIT Press, 1994.

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

[88] P. SZOR. *The Art of Computer Virus Research and Defense*, Addison-Wesley Professional, 2005.

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

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

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

[92] A. S. DE OLIVEIRA. *Réécriture et Modularité pour les Politiques de Sécurité*, Ph. D. Thesis, Univesité Henri Poincaré, Nancy, France, 2008.

[93] A. S. DE OLIVEIRA, E. K. WANG, C. KIRCHNER, H. KIRCHNER. *Weaving rewrite-based access control policies*, in "FMSE '07: Proceedings of the 2007 ACM workshop on Formal methods in security engineering, New York, NY, USA", ACM, 2007, p. 71–80.

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

[95] J. VON NEUMANN, O. MORGENSTERN. *Theory of Games and Economic Behavior*, First, Princeton University Press, Princeton, New Jersey, 1944.