INRIA

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

Table of contents

1. Team

Research Scientist

Isabelle Gnaedig [CR INRIA]
Mathieu Hoyrup [CR INRIA, from October 2009 – PostDoc INPL-Région Lorraine until October 2009]

Faculty Member

Jean-Yves Marion [Professor, Nancy-University, INPL, ENSMN, Team Leader, HdR]
Guillaume Bonfante [Assistant Professor, Nancy-University, INPL, ENSMN]
Emmanuel Hainry [Assistant Professor, Nancy-Univsersity, IUT Nancy Brabois, UHP]
Romain Péchoux [Assistant Professor, Nancy-Univsersity, UFR MI, Université Nancy 2, Nancy-Université, from September 2009]

Technical Staff

Matthieu Kaczmarek [Senior engineer INRIA from September 2009 – ATER INPL until October 2009]
Wadie Guizani [Engineer, INRIA]

PhD Student

Philippe Beaucamps [BDI CNRS-DGA (since October 1st 2007), defense planned in 2010]
Joan Calvet [Ministère (since October 1st 2009), defense planned in 2012, joint PHD with Ecole Polytechnique de Montréal]
Octave Boussaton [Bourse Région until September 2009, ACET contract since September]
Daniel Reynaud [BDI CNRS-Région, defense planned in 2010]

Post-Doctoral Fellow

Walid Gomaa [INRIA postdoctoral fellow]

Administrative Assistant

Chantal Llorens

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 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

- Winner of the Oseo prize in the category Emergence of the project of malware detector based on morphological analysis. The current implementation is validated with large scale experiments.
- Opening of the High Security lab (LHS), see http://lhs.loria.fr/
- Survey in the Journal TOCL [18] on sup-interpretation methods to analyze program resources
- With code instrumentation, thanks to PIN and TraceSurfer, we are able to deal with packed malware [31].
- We propose a fined grained stratification to characterize small complexity classes with a diagonalisation method to separate classes [17].

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 [51], or recursive analysis [108]. Models of machines working with continuous time and space can also be considered: see e.g. the General Purpose Analog Computer of Claude Shannon [102].

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 [82], [97].

Computation theory of continuous dynamical systems allows 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 [54].

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 [57], MAUDE [60], and TOM [95].

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 [47], [63], [104].

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 [62]. 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 comupter viruses [106], [107].

3.4. Algorithmic Game Theory

Game theory aims at discussing situations of competition between rational players [101]. After the seminal works of Emile Borel and John von Neumann, one key events was the publication in 1944 of the book [109] 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 [98], 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 developments of classical game theory have been realized in a mathematical context, without true considerations on effectiveness 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 [93], the loss of performance due to individual behavior in distributed algorithmics [45], the design of incentive mechanisms [99], the problems related to the pricing of services in some protocols [64]...

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 [105], Cohen [61] and Adleman [44].

The literature which explains and discusses practical issues is quite extensive, see for example Ludwig's book [86] or Szor's one [103] and all web sites...But, we think that the best references are both books of Filiol [65] (English translation [66]) and [68]. 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 [110]. More precisely we establish in [53] that Kleene's second recursion theorem [83] 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 is yet a difference. A malware has a payload, and virus may not have one.) For example, worms are an autonous self-replicating malware and so fall into our definition. In fact, the current malware taxonomy (virus, worms, trojans, ...) is unclear and subject to debate.

4. Application Domains

4.1. Computations by Dynamical Systems

4.1.1. 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. [46]), control theory (see e.g. [58]), neural networks (see e.g. [100]), and so on.

We are interested in the formal decidability of properties of dynamical systems, such as reachability [81], the Skolem-Pisot problem [50], the computability of the ω-limit set [80]. Those problems are analogous to verification of safety properties.

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 [102], on recursive analysis [108], on the algebraic approach [94] and on computability in a probabilistic context [84].

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.

4.1.2. 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.

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 [70], [71], [72], to weak termination [73], sufficient completeness [77] and probabilistic termination [76].

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 [78], [79]. 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 [88], [89], [90]. 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

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 constrain 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

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 [67] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [69], 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 [96].

4.3.3. 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.

4.3.4. 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

Participant: Isabelle Gnaedig [correspondant].

In the context of our study of rule-based program proof and validation, we develop and distribute CARIBOO (http://cariboo.loria.fr/), 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 [74], [75].

5.2. CROCUS

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.

5.3. Detection engine by morphological analysis

Participants: Guillaume Bonfante, Matthieu Kaczmarek [correspondant], Jean-Yves Marion.

We develop a new approach to detect malware that we name morphological analysis. Refer to Section 4.3 for explanations on how it works. This software is registered (APP deposit). Thanks to this malware detection engine, we won an OSEO prize in the category Emergence in 2009. Publications related to this are [15].

5.4. TraceSurfer

Participants: Wadie Guizani, Jean-Yves Marion, Daniel Reynaud [correspondant].

TraceSurfer is our prototype implementation using dynamic binary instrumentation for malware analysis. This tool, based on Pin [87], can reconstruct the code waves used in self-modifying programs and detect protection patterns based on these code waves. Publications related to TraceSurfer are [31], [37], [43]

5.5. Mr. Waffles

Participant: Daniel Reynaud [correspondant].

Mr. Waffles is a small implementation of the CTL model checking algorithm described in the classical textbook by Clarke et al. Its purpose is to allow easy experimentation with model checking for academic projects, with a focus on program analysis. For this reason, we actually implemented checking over CTL-FV formulas, a more expressive extension of CTL with backward branches and free variables originally described in [85]. The project has been released as an open source Python library at http://mrwaffles.gforge.inria.fr/.

5.6. Tartetatintools

Participant: Daniel Reynaud [correspondant].

Tartetatintools is made of four program instrumentation tools which are useful for program analysis :

- antiantidebug: detects a few anti-debugging tricks on Windows
- puppetmaster: detects a few CPU-based VMM detection tricks
- stracewin_ia32: logs system calls and their parameters for the traced program
- tracesurfer: a self-modifying code analyzer (along with an IDA add-on)

The project has been released as an open source project at http://code.google.com/p/tartetatintools/.

5.7. Cremebrulee

Participant: Daniel Reynaud [correspondant].

Cremebrulee is an experimental Javascript dynamic instrumentation engine. It takes a script, rewrites it (i.e. instruments it) and runs the instrumented version. During the rewrite, a few modifications occur that log interesting events in obfuscated scripts.

This tool is useful to analyze programs written in java script.

The project has been released as an open source project at http://code.google.com/p/cremebrulee/.

5.8. Pym's

Participant: Matthieu Kaczmarek [correspondant].

In the context of program analysis, we develop and distribute Pym's library http://code.google.com/p/pymsasid/ and Pym's online disassembler http://disasm86.appspot.com/.

The former is is python disassembling library that provides interfaces for reverse engineering and static analysis such as control flow extraction. The latter is a Software As A Service application which allows to visualize the instructions and the control flow of a program.

6. New Results

6.1. Computation over the Continuum

Participants: Walid Gomaa, Emmanuel Hainry, Mathieu Hoyrup.

While the notion of computable function over the natural numbers is universally accepted, its counterpart over continuous spaces, as the real line, is subject to discussion. The wide range of possible formalizations partly has its origin in the diversity of structures continuous spaces can be endowed with: e.g. the real line can be seen as a topological space, a measure space, a field, a vector space, a manifold, etc., depending on the particular problem one is concerned with. It happens that the topological structure of the set of real numbers is usually implicitly taken as a reference for the theory of computable functions.

On the other hand, we are interested in the analysis of dynamical systems from the computability point of view. It happens that the probabilistic framework is of much interest to understand the behavior of dynamical systems, as it enables one to distinguish physically relevant features of such systems, providing at the same time a way to understand robustness to noise. In [26], Mathieu Hoyrup, together with Peter Gács and Cristóbal Rojas, fully characterize the algorithmic effectivity of a natural class of properties arising naturally in dynamical systems.

As a result, restricting to a topological approach is somewhat limitative and we are interested in a theory of computable functions that would fit well with probabilities. We carry out such a development in [32]. Here the algorithmic theory of randomness, initiated by Martin-Löf in 1966 [92], come into play. This theory offers a way to distinguish, in a probability space, elements that are plausible w.r.t. the probability measure put onto the space, the *random* elements. This theory is already at the intersection between probability and computability. In [32] Mathieu Hoyrup and Cristóbal Rojas show that it gives a powerful and elegant way of handling computability in a probabilistic context. They present applications of this framework in [33].

Olivier Bournez, Walid Gomaa, and Emmanuel Hainry presented in [39] a framework that uses approximation to characterize both computability and complexity classes of functions from recursive analysis. This work provides an algebraical characterization of polynomial-time computable functions in the sense of Ko [84] and also extends techniques introduced in [59] for comparing discrete models with continuous models.

Walid Gomaa in [29] compares between computation over the space of continuous rational functions and the corresponding space of real functions. This investigation provides deeper insights into the role that continuity and smoothness of a real function play in the computability and/or complexity of the computation of such function.

6.2. Analysis and verification of adversary systems

Participant: Isabelle Gnaedig.

In the last few years, a significative amount of work has been done to propose correctness proof methods for rewriting-based programming. For termination and sufficient completeness, for instance, various proof techniques are now available, when the reduction relation is enriched by equations, conditions, or when it is applied with particular strategies. Nevertheless, there is still a lack of techniques, for example for certain strategies, of for weak properties i.e., properties that are not verified on every computation branch of the reduction relation. The latter properties are interesting since in practice, programs do not always verify the properties in their strong acceptance.

For several years, we have been trying to answer the above problem in developing an induction based proof approach. For the problem of strategies, specific procedures were given for proving termination under innermost, outermost and local strategies [71], [70], [72]. We then have extracted the common mechanisms of theses procedures, to propose a simpler and more general framework, parametrized by the strategy [16]. We also have proposed an instance of this mechanism for priority rewriting, for which there was no specific termination proof method until now [27].

For the problem of weak properties, our technique was applied to weak termination under the innermost strategy [73], and to C-reducibility : a weak form of sufficient completeness, we have defined as the existence of a constructor form on at least one derivation branch from every term [77]. We have continued the generalization work of our approach for the proof of weak properties. Our inductive technique consists in developing proof trees from patterns representing ground terms, by abstracting subterms, induction can be applied on, and by narrowing. Thanks to a lifting mechanism, the proof trees model the rewriting trees, the

properties to be proved are defined on. For weak properties, the choice of narrowing branches of a term u is crucial. For weak termination, it is sufficient to consider a set of branches representing at least one rewriting step for every reducible ground instance of u. For C-reducibility, the set of narrowing branches has to be covering i.e., has to represent at least one reduction step for every ground instance of u. A new definition of narrowing has been proposed to integrate these conditions. The correctness proof of the approach has also been factorized, enlightening the common and the specific characteristics of both properties [41].

Whatever the property to be proved, the above inductive technique lies on the notion of reductibility on ground terms. We have characterized how to model reducibility and irreducibility of rewriting on ground terms using equational and disequational constraints. We have shown in particular that innermost (ir)reducibility can be modeled with a particular narrowing relation and that equational and disequational constraints are issued from the most general unifiers of this narrowing relation. We then have proposed a proof of an innermost lifting lemma using this (dis)equation-based characterization [40].

6.3. Implicit Computational Complexity

Participants: Jean-Yves Marion, Guillaume Bonfante, Romain Péchoux, Walid Gomaa, Emmanuel Hainry.

The goal of implicit computational complexity is to give ontogenetic models 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.

6.3.1. ICC Core

In [14], Guillaume Bonfante and Yves Guiraud have studied the computational model of polygraphs. For that, we consider polygraphic programs, a subclass of these objects, as a formal description of first-order functional programs. We explain their semantics and prove that they form a Turing-complete computational model. Their algebraic structure is used by analysis tools, called polygraphic interpretations, for complexity analysis. In particular, we delineate a subclass of polygraphic programs that compute exactly the functions that are Turing-computable in polynomial time.

Jean-Yves Marion [17] refines predicative analysis, on which ICC foundation leans, by using a ramified Ackermann's construction of a non-primitive recursive function. We obtain a hierarchy of functions which characterizes exactly functions, which are computed in $O(n^k)$ time over register machine model of computation. For this, we introduce a strict ramification principle. Then, we show how to diagonalize in order to obtain an exponential function and to jump outside $\cup_k \mathrm{DTIME}(n^k)$. Lastly, we suggest a dependent typed lambda-calculus to represent this construction.

6.3.2. Interpretation methods

Guillaume Bonfante, together with Florian Deloup and Antoine Henrot have reconsidered the use of reals in the context of interpretation of programs in [22]. The main issue is that the ordering over the reals is not well-founded, and, consequently, bounds on the length of computations are lost. Actually, bounds on the size of terms are also lost. The contribution is to show that these bounds can be recovered when one uses interpretations defined by the functions max and polynomials. This comes from the Positivstellensatz, a deep result of algebraic geometry.

The sup-interpretation method is proposed as a new tool to control memory resources of first order functional programs with pattern matching by static analysis [18]. Basically, a sup-interpretation provides an upper bound on the size of function outputs. A criterion, which can be applied to terminating as well as non-terminating programs, is developed in order to bound polynomially the stack frame size. Sup-interpretations are proposed by Jean-Yves Marion and Romain Péchoux. Sup-interpretations may be used in various programming setting like object oriented language programming [91].

6.3.3. Recursive analysis

Olivier Bournez, Walid Gomaa and Emmanuel Hainry investigated the notion of implicit complexity in the framework of recursive analysis. In [35], was presented a characterization of polynomial-time computable functions as well as a framework to extend classical complexity result to the real field. This characterization is the first implicit characterization of this class of functions and as such opens the field of implicit complexity in recursive analysis.

6.4. Models of dynamics in Networks

Participant: Octave Boussaton.

We considered a model of interdomain routing proposed by a partner from SOGEA project that is based on the well known BGP protocol. 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 [56]. Considered models are particular continuous time models, and hence are also covered by the survey [55]. Octave Boussaton, who has now completed his PhD, 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 [48].

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. 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 [49].

Moreover, we presented both a game theoretic and an algorithmic approach for solving the routing problem of choosing the best path in a path based protocol such as BGP. We proposed a distributed learning algorithm which is able to learn Nash equilibria in a Wardrop network. This work was published in Parallel Processing Letters [12] and a newer result on the time of convergence was published in [19]. The complexity of the method depends on the total number of paths, which can become unsustainable if the network is too large. We subsequently developed another approach that is able to narrow down the complexity of the method which is now based on the number of nodes in the graph that represents the network. This work has not been published yet, it appears in Octave Boussaton's PhD and will soon lead to a submission.

6.5. Computer virology

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

The morphological analysis is a new methods of malware detection that we propose . It is based on signature recognition of abstraction of the control flow graphs of binaries. We provide fast rooted and directed acyclic graph pattern matching algorithm based on tree automata. Compare to other (industrial) approaches, the morphological analysis based detection engines have at least two advantage. First, there are quite robust wrt malware code mutation. Second, signatures may be automatically extracted. There is a running implementation that we currently test on thousand of samples coming from honeypots and the telescope which is operated by Madynes EPI in the context of LHS. This detector may run in two modes: (i) it analyses statically binaries and (ii) it analyses dynamically binaries using an instrumentation method based on PIN and related to our second main software developement TraceSurfer.

Most of the malware are nowadays packed in order to protect themselves against analysis performed by computers or by humans. In order to cope with packing techniques, we begin studies on self-modifying programs from both a theoretical and a practical perspective. In particular, packers are a particular case of self-modifying programs. We build a tool, TraceSurfer [31], [37], [43] , based on instruction-level trace analysis and a theoretical framework. In order to model self-modifying programs, we introduce the notion of pseudo-programs, for which the program text is not fixed wrt semantics. We then develop a type system which collects information at runtime (like tainting), but which also has the ability to predict information-flow properties (like traditional type systems). This leads us to explain a self-modifying program execution as a sequence of code waves. Next, we study non-interference like properties. Then, we use this typing information to define behavior patterns, which give a high level description of decrypted or scrambled code for example. With these behavior patterns we are able to classify binaries, to detect suspect runs, and to design security policies. TraceSurfer has been tested on thousand of binaries on large scale experiments and using the cluster of LHS.

On a more theoretical aspect and related to [52], Guillaume Bonfante, Jean-Yves Marion and Daniel Reynaud have proposed a new formalization of the notion of self-rewriting. To hide themselves from antivirus software, malware heavily use self-modification. In [24], we provide an operational semantics for an abstract programming language. We prove that both compilations, from non self-modifying programs to self-modifying programs, and conversely from self-modifying programs to self-modifying programs can be performed. These compilation procedures are based on two theoretical constructions: the Rogers isomorphism and the Futamura projection.

We work on behavioral analysis in order to detect malware. The idea is to detect a behavior like a keylogger. Again our approach is to have a sound theory in order to try to give solutions [20]. Lastly, we also propose an attack on electronic vote based on web browsers [21], [34].

In 2009, we pursue the construction of the high security lab (LHS) in order to make experiments about computer security on a safe platform The EPI Madynes is working with us on this project. There are currently two operational modules : A telescope and a "baby" cluster. There will be two equipped and secure rooms inside Loria building devoted to LHS.

7. Other Grants and Activities

7.1. Regional Actions

- CARTE is part of the “Sécurité et Sûreté des Systèmes (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 COMPLICE

The three-year “COMPLICE” began on January 2009. It deals with implicit computational complexity.

7.2.2. Project GIS 3SGS

We participate to a 18 month research project on CyS cybercriminalities and smartphones with Technology University of Troyes (UTT) and IRCGN (Institut de Recherche Criminelle de la Gendarmerie Nationale à Rosny-sous-Bois).

7.2.3. University and Région Lorraine project

We get funded on a project related to computer virology by INPL and Région Lorraine.

7.3. European Actions

- Jean-Yves Marion is member of the steering committee of the International workshop on Logic and Computational Complexity (LCC/ICC),
- Équipe Associée ComputR. The Équipe Associée ComputR began in january 2009. It involves members of the Carte team, members of the Laboratoire d'Informatique de l'École Polytechnique and members of the Instituto de Telecomunicaçoes, Instituto Superior Técnico from Lisbon. It deals with computation in a continuous context. The head of this project is Emmanuel Hainry.

  http://carte.loria.fr/index.php?option=com_content&view=article&id=60&Itemid=74

7.4. Visits and invitations of researchers

- José Fernandez from the Ecole polytechnique of Montreal was invited by Jean-Yves Marion.
- Marco Gaboardi from Torino University

8. Dissemination

8.1. Activities within the scientific community

- Guillaume Bonfante: member of the engineering part of the Comipers hiring committee at LORIA.
- Jean-Yves Marion:
  - member of the "équipe de direction" and associate director of Loria
  - member of CNU, section 27
  - Expert for AERES (LIP on december 2009)
  - elected to the scientific council of INPL in July 2003 and member of the board,
  - Expert PES (Prime d'excellence Scientifique) section 27
  - member of the board of GIS 3SGS,
  - member of the hiring committee at ESIAL - UHP

8.2. Workshop and conference organisation

- Guillaume Bonfante organised PCC at Loria.

8.3. Program Committees

- Jean-Yves Marion is co-chair of STACS in February 2009
- Jean-Yves Marion is member of the PC of Malware, EICAR and FOPARA

8.4. Teaching

- 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".
- Guillaume Bonfante is teaching (full service) at the "Ecole des Mines de Nancy".
- Emmanuel Hainry is teaching (full service) at the Institut Universitaire de Technologie Nancy Brabois (Nancy Université, Université Henri Poincaré).
- Romain Péchoux is teaching (full service) at Université Nancy 2.

8.5. Academic Supervision

- Jean-Yves Marion is supervising the thesis work of Philippe Beaucamps from November 2007.
- Jean-Yves Marion is supervising the thesis work of Daniel Reynaud from November 2007.
- Jean-Yves Marion and José Fernandez (Ecole Polytechnique of Montréal) are supervising the thesis work of Joan Calvet from September 2009.
- Emmanuel Hainry has been supervising two postdoctoral fellows, Walid Gomaa and Mathieu Hoyrup.

8.6. Thesis and admission committees

- Isabelle Gnaedig: ESIAL admission committee.
- Jean-Yves Marion is member of the jury of habilitation of Véronique Cortier, Ammar Oulamara and Radu State.

8.7. Participation to colloquia, seminars, invitations

Here is the list of talks given by members of the team during the year 2009.

- Walid Gomaa:
  - *Algebraic Characterization of Computable and Complexity-Theoretic Analysis*. "New Worlds of Computation" workshop, Orléans, January 12, 2009.
  - *A Survey of Recursive Analysis and Moore's Notion of Real Computation*. "Physics and Computation" satellite workshop of UC09, Ponta Delgada, September 10, 2009.
- Emmanuel Hainry:
  - *Computing over the reals, computing with the reals*. SIESTE (Student's seminar of the École Normale Supérieure de Lyon) on December 2, 2008.
  - *Decidability in continuous time dynamical systems*. "New Worlds of Computation" workshop, Orléans, January 12, 2009.
  - *Implicit complexity in recursive analysis*. 3rd meeting of the Complice ANR Project, Nancy, October 23, 2009.
- Mathieu Hoyrup:
  - *Effective probability theory*. Invited talk in the workshop "New Interactions between Analysis, Topology and Computability", Birmingham, January 9, 2009. Invited talk.
  - *Approches algorithmiques des probabilités*. "Séminaire de Probabilités de l'Institut Élie Cartan", Nancy, April 30, 2009.
  - *Layerwise computability*. Invited talk in the 4th conference on "Logic, Computablity and Randomness", CIRM Marseille, June 30, 2009.
  - *Dynamical systems: unpredicdability vs uncomputability*. Invited talk in the workshop "Physics and Computation", satellite of UC09, Ponta Delgada, September 10, 2009.
- Jean-Yves Marion
  - *NICS*. Invited talk at Torino university with a one week stay.

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", vol. 5, n^{o} 3, 2009, p. 263-270, http://hal.inria.fr/inria-00330022/en/.

[2] 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/.

[3] 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", vol. 43, ACM, 2008, p. 121-131, http://hal.archives-ouvertes.fr/hal-00342323/en/IT.

[4] 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/US.

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

[6] 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/.

[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), vol. 5028, Springer, 2008, 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), vol. 5555, Springer Berlin / Heidelberg, July 2009, 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", vol. 10, n^{o} 4, 2009, 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", V. V. R. HARIHARAN (editor), 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

[11] P. BAILLOT, J.-Y. MARION, S. RONCHI DELLA ROCCA. *Guest editorial: Special issue on implicit computational complexity*, in "ACM Trans. Comput. Log.", vol. 10, n^{o} 4, 2009, http://doi.acm.org/10.1145/1555746.1555747.

[12] D. BARTH, O. BOURNEZ, O. BOUSSATON, J. COHEN. *Distributed Learning of Equilibria in a Routing Game*, in "Parallel Processing Letters", vol. 19, 2009, p. 189–204, http://dx.doi.org/10.1142/S012962640900016X.

[13] P. BEAUCAMPS. *Extended recursion-based formalization of virus mutation*, in "Journal in Computer Virology", vol. 5, n^{o} 3, 2009, p. 209-219, http://dx.doi.org/10.1007/s11416-008-0090-4.

[14] G. BONFANTE, Y. GUIRAUD. *Polygraphic programs and polynomial-time functions*, in "Logical Methods in Computer Science (LMCS)", vol. 5, n^{o} 2:14, 2009, p. 1-37, http://hal.inria.fr/inria-00122932/en/.

[15] G. BONFANTE, M. KACZMAREK, J.-Y. MARION. *Architecture of a morphological malware detector*, in "Journal in Computer Virology", vol. 5, n^{o} 3, 2009, p. 263-270, http://hal.inria.fr/inria-00330022/en/.

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

[17] J.-Y. MARION. *On tiered small jump operators*, in "Logical Methods in Computer Science", vol. 5, n^{o} 1, 2009, http://arxiv.org/abs/0903.2410.

[18] J.-Y. MARION, R. PÉCHOUX. *Sup-interpretations, a semantic method for static analysis of program resources*, in "ACM Trans. Comput. Logic", vol. 10, n^{o} 4, 2009, p. 1–31, http://doi.acm.org/10.1145/1555746.1555751.

Invited Conferences

[19] D. BARTH, O. BOURNEZ, O. BOUSSATON, J. COHEN. *A dynamic approach to load balancing*, in "Gamecomm 2009", 2009.

International Peer-Reviewed Conference/Proceedings

[20] P. BEAUCAMPS, J.-Y. MARION. *On Behavioral Detection*, in "Proceedings of EICAR'09", 2009, http://www.loria.fr/~beaucphi/articles/beaucamps-marion-behavioral_detection-eicar09.pdf.

[21] P. BEAUCAMPS, D. REYNAUD, J.-Y. MARION, E. FILIOL. *On the Use of Internet Voting on Compromised Computers*, in "Proceedings of ICIW'09", 2009, http://www.loria.fr/~beaucphi/articles/beaucamps-reynaud-marion-filiol-internet_voting-iciw09.pdf.

[22] G. BONFANTE, D. FLORIAN, A. HENROT. *Polynomials over the reals are safe for program interpretations*, in "Foundational and Practical Aspects of Resource Analysis (FOPARA '09)", 2009.

[23] G. BONFANTE, B. GUILLAUME, M. MOREY. *Dependency Constraints for Lexical Disambiguation*, in "Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09), Paris, France", Association for Computational Linguistics, October 2009, p. 242–253, http://www.aclweb.org/anthology/W09-3840.

[24] G. BONFANTE, J.-Y. MARION, D. REYNAUD. *A computability perspective on self-modifying programs*, in "7th IEEE International Conference on Software Engineering and Formal Methods, Hanoi Viet Nam", November 2009, http://hal.inria.fr/inria-00431667/en/.

[25] M. GABOARDI, R. PÉCHOUX. *Upper Bounds on Stream I/O Using Semantic Interpretations*, in "23rd international Workshop on Computer Science Logic, CSL 2009, 18th Annual Conference of the EACSL Computer Science Logic, Coimbra Portugal", E. GRÄDEL, R. KAHLE (editors), Lecture Notes in Computer Science, vol. 5771, Springer, September 2009, p. 271-286, http://hal.inria.fr/inria-00431469/en/IT.

[26] 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/US.

[27] I. GNAEDIG. *Termination of Priority Rewriting*, in "Third International Conference on Language and Automata Theory and Applications - LATA 2009, Espagne Tarragona", A. H. DEDIU, A. M. IONESCU, C. MARTIN-VIDE (editors), vol. 5457, Springer, 2009, p. 386-397, http://hal.inria.fr/inria-00428679/en/.

[28] W. GOMAA. *A Survey of Recursive Analysis and Moore's Notion of Real Computation*, in "Proceedings of Physics and Computation", 2009, http://www.lix.polytechnique.fr/~bournez/PC2009/uploads/Main/WalidGomaaF.pdf EG .

[29] W. GOMAA. *Characterizing Polynomial Time Computability of Rational and Real Functions*, in "EPTCS - Proceedings of DCM09", vol. 9, 2009, p. 54-64, http://arxiv.org/abs/0911.2325v1, 0911.2325EG.

[30] W. GOMAA. *Polynomial Time Computation in the Context of Recursive Analysis*, in "Proceedings of FOPARA", 2009 EG .

[31] W. GUIZANI, J.-Y. MARION, D. REYNAUD. *Server-Side Dynamic Code Analysis*, in "4th International Conference on Malicious and Unwanted Software - Malware 2009, Canada Montréal", IEEE, Fernando C. Colon Osorio, 2009, p. 55-62, http://hal.inria.fr/inria-00425554/en/.

[32] M. HOYRUP, C. ROJAS. *An Application of Martin-Löf Randomness to Effective Probability Theory*, in "5th Conference on Computability in Europe - CiE 2009, Allemagne Heidelberg", K. AMBOS-SPIES, B. LÖWE, W. MERKLE (editors), vol. 5635, Springer Berlin / Heidelberg, July 2009, p. 260-269, http://hal.archives-ouvertes.fr/hal-00425556/en/.

[33] 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), vol. 5555, Springer Berlin / Heidelberg, July 2009, p. 549-561, http://hal.archives-ouvertes.fr/hal-00425560/en/.

Workshops without Proceedings

[34] P. BEAUCAMPS, E. FILIOL, J.-Y. MARION, D. REYNAUD. *On the Impact of Malware on Internet Voting*, in "1st Luxembourg Day on Security and Reliability, Luxembourg Luxembourg", 2009, http://hal.inria.fr/inria-00425584/en/.

[35] O. BOURNEZ, W. GOMAA, E. HAINRY. *Implicit complexity in recursive analysis*, in "Tenth International Workshop on Logic and Computational Complexity - LCC'09, États-Unis d'Amérique Los Angeles", 2009, http://hal.inria.fr/inria-00429964/en/EG.

[36] W. GOMAA. *Analog Computation and Function Algebras*, in "The Science and Philosophy of Unconventional Computing (SPUC09), Cambridge (UK)", 2009.

[37] D. REYNAUD, J.-Y. MARION. *Dynamic Binary Instrumentation for Deobfuscation and Unpacking*, in "IN-DEPTH SECURITY CONFERENCE 2009 EUROPE, Vienne Autriche", 2009, http://hal.inria.fr/inria-00431666/en/.

Books or Proceedings Editing

[38] S. ALBERS, J.-Y. MARION (editors). *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, Dagstuhl Seminar Proceedings, vol. 09001, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2009.

Research Reports

[39] O. BOURNEZ, W. GOMAA, E. HAINRY. *Algebraic Characterizations of Complexity-Theoretic Classes of Real Functions*, Laboratoire d'informatique de l'école polytechnique - LIX - CNRS : UMR7161 - Polytechnique - X - CARTE - INRIA Lorraine - LORIA - CNRS : UMR7503 - INRIA - Université Henri Poincaré - Nancy I - Université Nancy II - Institut National Polytechnique de Lorraine - Faculty of Engineering, Alexandria University - Alexandria University, 2009, http://hal.inria.fr/inria-00421561/en/, Rapport de rechercheEG.

[40] I. GNAEDIG, H. KIRCHNER. *Modeling reducibility on ground terms using constraints*, CARTE - INRIA Lorraine - LORIA - CNRS : UMR7503 - INRIA - Université Henri Poincaré - Nancy I - Université Nancy II - Institut National Polytechnique de Lorraine - Centre de Recherche INRIA Bordeaux - Sud-Ouest - INRIA, 2009, http://hal.inria.fr/inria-00387058/en/, Rapport de recherche.

[41] I. GNAEDIG, H. KIRCHNER. *Proving weak properties of rewriting*, CARTE - INRIA Lorraine - LORIA - CNRS : UMR7503 - INRIA - Université Henri Poincaré - Nancy I - Université Nancy II - Institut National Polytechnique de Lorraine - Centre de Recherche INRIA Bordeaux - Sud-Ouest - INRIA - INRIA, 2009, http://hal.inria.fr/inria-00338181/en/, Research Report.

[42] E. HAINRY. *Decidability and Undecidability in Dynamical Systems*, CARTE - INRIA Lorraine - LORIA - CNRS : UMR7503 - INRIA - Université Henri Poincaré - Nancy I - Université Nancy II - Institut National Polytechnique de Lorraine, 2009, http://hal.inria.fr/inria-00429965/en/, Rapport de recherche.

[43] J.-Y. MARION, D. REYNAUD. *Surfing Code Waves*, CARTE - INRIA Lorraine - LORIA - CNRS : UMR7503 - INRIA - Université Henri Poincaré - Nancy I - Université Nancy II - Institut National Polytechnique de Lorraine, 2009, http://hal.inria.fr/inria-00378667/en/, Rapport de recherche.

References in notes

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

[45] 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, http://doi.acm.org/10.1145/780542.780617.

[46] 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.

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

[48] 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/.

[49] 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, vol. 4465, Springer Berlin / Heidelberg, 2007, p. 266-275, http://hal.inria.fr/inria-00180914/en/, ISBN 978-3-540-72708-8.

[50] 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.

[51] 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.

[52] 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, p. 45-54.

[53] 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.

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

[55] 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/.

[56] 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, vol. 4664, Springer Berlin / Heidelberg, 2007, p. 12-23, http://hal.inria.fr/inria-00182738/en/, ISBN 978-3-540-74592-1.

[57] 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.

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

[59] M. L. CAMPAGNOLO, K. OJAKIAN. *The methods of approximation and lifting in real computation*, in "Third International Conference on Computability and Complexity in Analysis (CCA 2006)", 2006.

[60] 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.

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

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

[63] 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.

[64] J. FEIGENBAUM, C. H. PAPADIMITRIOU, S. SHENKER. *Sharing the Cost of Multicast Transmissions*, in "Journal of Computer and System Sciences", vol. 63, n^{o} 1, 2001, p. 21–41, http://dx.doi.org/doi:10.1006/jcss.2001.1754.

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

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

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

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

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

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

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

[79] 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/.

[80] 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), vol. 5204, Springer, 2008, p. 83–95, http://hal.inria.fr/inria-00250111/en/.

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

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

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

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

[85] D. LACEY, N. D. JONES, E. V. WYK, C. C. FREDERIKSEN. *Compiler Optimization Correctness by Temporal Logic*, in "Higher Order and Symbolic Computation", vol. 17, 2003, p. 173–206.

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

[87] C.-K. LUK, R. COHN, R. MUTH, H. PATIL, A. KLAUSER, G. LOWNEY, S. WALLACE, K. HAZELWOOD, V. J. REDDI. *Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation*, in "Programming Language Design and Implementation (PLDI)", 2005.

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

[89] 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.

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

[91] J.-Y. MARION, R. PÉCHOUX. *Analyzing the Implicit Computational Complexity of object-oriented programs*, in "FSTTCS", Dagstuhl Seminar Proceedings, vol. 08004, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2008.

[92] P. MARTIN-LÖF. *The Definition of Random Sequences*, in "Information and Control", vol. 9, n$^{\text{O}}$ 6, 1966, p. 602-619.

[93] R. MCKELVEY, A. MCLENNAN. *Computation of equilibria in finite games*, in "Handbook of Computational Economics", Elsevier, 1996, https://eprints.kfupm.edu.sa/31257/.

[94] C. MOORE. *Recursion Theory on the Reals and Continuous-Time Computation*, in "Theor. Comput. Sci.", vol. 162, n$^{\text{O}}$ 1, 1996, p. 23-44.

[95] 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, http://www.loria.fr/~moreau/Papers/MoreauRV-CC2003.ps.gz.

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

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

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

[99] 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, http://doi.acm.org/10.1145/301250.301287.

[100] 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://citeseer.ist.psu.edu/old/orponen97survey.html.

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

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

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

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

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

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

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

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

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

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