IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2011

Project-Team CARTE

Theoretical adverse computations, and safety

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

RESEARCH CENTER
Nancy - Grand Est

THEME
Programs, Verification and Proofs

Table of contents

Project-Team CARTE

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

1. Members

Research Scientists

Isabelle Gnaedig [Junior Researcher, INRIA]
Mathieu Hoyrup [Junior Researcher, INRIA]

Faculty Members

Jean-Yves Marion [Team leader, Vice head of LORIA, Professor, Nancy-University, INPL, ENSMN, HdR]
Guillaume Bonfante [Associate Professor, Nancy-University, INPL, ENSMN]
Emmanuel Hainry [Associate Professor, Nancy-University, IUT Nancy Brabois, UHP]
Romain Péchoux [Associate Professor, Nancy-University, UFR MI, Université Nancy 2]

Technical Staff

Fabrice Sabatier [Engineer, since August 2011]

PhD Students

Philippe Beaucamps [ATER UHP until September 2011, defense November 2011]
Joan Calvet [Ministère (since October 2009), defense planned in 2012, joint PhD with Ecole Polytechnique de Montréal]
Hugo Férée [ENS Lyon, since September 2011]
Thanh Dinh Ta [CORDI, since September 2010]
Stéphane Wloka [FEDER, since September 2010]

Administrative Assistant

Marie-Françoise Loubressac

2. Overall Objectives

2.1. Overall Objectives

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

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

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

- Computer Virology.
- Computation over Continuous Structures

3. Scientific Foundations

3.1. Computer Virology

From a historical point of view, the first official virus appeared in 1983 on Vax-PDP 11. In the very same time, a series of papers was published which always remain a reference in computer virology: Thompson [75], Cohen [43] and Adleman [32].

The literature which explains and discusses practical issues is quite extensive, see for example Ludwig's book [64] or Szor's one [73] and all web sites...But, we think that the best references are both books of Filiol [47] (English translation [48]) and [50]. 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 [79]. More precisely we establish in [38] that Kleene's second recursion theorem [62] is the cornerstone from which viruses and infection scenarios can be defined and classified. The bottom line of a virus behavior is

1. A virus infects programs by modifying them
2. A virus copies itself and can mutate
3. Virus spread throughout a system

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

3.2. Computation over continuous structures

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

While the notion of a computable function over discrete data is captured, according to the Church-Turing thesis, by the model of Turing machines, the situation is more delicate when the data are continuous, and several non-equivalent models exist. We mention computable analysis, which relates computability to topology [46], [78]; the Blum-Shub-Smale model (BSS), where the real numbers are treated as elementary entities [37]; the General Purpose Analog Computer (GPAC) introduced by Shannon [72] where the time is continuous.

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 [39], MAUDE [42], and TOM [69].

For basic rewriting, many techniques have been developed to prove properties of rewrite systems like confluence, completeness, consistency or various notions of termination. 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 [34], [45], [74].

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 [44]. Note that the rewriting and the completion mechanisms also enable transformation and simplification of formal systems or programs. Applications of rewriting-based proofs to computer security are various. Let us mention recent work using rule-based specifications for detection of computer viruses [76], [77].

4. Application Domains

4.1. Computer Virology

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

4.1.1. The theoretical track.

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

4.1.2. The virus detection track.

The crucial question is how to detect viruses or self-replicating malwares. Cohen demonstrated that this question is undecidable. The anti-virus heuristics are based on two methods. The first one consists in searching for virus signatures. A signature is a regular expression, which identifies a family of viruses. There are obvious defects. For example, an unknown virus will not be detected, like ones related to a 0-day exploit. We strongly suggest to have a look at the independent audit [49] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [51], this kind of methods is not yet really implemented. Moreover, the large number of false-positive implies this is barely usable. To end this short survey, intrusion detection encompasses virus detection. However, unlike computer virology, which has a solid scientific foundation as we have seen, the IDS notion of "malwares" with respect to some security policy is not well defined. The interested reader may consult [70].

4.1.3. The virus protection track.

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

4.1.4. The experimentation track.

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

4.2. Computations and Dynamical Systems

4.2.1. Continuous computation theories

Understanding computation theories for continuous systems leads to studying hardness of verification and control of these systems. This has been used to discuss problems in fields as diverse as verification (see e.g. [33]), control theory (see e.g. [40]), neural networks (see e.g. [71]), and so on.

We are interested in the formal decidability of properties of dynamical systems, such as reachability [61], the Skolem-Pisot problem [36], the computability of the ω-limit set [60]. Those problems are analogous to verification of safety properties.

Due to the difficulty of their analysis, the study of dynamical systems is often impossible without computer simulations. Nevertheless those simulations are often heuristic and due to round-off errors, what is observed on the screen is not guaranteed to reflect the actual behavior of the original mathematical system. Computable analysis has the advantage of getting rid of the truncation problems, integrating the management of errors to the computation. We then use this theory to investigate the possibility to compute characteristics of dynamical systems that are fundamental objects in the mathematical theory, such as attractors or invariant measures. Being asymptotic objects, they might not be always computable: for instance it has been proved in [41] that some Julia sets (see Figure 1) cannot be computed at all, i.e. there is no program that would plot such sets up to any resolution.

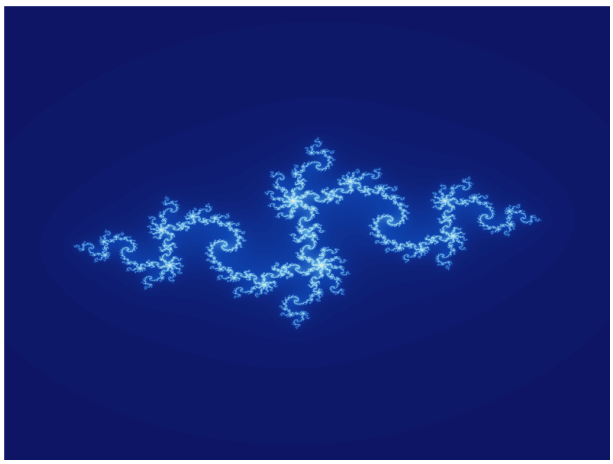

Figure 1. A Julia set: the set of points $z \in \mathbb{C}$ that do not go to ∞ when iterating $z \mapsto z^2 + c$ (here $c = -0.835 - 0.2321i$).

In [18] we prove that there exist computable systems for which the statistical long-term behavior (technically the invariant measures) is not computable.

In contrast with the discrete setting, it is of utmost importance to compare the various models of computation over the reals, as well as their associated complexity theories. In particular, we focus on the General Purpose Analog Computer of Claude Shannon [72], on recursive analysis [78], on the algebraic approach [68] and on computability in a probabilistic context [63].

A crucial point for future investigations is to fill the gap between continuous and discrete computational models. This is one deep motivation of our work on computation theories for continuous systems.

4.2.2. Analysis and verification of adversary systems

The other research direction on dynamical systems we are interested in is the study of properties of adversary systems or programs, i.e. of systems whose behavior is unknown or indistinct, or which do not have classical expected properties. We would like to offer proof and verification tools, to guarantee the correctness of such systems.

On one hand, we are interested in continuous and hybrid systems. In a mathematical sense, a hybrid system can be seen as a dynamical system, whose transition function does not satisfy the classical regularity hypotheses, like continuity, or continuity of its derivative. The properties to be verified are often expressed as reachability properties. For example, a safety property is often equivalent to (non-)reachability of a subset of unsure states from an initial configuration, or to stability (with its numerous variants like asymptotic stability, local stability,

mortality, etc ...). Thus we will essentially focus on verification of these properties in various classes of dynamical systems.

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

For several years, we have been developing specific procedures for property proofs of rewriting, for the sake of programming, in particular with an inductive technique, already applied with success to termination under strategies [52], [53], [54], to weak termination [55], sufficient completeness [57] and probabilistic termination [56].

The last three results take place in the context of adversary computations, since they allow for proving that even a divergent program, in the sense where it does not terminate, can give the expected results.

A common mechanism has been extracted from the above works, providing a generic inductive proof framework for properties of reduction relations, which can be parametrized by the property to be proved [58], [59]. 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 [65], [66], [67]. Implicit complexity is an approach to the analysis of the resources that are used by a program. Its tools come essentially from proof theory. The aim is to compile a program while certifying its complexity.

5. Software

5.1. Morphus/MMDEX

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

5.2. PYMS

Online disassembler. http://pyms86.appspot.com/

5.3. TraceSurfer

A self-modifying code analyzer coming with an IDA add-on. http://code.google.com/p/tartetatintools/

5.4. Crême Brûlée

Crême Brûlée is an experimental Javascript dynamic instrumentation engine. http://code.google.com/p/cremebrulee/

6. New Results

6.1. Resource analysis by quasi-interpretation

Participants: Guillaume Bonfante, Jean-Yves Marion.

In [15], Guillaume Bonfante, Jean-Yves Marion and Jean-Yves Moyen show how quasi-interpretations can be used to deal with the resource analysis of first order functional programs. This work has been a root for several further development in implicit computational complexity.

6.2. Characterization of programs based on embedding

Participant: Guillaume Bonfante.

So far, in the implicit complexity characterizations based on the ordering MPO developed in the team, we were using the subterm relation to compare values. In [21], we have shown that the embedding relation is a generalization of these results.

6.3. Property proofs for adversary rewrite systems

Participant: Isabelle Gnaedig.

We have continued to work on rewriting property proofs in the adversary context. Our inductive proof technique, initially developed for proving termination of rewriting for systems that do not enjoy the strong termination property, was first proposed to establish termination proofs under particular strategies: the innermost, outermost, local strategies [58].

We then have tackled the proof problem of weak properties i.e., properties that do hold only on certain derivation branches. Weak property proofs are still marginal in the domain of rewriting, probably because classical proof techniques, especially for termination, work on the rules, so that the phenomenons arising in the induced rewriting relation are hidden. Our technique, developing proof trees simulating rewriting trees by abstraction and narrowing, explicitly describes the behavior of the studied property on derivation branches, allowing to establish it on good branches. In addition, it is constructive, which is very useful in the programming context: the good branches are identified at compile time, when the proof is established. At run time, derivations are computed only on a good derivation branch, which avoids using the costly breadth-first strategy.

We then have proposed a procedure, based on our inductive principle, for weak termination and C-reducibility, which can be seen as a weak notion of sufficient completeness. The procedure principle is generic and can be instantiated by specific mechanisms related to both properties [20].

6.4. Computer virology: behavioral analysis

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

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

Last year, we had proposed to abstract trace automata by rewriting them with respect to a set of predefined behavior patterns defined as a regular language described by a string rewriting system [35]. We have increased the power of our approach on two aspects. We fist have modified the abstraction mechanism, keeping the abstracted patterns in the rewritten traces, by just marking them. This now allows us to handle interleaved patterns. Second, we have extended the rewriting framework to express data constraints on action parameters by using term rewriting systems. An important consequence is that, unlike in [35], using the data-flow, we can now detect information leaks in order to prevent unauthorized disclosure or modifications of information [28].

The previous approach has also been extended to a probabilistic model of rewriting, in order to express uncertainty in the behavior pattern recognition. All these results on detection of malware by behavior abstraction have been given in the PhD thesis of Philippe Beaucamps, directed by Isabelle Gnaedig and Jean-Yves Marion, and defended 14 November, 2011 [11].

6.5. Randomness and ergodicity: compressibility

Participant: Mathieu Hoyrup.

In [25], we solve a problem that has been open for 15 years. It relates three notions of complexity and information: Shannon information and entropy, Kolmogorov algorithmic information and Martin-Löf randomness. We obtain that the limit rate of compressibility of a random sequence equals the entropy of the underlying ergodic measure. This result is the achievement of several years of development.

6.6. Randomness and ergodicity: decomposition

Participant: Mathieu Hoyrup.

Results about the forcasting of the long-term statistics in dynamical systems. In previous works we studied the computability of the limit-frequencies. We had proved in particular that in general they cannot be computed, we have turned to the following question: can they be computed, allowing the observation of the system as an oracle? In [24], we obtain several positive results, leaving the general problem open.

6.7. Computability and measure theory

Participant: Mathieu Hoyrup.

In [26], we study the constructive content of the Radon-Nikodym theorem, show that it is not computable in general and precisely locate its non-computability in the Weihrauch lattice.

6.8. Randomness and ergodicity: limit frequencies

Participant: Mathieu Hoyrup.

A new constructive proof of Birkhoff's ergodic theorem, with as an application a strengthening of former results on random elements: in ergodic systems, random elements eventually reach effective closed sets of positive measure (while it was only known for a more restricted class of sets). The paper [19] is in press and will appear soon in Information and Computation.

6.9. Randomness for a class of measures

Participant: Mathieu Hoyrup.

New results about randomness for a class of measures (and not only for one particular measure) are presented in [14].

6.10. Decidability in Perturbed Dynamical Systems

Participant: Emmanuel Hainry.

We have studied the link between undecidability and robustness in dynamical systems. Indeed, undecidability occurs very easily in dynamical systems. However there exist good decision algorithms that work for most systems that are not pathological. We argue that this decidability trait may be related to their robustness to infinitesimal noise. We have proved that in smooth dynamical systems, robustness is equivalent to decidability of the reachability problem. This result relies on various hypotheses depending on the compactness of the domain and whether time is discrete or continuous [31]

6.11. Complexity in Recursive Analysis

Participant: Emmanuel Hainry.

In [30], we present a characterization of polytime computable functions in the Recursive Analysis setting. This paper in fact presents a generic framework for lifting characterizations of complexity or computability classes in the classical setting into analog characterizations in Recursive Analysis.

6.12. A soft linear logic characterization of polynomial space

Participant: Jean-Yves Marion.

Jean-Yves Marion has worked on light (soft) linear logics with Marco Gaboardi and Simona Ronchi Della Rocca in [16]. This work is based on an extension of a soft linear lambda calculus by means of a conditional construction. It provides a correspondence with the well-known result APTIME=PSPACE.

6.13. From control flow analysis to complexity

Participant: Jean-Yves Marion.

Jean-Yves Marion proposed a type system for an imperative programming language which certifies time bounds in [27]. It is based on secure flow information analysis as proposed for instance by Bell and La Padula. Thus, a link is done between computational complexity and security-typed languages.

7. Contracts and Grants with Industry

7.1. Contracts with Industry

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

8. Partnerships and Cooperations

8.1. National Initiatives

- ANR Complice
- Project CyS of GIS 3SGS on smartphone forensics.

We have active collaborations with:

- Alexander Shen (LIF),
- Laurent Bienvenu (LIAFA),
- Florian Deloup came in our group for six months as a CNRS researcher.

8.2. European Initiatives

8.2.1. FP7 Projet

8.2.1.1. FI-WARE

Title: Morphus

Type: COOPERATION (ICT)

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

Instrument: Integrated Project (IP)

Duration: May 2011 - April 2014

Coordinator: Telefonica (Spain)

Others partners:Thales, SAP, INRIA

See also: http://www.fi-ware.eu/

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

8.2.2. Major European Organizations with which Carte has followed Collaborations

Stefano Galatolo (Universitá di Pisa),

Daniel Graça (University of Faro),

Georg Moser (University of Innsbruck),

Klaus Weihrauch (FernUniversität Hagen).

8.3. International Initiatives

- ARC CaCO3 (France-Egypt), http://carte.loria.fr/index.php?option=com_content&view=article&id=63&Itemid=77

8.3.1. INRIA Associate Teams

8.3.1.1. COMPUTR

Title: COntinuous tiMe comPUTations, computation on the Reals

INRIA principal investigator: Emmanuel Hainry

International Partner:

Institution: Instituto de Telecomunicaçoes (Portugal)

Laboratory: Security and Quantum Information Group

Duration: 2009 - 2011

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

8.3.1.2. CRISTAL

Title: Resource Control by Semantic Interpretations and Linear Proof Theory

INRIA principal investigator: Romain Péchoux

International Partner:

Institution: Universita degli Studi di Torino (Italy)

Laboratory: Dipartimento di informatica

Duration: 2010 - 2012

See also: http://carte.loria.fr/index.php?option=com_content&view=article&id=61&Itemid=75

8.3.2. INRIA International Partners

We have active collaboration with:

- Peter Gács (Boston University),
- Cristóbal Rojas (Toronto),
- José Fernandez (Montreal),

We also start some collaborations with Dawn Song at Berkeley.

8.3.3. Visits of International Scientists

8.3.3.1. Internship

- Daniel Leivant (Indiana University, invited for six months)
- John Case (University of Delaware), http://www.cis.udel.edu/~case/
- Walid Gomaa (University of Cairo), http://www.alexeng.edu.eg/~wgomaa/

9. Dissemination

9.1. Animation of the scientific community

Guillaume Bonfante:

- has been invited to give a talk at the Workshop on Logic and Computation, http://www.jaist.ac.jp/is/labs/ogawa-lab/wlc11.html,
- has been a referee of the PhD of Jean-Marie Borello. The thesis, dealing with virology, is entitled "Étude du métamorphisme viral : modélisation, conception et détection",
- was in the jury of the PhD of Matthieu Morey. The thesis deals with Natural Language Processing,
- contributed to the papers [23], [22] whose aim is to compute the semantics of a sentence (in a natural language) from its syntactical analysis,
- is a member of the program committee of the workshop LCC 2011, http://www.cs.swansea.ac.uk/lcc2011/.

Isabelle Gnaedig:

- is co-leader of the Carte research team,
- was co-director of the PhD thesis of Philippe Beaucamps, defended 14 November, 2011,
- is member of the scientific mediation committee of INRIA Nancy Grand-Est,
- participated to the ESIAL admission committee.

Mathieu Hoyrup:

- is the organizer of the Seminar of the Department Formal Methods (http://cassis.loria.fr/fm-departement/Wiki.jsp?page=Seminar#section-Seminar-FormalMethodsSeminar),
- gave a course (2 hours) at the École Jeunes Chercheurs en Informatique et Mathématiques in Amiens, April 2011, on computability over the real numbers.

Jean-Yves Marion:

- has been in the program committee of FOPARA 2011, CSL 2011,
- has been the chair of DICE 2011, Malware 2011,
- is co-chair of Complexity and Logic week, 30/01-03/02, at the winter school at CIRM

Romain Péchoux:

- is french coordinator of the Associated team Cristal.

9.2. Teaching

Guillaume Bonfante is teaching at the Ecole des Mines:

"Java", L3,

"Modelling and UML", M1,

"Video-games", M1,

"Semantics", M1

Safety of software, M2

Isabelle Gnaedig is teaching at ESIAL (Université Henri Poincaré):

Module "Design of Safe Software", Coordination of the module, M2,

"Rule-based Programming", 20 hours, M2,

Emmanuel Hainry is teaching courses

on operating systems, algorithmics, object programming, functional programming and databases at IUT Nancy-Brabois, Université Henri Poincaré (level L1, L2)

Romain Péchoux teaches at Université Nancy 2 the following courses:

Préparation au c2i, L1, Université Nancy 2, France

Langage de programmation orienté objet, L3 MIAGE, UFR MI, Université Nancy 2, France

Java avancé, M1 MIAGE, UFR MI, Université Nancy 2, France

Base de données, L3 LSG, IAE, Université Nancy 2, France

Complexité algorithmique, L3 MIAGE, IGA Casablanca, Maroc

Romain Péchoux was responsible of "Préparation au c2i" at Nancy 2 University up to june 2011 (approximatively 3500 students) and he is director of licence MIAGE, UFR MI, Nancy 2 University since july 2011.

PhD : Philippe Beaucamps, Analyse de Programmes Malveillants par Abstraction de Comportements, Université Henri Poincaré, Nancy, defended 14th November, 2011, Directors: Jean-Yves Marion, Isabelle Gnaedig.

10. Bibliography

Major publications by the team in recent years

[1] G. BONFANTE, M. KACZMAREK, J.-Y. MARION. *Architecture of a morphological malware detector*, in "Journal in Computer Virology", 2009, vol. 5, n^{o} 3, p. 263-270 [*DOI :* 10.1007/s11416-008-0102-4], http://hal.inria.fr/inria-00330022/en/.

[2] G. BONFANTE, J.-Y. MARION, J.-Y. MOYEN. *Quasi-interpretations a way to control resources*, in "Theoretical Computer Science", May 2011, vol. 412, n^{o} 25, p. 2776-2796 [*DOI :* 10.1016/J.TCS.2011.02.007], http://hal.inria.fr/hal-00591862/en.

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

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

[5] P. GACS, M. HOYRUP, C. ROJAS. *Randomness on Computable Probability Spaces - A Dynamical Point of View*, in "Proceedings of the 26th Annual Symposium on the Theoretical Aspects of Computer Science STACS 2009", Freiburg Allemagne, S. ALBERS, J.-Y. MARION (editors), IBFI Schloss Dagstuhl, February 2009, p. 469-480, http://hal.inria.fr/inria-00360519/en/.

[6] I. GNAEDIG, H. KIRCHNER. *Proving Weak Properties of Rewriting*, in "Theoretical Computer Science", 2011, vol. 412, p. 4405-4438 [*DOI :* 10.1016/J.TCS.2011.04.028], http://hal.inria.fr/inria-00592271/en.

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

[8] M. HOYRUP. *The dimension of ergodic random sequences*, in "STACS", Paris, France, July 2011, http://hal.inria.fr/inria-00606457/en.

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

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

Publications of the year

Doctoral Dissertations and Habilitation Theses

[11] P. BEAUCAMPS. *Analyse de Programmes Malveillants par Abstraction de Comportements*, Institut National Polytechnique de Lorraine - INPL, November 2011, http://hal.inria.fr/tel-00646395/en.

[12] G. BONFANTE. *Complexité implicite des calculs : interprétation de programmes*, Institut National Polytechnique de Lorraine - INPL, December 2011, Habilitation à Diriger des Recherches, http://hal.inria.fr/tel-00656766/en.

Articles in International Peer-Reviewed Journal

[13] L. BIENVENU, A. DAY, M. HOYRUP, I. MEZHIROV, A. SHEN. *A constructive version of Birkhoff's ergodic theorem for Martin-Löf random points*, in "Information and Computation", October 2011, p. 1–16 [*DOI :* 10.1016/J.IC.2011.10.006], http://hal.inria.fr/hal-00643629/en.

[14] L. BIENVENU, P. GACS, M. HOYRUP, C. ROJAS, A. SHEN. *Algorithmic tests and randomness with respect to a class of measures*, in "Proceedings of the Steklov Institute of Mathematics", November 2011, vol. 274, n° 1, p. 34-89 [*DOI :* 10.1134/S0081543811060058], http://hal.inria.fr/hal-00644785/en.

[15] G. BONFANTE, J.-Y. MARION, J.-Y. MOYEN. *Quasi-interpretations a way to control resources*, in "Theoretical Computer Science", May 2011, vol. 412, n^{o} 25, p. 2776-2796 [*DOI :* 10.1016/J.TCS.2011.02.007], http://hal.inria.fr/hal-00591862/en.

[16] M. GABOARDI, J.-Y. MARION, S. RONCHI DELLA ROCCA. *An Implicit Characterization of PSPACE*, in "ACM Transactions on Computational Logic", May 2011, p. 1–39, http://hal.inria.fr/hal-00591868/en.

[17] P. GACS, M. HOYRUP, C. ROJAS. *Randomness on Computable Probability Spaces-A Dynamical Point of View*, in "Theory of Computing Systems", 2011, vol. 48, n^{o} 3, p. 465–485 [*DOI :* 10.1007/S00224-010-9263-X], http://hal.inria.fr/inria-00531640/en.

[18] S. GALATOLO, M. HOYRUP, C. ROJAS. *Dynamics and abstract computability: computing invariant measures*, in "Discrete and Continuous Dynamical Systems: Series A", January 2011, vol. 29, n^{o} 1, p. 193-212 [*DOI :* 10.3934/DCDS.2011.29.193], http://hal.inria.fr/inria-00517367/en.

[19] S. GALATOLO, M. HOYRUP, C. ROJAS. *Statistical properties of dynamical systems - simulation and abstract computation.*, in "Chaos, Solitons and Fractals", January 2012, vol. 45, n^{o} 1, p. 1-14 [*DOI :* 10.1016/J.CHAOS.2011.09.011], http://hal.inria.fr/hal-00644790/en.

[20] I. GNAEDIG, H. KIRCHNER. *Proving Weak Properties of Rewriting*, in "Theoretical Computer Science", 2011, vol. 412, p. 4405-4438 [*DOI :* 10.1016/J.TCS.2011.04.028], http://hal.inria.fr/inria-00592271/en.

International Conferences with Proceedings

[21] G. BONFANTE. *Course of value distinguishes the intentionality of programming languages*, in "2nd International Symposium on Information and Communication Technology - SoICT 2011", Hanoi, Viet Nam, October 2011, http://hal.inria.fr/hal-00642731/en.

[22] G. BONFANTE, B. GUILLAUME, M. MOREY, G. PERRIER. *Enrichissement de structures en dépendances par réécriture de graphes*, in "Traitement Automatique des Langues Naturelles (TALN)", Montpellier, France, 2011, http://hal.inria.fr/inria-00579251/en.

[23] G. BONFANTE, B. GUILLAUME, M. MOREY, G. PERRIER. *Modular Graph Rewriting to Compute Semantics*, in "9th International Conference on Computational Semantics - IWCS 2011", Oxford, United Kingdom, J. BOS, S. PULMAN (editors), January 2011, p. 65–74, http://hal.inria.fr/inria-00579244/en.

[24] M. HOYRUP. *Randomness and the ergodic decomposition*, in "Computability in Europe", Sofia, Bulgaria, Lecture Notes in Computer Science, June 2011, vol. 6735, p. 122-131, http://hal.inria.fr/inria-00586736/en.

[25] M. HOYRUP. *The dimension of ergodic random sequences*, in "STACS", Paris, France, July 2011, http://hal.inria.fr/inria-00606457/en.

[26] M. HOYRUP, C. ROJAS, K. WEIHRAUCH. *Computability of the Radon-Nikodym derivative*, in "Computability in Europe", Sofia, Bulgaria, B. LÖWE, D. NORMANN, I. SOSKOV, A. SOSKOVA (editors), LNCS, Springer-Verlag, June 2011, vol. 6735, p. 132-141, http://hal.inria.fr/inria-00586740/en.

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

Research Reports

[28] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. *Behavior Analysis of Malware by Rewriting-based Abstraction - Extended Version*, Institut National Polytechnique de Lorraine, May 2011, http://hal.inria.fr/inria-00594396/en.

Scientific Popularization

[29] J.-Y. MARION. *Informatique et société : Un laboratoire de haute sécurité en informatique : entretien avec Jean-Yves Marion*, in "La Recherche. Les Cahiers de l'Inria", January 2011, n° 448 janvier 2011, http://hal.inria.fr/inria-00591075/en.

Other Publications

[30] O. BOURNEZ, W. GOMAA, E. HAINRY. *Algebraic Characterizations of Complexity-Theoretic Classes of Real Functions*, 2011, Accepted for publication in International Journal of Unconventional Computing, http://hal.inria.fr/hal-00644361/en.

[31] O. BOURNEZ, D. GRAÇA, E. HAINRY. *Computation with perturbed dynamical systems*, 2011, Soumis, http://hal.inria.fr/hal-00643634/en.

References in notes

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

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

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

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

[36] 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.

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

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

[39] M.G.J. VAN DEN. BRAND, A. VAN. DEURSEN, J. HEERING, H.A. DE JONG, M. DE JONGE, T. KUIPERS, P. KLINT, L. MOONEN, P. OLIVIER, J. SCHEERDER, J. VINJU, E. VISSER, J. VISSER. *The ASF+SDF*

Meta-Environment: a Component-Based Language Development Environment, in "Compiler Construction (CC '01)", R. WILHELM (editor), Lecture Notes in Computer Science, Springer, 2001, vol. 2027, p. 365–370.

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

[41] M. BRAVERMAN, M. YAMPOLSKY. *Constructing non-computable Julia sets*, in "STOC", D. S. JOHNSON, U. FEIGE (editors), ACM, 2007, p. 709-716.

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

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

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

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

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

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

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

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

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

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

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

[53] 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.

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

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

[56] 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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