IN PARTNERSHIP WITH:
CNRS

Université de Lorraine

Activity Report 2012

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, Dynamical System, Rewriting

Creation of the Project-Team: January 01, 2009 .

1. Members

Research Scientists
Isabelle Gnaedig [Researcher, Inria]
Mathieu Hoyrup [Researcher, Inria]

Faculty Members
Emmanuel Jeandel [Professor, Université de Lorraine, since September 2012, HdR]
Jean-Yves Marion [Team leader, Vice head of LORIA, Professor, Université de Lorraine, HdR]
Guillaume Bonfante [Associate Professor, Université de Lorraine, HdR]
Emmanuel Hainry [Associate Professor, Université de Lorraine]
Romain Péchoux [Associate Professor, UFR MI, Université de Lorraine]

Engineer
Fabrice Sabatier [Engineer, since August 2011]

PhD Students
Joan Calvet [Université de Lorraine, since October 2009, joint PhD with Ecole Polytechnique de Montréal]
Hugo Férée [ENS Lyon, since September 2011]
Thanh Dinh Ta [CORDI, since September 2010]
Aurélien Thierry [CORDI, since October 2011]

Administrative Assistants
Christel Wiemert [from January 2012 until September 2012]
Véronique Constant [since September 2012]

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 systems, using tools coming from Continuous Computation Theory. The second approach is to build defenses with tools coming from logic, rewriting and, more generally, from Programming Theory.

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

- Computation over Continuous Structures
- Computer Virology.

2.2. Highlights of the Year

- We solved a problem that has been open for 15 years, relating three notions of complexity and information: Shannon information and entropy, Kolmogorov algorithmic information and Martin-Löf randomness [23].
- We developed a tool which is able to retrieve implementations of cryptographic primitives inside a trace of a binary. This result is published at CCS [22].
- We presented our work on behavioural malware detection using rewriting and model checking at ESORICS 2012 [20].
- For the Alan Turing year, we published an invited paper in the journal Phil. Trans. R. Soc. [16].

3. Scientific Foundations

3.1. Computer Virology

From a historical point of view, the first official virus appeared in 1983 on Vax-PDP 11. At the very same time, a series of papers was published which always remains a reference in computer virology: Thompson [70], Cohen [39] and Adleman [29]. The literature which explains and discusses practical issues is quite extensive [43], [45]. However, there are only a few theoretical/scientific studies, which attempt to give a model of computer viruses.

A virus is essentially a self-replicating program inside an adversary environment. Self-replication has a solid background based on works on fixed point in λ-calculus and on studies of von Neumann [75]. More precisely we establish in [35] that Kleene's second recursion theorem [58] is the cornerstone from which viruses and infection scenarios can be defined and classified. The bottom line of a virus behavior is

1. a virus infects programs by modifying them,
2. a virus copies itself and can mutate,
3. it spreads throughout a system.

The above scientific foundation justifies our position to use the word virus as a generic word for self-replicating malwares. There is yet a difference. A malware has a payload, and virus may not have one. For example, 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 a growing community of researchers working on the extension of this theory to continuous structures arising in mathematics. One goal is to give foundations of numerical analysis, by studying the limitations of machines in terms of computability or complexity, when computing with real numbers. Classical questions are : if a function $f : \mathbb{R} \to \mathbb{R}$ is computable in some sense, are its roots computable? in which time? Another goal is to investigate the possibility of designing new computation paradigms, transcending the usual discrete-time, discrete-space computer model initiated by the Turing machine that is at the base of modern computers.

While the notion of a computable function over discrete data is captured by the model of Turing machines, the situation is more delicate when the data are continuous, and several non-equivalent models exist. In this case, let us mention computable analysis, which relates computability to topology [42], [74]; the Blum-Shub-Smale model (BSS), where the real numbers are treated as elementary entities [34]; the General Purpose Analog Computer (GPAC) introduced by Shannon [68]with continuous time.

3.3. Rewriting

The rewriting paradigm is now widely used for specifying, modelizing, programming and proving. It allows to easily express deduction systems in a declarative way, and to express complex relations on infinite sets of states in a finite way, provided they are countable. Programming languages and environments with a rewriting based semantics have been developed ; see ASF+SDF [36], MAUDE [38], and TOM [64].

For basic rewriting, many techniques have been developed to prove properties of rewrite systems like confluence, completeness, consistency or various notions of termination. Proof methods have also been proposed for extensions of rewriting such as equational extensions, consisting of rewriting modulo a set of axioms, conditional extensions where rules are applied under certain conditions only, typed extensions, where rules are applied only if there is a type correspondence between the rule and the term to be rewritten, and constrained extensions, where rules are enriched by formulas to be satisfied [31], [41], [69].

An interesting aspect of the rewriting paradigm is that it allows automatable or semi-automatable correctness proofs for systems or programs: the properties of rewriting systems as those cited above are translatable to the deduction systems or programs they formalize and the proof techniques may directly apply to them.

Another interesting aspect is that it allows characteristics or properties of the modelled systems to be expressed as equational theorems, often automatically provable using the rewriting mechanism itself or induction techniques based on completion [40]. Note that the rewriting and the completion mechanisms also enable transformation and simplification of formal systems or programs.

Applications of rewriting-based proofs to computer security are various. Approaches using rule-based specifications have recently been proposed for detection of computer viruses [72], [73]. For several years, in our team, we have also been working in this direction. We already have proposed an approach using rewriting techniques to abstract program behaviors for detecting suspicious or malicious programs [32].

4. Application Domains

4.1. Computer Virology

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 [44] in order to understand the limits of this method. The second one consists in analysing the behavior of a program by monitoring it. Following [46], 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 [65].

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. [30]), control theory (see e.g. [37]), neural networks (see e.g. [66]), and so on.

We are interested in the formal decidability of properties of dynamical systems, such as reachability [56], the Skolem-Pisot problem [33], the computability of the ω-limit set [55]. Those problems are analogous to verification of safety properties.

Contrary to computability theory, complexity theory over continuous spaces is underdeveloped and not well understood. A central issue is the choice of the representation of objects by discrete data and its effects on the induced complexity notions. As for computability, it is well known that a representation is gauged by the topology it induces. However more structure is needed to capture the complexity notions: topologically equivalent representations may induce different classes of polynomial-time computable objects. For example, developing a sound complexity theory over continuous structures would enable us to make abstract computability results more applicable by analysing the corresponding complexity issues. We think that the preliminary step towards such a theory is the development of higher-order complexity, which we are currently carrying out.

In contrast with the discrete setting, it is of utmost importance to compare the various models of computation over the reals, as well as their associated complexity theories. In particular, we focus on the General Purpose Analog Computer of Claude Shannon [68], on recursive analysis [74], on the algebraic approach [63] and on computability in a probabilistic context [59].

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

4.2.2. Analysis and verification of adversary systems

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

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

We are also interested in rewriting techniques, as a mean to describe dynamical systems, in particular in the adversary context. As they were initially developed in the context of automated deduction, rewriting proof techniques, although now numerous, are not yet adapted to the complex framework of modelling and programming. An important stake in the domain is 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 [47], [48], [49], to weak termination [50], sufficient completeness [52] and probabilistic termination [51].

The last three results are in the context of adversary computations, since they allow to prove that a program can give the expected results, even when it diverges i.e., even when it has not the usual termination property. A common mechanism has been extracted from the above works, and provides a generic inductive proof framework for properties of reduction relations, which can be parametrized by the property to be proved [53], [54]. Provided that 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 [60], [61], [62]. 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

MMDEX is a virus detector based on morphological analysis. It is composed of our own disassembler tool, on a graph transformer and a specific tree-automaton implementation. The tool is used in the EU-Fiware project and by some other partners (e.g. DAVFI project).
Written in C, 20k lines.
APP License, IDDN.FR.001.300033.000.R.P.2009.000.10000, 2009.

5.2. TraceSurfer

TraceSurfer is a self-modifying code analyzer coming with an IDA add-on. It works as a wave-builder. In the analysis of self-modifying programs, one basic task is indeed to separate parts of the code which are self-modifying into successive layers, called waves. TraceSurfer extracts waves from traces of program executions. Doing so drastically simplifies program verification.
Written in C, 5k lines.
Private licence.
http://code.google.com/p/tartetatintools/

5.3. CROCUS

CROCUS is a program interpretation synthetizer. Given a first order program (possibly written in OCAML), it outputs a quasi-interpretation based on max, addition and product. It is based on a random algorithm. The interpretation is actually a certificate for the program's complexity. Users are non academics (some artists).
Written in Java, 5k lines.
Private licence.

6. New Results

6.1. Dynamical systems

Participant: Mathieu Hoyrup.

Birkhoff theorem is a central result in ergodic theory. Consider a dynamical system $(X, T : X \to X)$, start with an initial condition $x \in X$ and construct the trajectory $(x, T(x), T^2(x), ...)$. How is this trajectory distributed in X? What is the limit frequency of visits of a set $A \subseteq X$? Ergodic theorems answer to these questions by showing (i) that the distribution of *almost every* point converges and (ii) by describing the possible distributions associated to trajectories.

For several years we have been working on the project of identifying the exact computational content of several ergodic theorems: can the speed of convergence of limit frequencies be computed? Can one distinguish between points with different limit frequencies? Can we construct (compute) points whose trajectory follow a prescribed distribution? How random (i.e. incompressible) a point has to be for the distribution of its trajectory to converge?

6.1.1. Limit frequencies

We have obtained new insight in the above questions by proving that random elements eventually reach effective closed sets of positive measure (while it was only known for a more restricted class of sets). The paper appeared in *Information and Computation* [11]. This result is a key tool in the proof of the result published in [23].

6.1.2. Information

A chaotic system is unpredictable because it has much more trajectories than observable initial conditions: hence many undistinguishable initial points lead to radically different trajectories. As there are many trajectories, most of them are complex in the sense that they can hardly be compressed, i.e. described in a shorter way than simply listing them. The Shannon-McMilan-Breiman theorem states that the compression-rate of most trajectories coincides with the entropy of the system.

We have been interested in the computational content of this theorem: how random a point has to be to generate a trajectory whose compression rate is the entropy? This question was raised in [71] and has been left open for 14 years. We have solved the problem by showing that Martin-Löf notion of randomness is sufficient. Our recent result presented in [11]is a key ingredient of our proof. We presented the result at *STACS* [23].

6.1.3. Decomposition

The ergodic decomposition theorem states that a dynamical system can always be uniquely decomposed into indecomposable subsystems, technically *ergodic* subsystems. We have been interested in the computability of the decomposition operation. It is known from [71] that this operation is not computable in general. Whether this operation is still not computable when the system can be decomposed into a *finite* number of subsystems was open. We raised the question and answer it negatively in [57]. More precisely, we prove the existence of ergodic measures P and Q such that neither P nor Q is computable relative to $P + Q$. In other words, the operation of splitting a non-ergodic process into ergodic components is not computable, even in the trivial case of a combination of 2 ergodic processes. The paper is currently in press and will appear in *Annals of Pure and Applied Logic* [14].

6.2. Computations

Participant: Mathieu Hoyrup.

6.2.1. Inversion of computable functions

We strengthen the preceding result by making P and Q computable. This result is a particular case of a more general problem. In many situations an operator $F \to Y$ can be computed but can hardly be reversed: given $F(x)$, x cannot always we recovered (computed) even when F is one-to-one. We introduce a strong notion of discontinuity for the inverse of F and prove that it entails the existence of a non-computable $x \in X$ such that $F(x)$ is computable. Our result on the ergodic decomposition can be derived by applying our general result to the operator $F(P, Q) = P + Q$ which is computable but difficult to reverse. At the same time we prove a significant improvement of a classical result of Pour-El and Richards [67] about the computability of linear operators. The paper [26] is currently submitted.

6.2.2. Computability and measure theory.

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. The paper [15] appeared in the first issue of the new journal *Computability*.

6.3. Computer virology

6.3.1. Behavioral analysis

Participants: Isabelle Gnaedig, Jean-Yves Marion.

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.

We had previously 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 [32]. We then 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, which 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 [32], using the data-flow, we can now detect information leaks in order to prevent unauthorized disclosure or modifications of information.

We also have introduced model checking in our approach: the predefined behavior patterns, used to abstract program traces, have been defined by first order temporal logic formulas, as well as the reference suspicious behaviors, given in a signature. The infection problem can then be seen as the satisfaction problem of the formula of the signature by an abstracted trace of the program, which can be checked using existing model checking techniques. This work has been published at the ESORICS conference [20].

6.3.2. Analyzing cryptographic implementations

Participants: Joan Calvet, Jean-Yves Marion.

Analyzing cryptographic implementations has important applications, especially for malware analysis where they are an integral part both of the malware payload and the unpacking code that decrypts this payload. These implementations are often based on well-known cryptographic functions, whose description is publicly available. While potentially very useful for malware analysis, the identification of such cryptographic primitives is made difficult by the fact that they are usually obfuscated. Current state-of-the-art identification tools are ineffective due to the absence of easily identifiable static features in obfuscated code. However, these implementations still maintain the input-output (I/O) relationship of the original function. In a joint work with José M. Fernandez published in [22], we present a tool that leverages this fact to identify cryptographic functions in obfuscated programs, by retrieving their I/O parameters in an implementation-independent fashion, and comparing them with those of known cryptographic functions. In experimental evaluation, we successfully identified the cryptographic functions TEA, RC4, AES and MD5 in obfuscated programs. In addition, our tool was able to recognize basic operations done in asymmetric ciphers such as RSA.

6.3.3. Self-replication

Participant: Jean-Yves Marion.

Self-replication is one of the fundamental aspects of computing where a program or a system may duplicate, evolve and mutate. Our point of view is that Kleene's (second) recursion theorem is essential to understand self-replication mechanisms. An interesting example of self-replication codes is given by computer viruses. This was initially explained in the seminal works of Cohen and of Adleman in the eighties. In fact, the different variants of recursion theorems provide and explain constructions of self-replicating codes and, as a result, of various classes of malware. None of the results are new from the point of view of computability theory. We just propose a self-modifying register machine as a model of computation in which we can effectively deal with self-reproduction and in which new offsprings can be activated as independent organisms. This work was published by Jean-Yves Marion in a special issue on the honor of Alan Turing [16].

6.3.4. Reverse engineering by morphological analysis

Participants: Guillaume Bonfante, Jean-Yves Marion, Fabrice Sabatier, Aurélien Thierry.

Let us suppose we are given some malware and we want to know what it is doing. One may run it, or one may analyze it more or less statically. Typically, an expert tries to guess the behavior of a malware through the analysis of its binary code (in tools such as Ida). The task is much simpler if the expert already knows some part of the code. We have shown that morphological analysis could be used in such a context. We have rediscovered the parts of the malware `Duqu` within `Stuxnet`. We have rediscovered the compilation options used to include OpenSSL's functions within `Waledac` [21].

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

We are currently working with the consortium “malware.lu”.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

- Emmanuel Jeandel is a member of ANR Blanche ANR-09-BLAN-0164 (EMC: *Emerging Phenomena in Computation Models*).
- We obtained an ANR project called Binsec which will start in 2013. The aim of the BINSEC project is to fill part of the gap between formal methods over executable code on one side, and binary-level security analyses currently used in the security industry. We target two main applicative domains: vulnerability analysis and virus detection. Two other closely related applications will also be investigated: crash analysis and program deobfuscation.

8.2. European Initiatives

8.2.1. FP7 Projects

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: September 2011 - May 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 for building a true foundation for the Future Internet.

8.3. International Initiatives

8.3.1. Inria Associate Teams

8.3.1.1. CRISTAL

Title: Resource Control by Semantic Interpretations and Linear Proof Theory

Inria principal investigator: Romain Péchoux

International Partner (Institution - Laboratory - Researcher):

Universita degli Studi di Torino (Italy) - Dipartimento di informatica

Duration: 2010 - 2012

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

Topic: resource control using semantics interpretations and linear proof theory.

8.3.2. Participation In International Programs

Mathieu Hoyrup is the principal investigator of a Partenariat Hubert Curien Imhotep 2011-2012 together with Walid Gooma, University of Alexandria, Egypt.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

Daniel Leivant: October 25th to November 5th, 2012, Indiana University, USA.

Walid Gooma: December, 2012, University of Alexandria, Egypt.

8.4.2. Visits to International Teams

Guillaume Bonfante: July 7th to 15th, 2012, invited by Stanislas Leibler from the ’Institute of Advanced Studies’, Princeton, USA. He gave a course on computer virology at the summer school "PiTP", Prospects in Theoretical Physics http://www.sns.ias.edu/pitp2/index.html

Jean-Yves Marion: October 25th to November 5th, 2012, Indiana University, USA, work with Daniel Leivant.

Romain Péchoux: February and August 2012, University of Pennsylvania, USA, invited talk to the PLclub seminar.

9. Dissemination

9.1. Scientific Animation

9.1.1. Responsibilities

Guillaume Bonfante:

- was a member of the program committee of the workshop DICE 2012, Tallinn, Estonia,
- refereed some projects for the Netherlands Organisation for Scientific Research (NWO).

Isabelle Gnaedig is:

- co-leader of the EPI Carte,
- member of the scientific mediation committee of Inria Nancy Grand-Est,
- member of the engineer recruitment committee of Inria Nancy-Grand Est,
- social referee at Inria Nancy-Grand Est.

Mathieu Hoyrup is:

- the organizer of the Seminar of the LORIA Department “Formal Methods”.

Jean-Yves Marion is:

- head of the EPI Carte,
- head of the High Security Lab,
- member of the executive committee of Inria Nancy-Grand Est,
- member if the AERES visiting committee,
- member of the CNU, section 27,
- member of the steering committee of DICE and LCC,
- chair of Malware 2012 and PC member of PREW 2012,
- co-organizer of a winter school at CIRM (one week on complexity).

Romain Pechoux is:

- principal investigator of the Inria associate team Cristal.

9.1.2. Talks

Guillaume Bonfante:

- talk at the "Rewriting and Logic Seminar" at the JAIST (Japan Advanced Institute of Science and Technology), Kanazawa, March 2012.

Emmanuel Hainry:

- presentation at Logic and Interactions 2012, complexity week in Marseille, January 30th to February 3rd, 2012.

Mathieu Hoyrup:

- invited to give the talk "On the inversion of computable functions" at the 7th International Conference on Computability, Complexity and Randomness, in Cambridge, July 2012,
- gave the talk "Computability in Ergodic Theory” at the Journées Calculabilités, Paris, March 5 and 6, 2012.

Emmanuel Jeandel

- invited to give a four hours minicourse: "Computational aspects of multidimensional symbolic dynamics", at the first DySyCo school (Dynamical Systems and Computation) in Santiago, Chile, December 2012.

Jean-Yves Marion

- talk on computer viruses at Corte, September, 2012 and at Grenoble, December, 2012,
- two talks on computer viruses at Bloomington, Indiana University, October 29th and October 31st, 2012.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Licence:

- Guillaume Bonfante:
  - Java, L3, Ecole des Mines, Université de Lorraine, France.
- Emmanuel Hainry:
  - Operating Systems, 60 h, L1, IUT Nancy Brabois, Université de Lorraine, Nancy, France,
  - Algorithms and Programs, 60 h, L1, IUT Nancy Brabois, Université de Lorraine, Nancy, France,
  - Object Oriented Programming, 24 h, L1, IUT Nancy Brabois, Université de Lorraine, Nancy, France,
  - Databases, 24 h, L2, IUT Nancy Brabois, Université de Lorraine, Nancy, France,
  - Advanced Databases, 20 h, L2, IUT Nancy Brabois, Université de Lorraine, Nancy, France,
  - Complexity, 28 h, L2, IUT Nancy Brabois, Université de Lorraine, Nancy, France,
  - Algorithmics, 12 h, DU PFST (equiv. L1), IUT Nancy Brabois, Université de Lorraine, Nancy, France,
  - Operating Systems, 9 h, DU PFST (equiv. L1), IUT Nancy Brabois, Université de Lorraine, Nancy, France.
- Emmanuel Jeandel
  - Probability for Computer Science, 46 hours, L3 Informatique, FST, Université de Lorraine, Nancy, France.
- Romain Péchoux:
  - c2i, 20, L1 , Université de Lorraine, Nancy, France,
  - Propositional logic, 25 hours, L1 ISC, UFR MI, Université de Lorraine, Nancy, France,
  - Introduction to Java, 45 hours, L3 ISC MIAGE , UFR MI, Université de Lorraine, Nancy, France,
  - Databases, 42 hours, L3 LSG, ISAM-IAE, Université de Lorraine, Nancy, France,
  - Algorithmic complexity, 30 hours, L3 ISC MIAGE, IGA, Casablanca, Maroc.

Master:

- Guillaume Bonfante:
  - Modelling and UML, M1, Ecole des Mines, Université de Lorraine, Nancy, France,
  - Video games, M1, Ecole des Mines, Université de Lorraine, Nancy, France,
  - Semantics, M1, Ecole des Mines, Université de Lorraine, Nancy, France,
  - Safety of software, M2, Ecole des Mines, Université de Lorraine, Nancy, France.
- Isabelle Gnaedig:
  - “Design of Safe Software”, Coordination of the module, M2, ESIA-Telecom (Université de Lorraine), Nancy, France,

  - "Rule-based Programming", 20 hours, M2, ESIAL-Telecom (Université de Lorraine), Nancy, France.
- Emmanuel Jeandel:
  - Algorithms and Complexity, 35 hours, M1 Informatique, FST, Université de Lorraine, Nancy, France.
- Jean-Yves Marion:
  - Automata Theory, M1, Ecole des Mines, Université de Lorraine, Nancy, France,
  - Logic, M1, Ecole des Mines, Université de Lorraine, Nancy, France,
  - Data Base, M1, Ecole des Mines, Université de Lorraine, Nancy, France,
  - Algorithmic, M1, Ecole des Mines, Université de Lorrain, Nancy, France,
  - Security, M2, Ecole des Mines, Université de Lorraine, Nancy, France,
  - Head of the M2 internships, M2, Ecole des Mines, Université de Lorraine, Nancy, France.
- Romain Péchoux:
  - Advanced Java, 52 hours, M1 MIAGE, UFR MI, Université de Lorraine, Nancy, France,
  - Mathematics for computer science, 20 hours, M1 SCA, UFR MI, Université de Lorraine, Nancy, France.

9.2.2. Supervision

PhD : Pascal Vanier, "Pavages : périodicité et complexité calculatoire", Aix-Marseille Université, defended November 22th, 2012, Emmanuel Jeandel.

PhD in progress : Thanh Dinh Ta, Malware Algebraic Modeling and Detection, started Sept. 2010, Jean-Yves Marion (director) and Romain Péchoux (co-advisor).

PhD in progress: Hugo Férée, Computational Complexity in Analysis, started Sept. 2011, Jean-Yves Marion (director) and Mathieu Hoyrup (co-advisor).

PhD in progress: Aurélien Thierry, Morphological Analysis of Malware, started October 2011, Jean-Yves Marion.

PhD in progress: Joan Calvet, Viruses: how they interact with their environment, started October 2009, Jean-Yves Marion and José M. Fernandez, joint PhD with Ecole Polytechnique de Montréal.

9.2.3. Juries

Isabelle Gnaedig:

- participation to the ESIAL-Telecom admission committee.

Emmanuel Hainry:

- in charge of the admission committee for dept. Réseaux et Télécoms at IUT Nancy Brabois.

Mathieu Hoyrup:

- participation to the jury of Master 2 "Logique Mathématique et Fondements de l'Informatique" of Benoît Monin.

Jean-Yves Marion:

- November 30th, 2012, HDR of Virgile Mogbil, U. Paris 13 (reviewer),
- December 3th, 2012, PhD of Vincent Cheval, ENS Cachan (president),
- December 6th, 2012, PhD of Antoine Madet (reviewer),
- December 7th, 2012, PhD of Vincent Demange,
- December 13th, 2012, HDR of Emmanuel Thomé,
- December 17th, 2012, PhD of Hedi Benzina, ENS Cachan (reviewer).

Romain Péchoux:

- participation to the UFR MI, licence Miage, admission committee.

9.3. Popularization

Guillaume Bonfante:

- talk at the "Rendez-Vous du Numérique et de l'Intelligence Economique", about "Computer viruses", Strasbourg, October 2012,
- talk at the "Journées francophones de l'investigation numérique (JFIN)", about "Les virus, amis ou ennemis de l'investigation numérique ? ", Levallois-Perret, November 2012.

Emmanuel Hainry:

- presentation at Inria Jam Session 2012, June 19th, 2012 http://www.inria.fr/centre/nancy/agenda/jam-session-alumni.

Jean-Yves Marion:

- redaction of an article in "Pour la Science" http://www.pourlascience.fr/ewb_pages/f/fiche-article-les-i-mac-i-resistent-ils-mieux-aux-virus-que-les-i-pc-i-30414.php.

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. 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'12 (29th Symposium on Theoretical Aspects of Computer Science)", Paris, France, C. DÜRR, T. WILKE (editors), LIPIcs, March 2012, vol. 14, p. 567-576, http://hal.inria.fr/hal-00678161.

[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^{o} 4, p. 1–31, http://doi.acm.org/10.1145/1555746.1555751.

Publications of the year

Articles in International Peer-Reviewed Journals

[11] 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", 2012, vol. 210, p. 21-30 [*DOI :* 10.1016/J.IC.2011.10.006], http://hal.inria.fr/hal-00643629.

[12] M. GABOARDI, J.-Y. MARION, S. R. DELLA ROCCA. *An Implicit Characterization of PSPACE*, in "ACM Trans. Comput. Log.", October 2012, vol. 13, n^{o} 2 [*DOI :* 10.1145/2159531.2159540], http://hal.inria.fr/hal-00763639.

[13] 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.

[14] M. HOYRUP. *Computability of the ergodic decomposition*, in "Annals of Pure and Applied Logic", November 2012 [*DOI :* 10.1016/J.APAL.2012.11.005], http://hal.inria.fr/hal-00746473.

[15] M. HOYRUP, C. ROJAS, K. WEIHRAUCH. *Computability of the Radon-Nikodym derivative.*, in "Computability", January 2012, vol. 1, n^{o} 1, p. 3-13 [*DOI :* 10.3233/COM-2012-005], http://hal.inria.fr/hal-00726044.

[16] J.-Y. MARION. *From Turing machines to computer viruses*, in "Philosophical Transactions A: Mathematical, Physical and Engineering Sciences", 2012, vol. 370, p. 3319-3339 [*DOI :* 10.1098/RSTA.2011.0332], http://hal.inria.fr/hal-00762923.

[17] J.-Y. MARION. *Viruses in Turing's Garden*, in "ERCIM News", 2012, vol. 2012, n^{o} 91, http://hal.inria.fr/hal-00762918.

[18] J.-Y. MARION, T. SCHWENTICK. *Theoretical Aspects of Computer Science*, in "Theory Comput. Syst.", 2012, vol. 51, n^{o} 2, p. 123-124, http://hal.inria.fr/hal-00763646.

[19] R. PÉCHOUX. *Synthesis of sup-interpretations: a survey*, in "Theoretical Computer Science", November 2012, 24 [*DOI :* 10.1016/J.TCS.2012.11.003], http://hal.inria.fr/hal-00744915.

International Conferences with Proceedings

[20] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. *Abstraction-based Malware Analysis Using Rewriting and Model Checking*, in "ESORICS", Pisa, Italie, S. FORESTI, M. YUNG (editors), LNCS, Springer, 2012, vol. 7459, p. 806-823 [*DOI :* 10.1007/978-3-642-33167-1], http://hal.inria.fr/hal-00762252.

[21] G. BONFANTE, J.-Y. MARION, F. SABATIER, A. THIERRY. *Code synchronization by morphological analysis*, in "7th International Conference on Malicious and Unwanted Software (Malware 2012)", IEEE Xplore, October 2012, http://hal.inria.fr/hal-00764286.

[22] J. CALVET, J. FERNANDEZ, J.-Y. MARION. *Aligot: cryptographic function identification in obfuscated binary programs*, in "ACM Conference on Computer and Communications Security", Raleigh, États-Unis, 2012, p. 169-182, http://hal.inria.fr/hal-00762924.

[23] M. HOYRUP. *The dimension of ergodic random sequences*, in "STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)", Paris, France, C. DÜRR, T. WILKE (editors), LIPIcs, March 2012, vol. 14, p. 567-576, http://hal.inria.fr/hal-00678161.

Research Reports

[24] G. BONFANTE, V. MOGBIL. *A circuit uniformity sharper than DLogTime*, Inria, April 2012, n^{o} rapport interne LIPN, 18, http://hal.inria.fr/hal-00701420.

[25] E. HAINRY, J.-Y. MARION, R. PÉCHOUX. *Type-based complexity analysis for concurrent programs*, Inria, 2012, 15, http://hal.inria.fr/hal-00755450.

[26] M. HOYRUP. *On the inversion of computable functions*, Inria, September 2012, http://hal.inria.fr/hal-00735681.

[27] J.-Y. MARION, R. PÉCHOUX. *Complexity Information Flow in a Multi-threaded Imperative Language*, Inria, March 2012, 16, http://hal.inria.fr/hal-00684026.

Scientific Popularization

[28] J.-Y. MARION. *Les Mac résistent-ils mieux aux virus que les PC ?*, in "Pour la Science", October 2012, http://hal.inria.fr/hal-00763683.

References in notes

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

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

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

[32] P. BEAUCAMPS, I. GNAEDIG, J.-Y. MARION. *Behavior Abstraction in Malware Analysis*, in "1st International Conference on Runtime Verification", St. Julians, Malte, G. ROSU, O. SOKOLSKY (editors), Lecture

Notes in Computer Science, Springer-Verlag, August 2010, vol. 6418, p. 168-182, http://hal.inria.fr/inria-00536500/en/.

[33] P. Bell, J.-C. Delvenne, R. Jungers, V. D. Blondel. *The Continuous Skolem-Pisot Problem: On the Complexity of Reachability for Linear Ordinary Differential Equations*, 2008, http://arxiv.org/abs/0809.2189.

[34] L. Blum, M. Shub, S. Smale. *On a theory of computation and complexity over the real numbers; NP completeness, recursive functions and universal machines*, in "Bulletin of the American Mathematical Society", July 1989, vol. 21, n^{o} 1, p. 1–46.

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

[36] M.G.J. van den. Brand, A. van. Deursen, J. Heering, H.A. de Jong, M. de Jonge, T. Kuipers, P. Klint, L. Moonen, P. Olivier, J. Scheerder, J. Vinju, E. Visser, J. Visser. *The ASF+SDF Meta-Environment: a Component-Based Language Development Environment*, in "Compiler Construction (CC '01)", R. Wilhelm (editor), Lecture Notes in Computer Science, Springer, 2001, vol. 2027, p. 365–370.

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

[38] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, C. Talcott. *The Maude 2.0 System*, in "Proceedings of the 14th International Conference on Rewriting Techniques and Applications", R. Nieuwenhuis (editor), Lecture Notes in Computer Science, Springer, June 2003, vol. 2706, p. 76-87.

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

[40] H. Comon. *Inductionless Induction*, in "Handbook of Automated Reasoning", A. Robinson, A. Voronkov (editors), Elsevier Science, 2001, vol. I, chap. 14, p. 913-962.

[41] 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.

[42] 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.

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

[44] 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.

[45] E. Filiol. *Techniques virales avancées*, Springer, 2007.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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/.

[54] 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/.

[55] 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/.

[56] 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/.

[57] M. HOYRUP, C. ROJAS, K. WEIHRAUCH. *Computability of the Radon-Nikodym derivative*, in "Computability in Europe", Sofia, Bulgarie, 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.

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

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

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

[61] 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.

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

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

[64] 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.

[65] 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.

[66] 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.

[67] M. B. Pour-El, J. I. Richards. *Computability in Analysis and Physics*, Perspectives in Mathematical Logic, Springer, Berlin, 1989.

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

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

[70] 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.

[71] V. V. V'yugin. *Ergodic Theorems for Individual Random Sequences*, in "Theoretical Computer Science", 1998, vol. 207, n^{o} 4, p. 343–361.

[72] 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.

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

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

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