

Notice individuelle

Thierry CACHAT

Laboratoire d'Informatique Algorithmique : Fondements et Applications
Université Paris 7 Denis-Diderot

24 janvier 2008

Table des matières

1	Curriculum vitae	2
2	Publications	4
3	Autres exposés : séminaires, groupes de travail, ...	6
4	Enseignement	8
5	Tâches d'intérêt collectif	10
6	Système allemand	11
7	Thèse	11
8	Travaux effectués	12

1 Curriculum vitae

Thierry CACHAT né le 6 juin 1975 à Montélimar (Drôme), nationalité française

url : <http://www.liafa.jussieu.fr/~txc/>

Situation actuelle : Maître de Conférence en informatique à l'Université Paris 7

Expérience professionnelle, stages

2004–... Maître de Conférence (27) à l'Université PARIS 7 Denis Diderot, UFR d'informatique, membre du LIAFA.

2003–2004 ATER (mi-temps) au département informatique de l'École Normale Supérieure de CACHAN, recherche au Laboratoire Spécification et Vérification.

2000–2003 assistant de Wolfgang THOMAS dans l'unité Informatik VII, RWTH AACHEN (université d'Aix-la-Chapelle).

2000 Stage de DEA à l'IRISA, RENNES, encadré par Didier CAUCAL : rationalité du produit d'ensembles rationnels d'entiers. 1er février au 30 juin.

1997 Stage de 2^{ème} année de magistère à l'Université de CONSTANCE (Allemagne), encadré par Gottfried BARTHEL : amélioration de l'algorithme de calcul des bases de Gröbner avec Maple. 5 juillet au 5 septembre.

Formation

2000–2003 Doctorat en informatique de l'Université d'Aix-la-Chapelle (RWTH AACHEN) soutenue le 18 décembre 2003 : « Games on Pushdown Graphs and Extensions ». Jury : rapporteurs : Wolfgang THOMAS (directeur de thèse) et Didier CAUCAL, examinateurs : Manfred NAGL et Peter ROSSMANITH.

Doctorat en informatique de l'Université de RENNES 1 soutenue le 19 mars 2004 :

« Jeux sur des graphes d'automates à pile et leurs extensions ». Jury :

rapporteurs : Damian NIWIŃSKI et Igor WALUKIEWICZ,

examineurs : Didier CAUCAL, Antoine PETIT, Jean-Claude RAOULT et Wolfgang THOMAS.

1999–2000 DEA d'informatique de l'Université de RENNES 1 mention TB (classé 1^{er} du DEA), 4^{ème} année d'ENS.

1998–1999 Reçu à l'agrégation de mathématiques (à Paris). Début du DEA de logique et fondements de l'informatique de l'Université PARIS 7, obtenu en 2000. (ENS : congé pour convenances personnelles)

1997–1998 Entrée en 3^{ème} année à l'ENS de CACHAN, antenne de Bretagne, département de mathématiques. Préparation de l'agrégation.

1995–1997 Licence puis maîtrise de mathématiques de l'Université GRENOBLE 1, mentions TB et TB. 1^{ère} et 2^{ème} années du Magistère de Mathématiques et Applications.

1993–1995 Mathématiques supérieures puis spéciales M', lycée Clemenceau, NANTES. Reçu à l'École Centrale de Paris, à toutes les ENSI.

1993 Baccalauréat C, mention Bien, MONTÉLIMAR, Drôme.

Visites

18 au 28 décembre 2006 à l'*Institute for Information Transmission Problems (IITP) of the Russian Academy of Sciences*, Moscou, dans le cadre d'un projet CNRS-RAS entre le LIAFA et l'IITP.

Langues

Français : langue maternelle,
Allemand : courant, nombreux séjours en Allemagne,
Anglais : lu, écrit, parlé.

Programmation

C, HTML, Java, L^AT_EX, Maple, Pascal, PHP, Visual Basic.

2 Publications

Entre parenthèses figurent le nombre total d'articles acceptés et soumis à chaque conférence, lorsqu'il est connu. Ces publications sont disponibles sur

<http://www.liafa.jussieu.fr/~txc/publi.html>

Conférences internationales avec comité de lecture

- [Cac02a] Thierry Cachat. The power of one-letter rational languages. In *Proceedings of the 5th international conference Developments in Language Theory, DLT'01*, volume 2295 of *Lecture Notes in Computer Science*, pages 145–154. Springer, 2002. (24/64)
- [Cac02b] Thierry Cachat. Symbolic strategy synthesis for games on pushdown graphs. In *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming, ICALP'02*, volume 2380 of *Lecture Notes in Computer Science*, pages 704–715. Springer, 2002. (83/269)
- [CDT02] Thierry Cachat, Jacques Duparc, and Wolfgang Thomas. Solving pushdown games with a Σ_3 winning condition. In *Proceedings of the 11th Annual Conference of the European Association for Computer Science Logic, CSL'02*, volume 2471 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2002. (37/111)
- [Cac03] Thierry Cachat. Higher order pushdown automata, the Caucal hierarchy of graphs and parity games. In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming, ICALP'03*, volume 2719 of *Lecture Notes in Computer Science*, pages 556–569. Springer, 2003. (84/212)
- [Cac06a] Thierry Cachat. Controller synthesis and ordinal automata. In *Proceedings of the 4th International Symposium Automated Technology for Verification and Analysis, ATVA'06*, volume 4218 of *Lecture Notes in Computer Science*, pages 215–228, Springer, 2006. (35/137)
- [Cac06b] Thierry Cachat. Tree automata make ordinal theory easy. In *Proceedings of the 26th International Conference Foundations of Software Technology and Theoretical Computer Science, FSTTCS'06*, volume 4337 of *Lecture Notes in Computer Science*, pages 286–297. Springer, 2006. (34/155)

« Workshop » international avec comité de lecture

- [Cac02d] Thierry Cachat. Uniform solution of parity games on prefix-recognizable graphs. In Antonin Kucera and Richard Mayr, editors, *Proceedings of the 4th International Workshop on Verification of Infinite-State Systems, INFINITY'02*, volume 68 (6) of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2002.
- [ACS+07] Eugene Asarin, Thierry Cachat, Alexander Seliverstov, Tayssir Touili, and Vasily A. Lyubetsky. Attenuation regulation as a term rewriting system. In Hirokazu Anai, Katsuhisa Horimoto, and Temur Kutsia, editors, *Proceedings of the Second International Conference on Algebraic Biology, AB'07*, volume 4545 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2007. (19/40)

Autres

- [Cac02c] Thierry Cachat. Two-way tree automata solving pushdown games. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata, Logics, and Infinite Games*, volume 2500 of *Lecture Notes in Computer Science*, pages 303–317. Springer, 2002.
- [ABC⁺05] Karine Altisen, Patricia Bouyer, Thierry Cachat, Franck Cassez, and Guillaume Gardey. Introduction au contrôle des systèmes temps-réel. In Hassane Alla and Éric Rutten, editors, *Actes du 5ème Colloque sur la Modélisation des Systèmes Réactifs (MSR'05)*, pages 367–380, Autrans, France, October 2005. Hermès.
- [CW07] Thierry Cachat and Igor Walukiewicz. The complexity of games on higher order pushdown automata. Research report available at <http://hal.archives-ouvertes.fr/hal-00144226> and <http://fr.arxiv.org/abs/0705.0262>, May 2007.

3 Autres exposés : séminaires, groupes de travail, ...

1. 6 novembre 2007, *Tree Automata Make Ordinal Theory Easy*, Dagstuhl-Seminar Algorithmic-Logical Theory of Infinite Structures.
2. 8 mars 2007, *Attenuation Regulation as a Term Rewriting System*, Groupe de travail BDAA, LIF, Marseille.
3. 23 janvier 2007, *Les automates d'arbres facilitent la théorie des ordinaux*, Rencontres Arithmétique de l'Informatique Mathématique (GDR IM), RAIM 2007, Montpellier.
4. 8 décembre 2007, *Les automates d'arbres facilitent la théorie des ordinaux*, séminaire "automates" du LIAFA, Paris.
5. 16 novembre 2006, *Tree Automata Make Ordinal Theory Easy*, Groupe de travail MoVe, LIF, Marseille.
6. 23 mars 2006, *Active XML, context-free games and strategy synthesis*, Groupe de travail MoVe, LIF, Marseille.
7. 16 mars 2006, *Jeux de parité sur les graphes d'automates à pile de pile*, Séminaire PPS, Paris.
8. 13 octobre 2005, *Controller Synthesis and Ordinal Automata*, PROCOPE project Aachen-Rennes, Rennes.
9. 21 juin 2005, *Some Remarks About Active XML and Context-free Games*, Séminaire « logique et automates », Aix-la-Chapelle.
10. 10 juin 2004, *Introduction à la théorie des jeux*, Séminaire du LIF, Marseille.
11. 6 avril 2004, *Introduction à la théorie des jeux*, Séminaire du Magistère d'Informatique et Modélisation, ENS Lyon.
12. 20 novembre 2003, *Introduction à la théorie des jeux*, Séminaire Verimag, Grenoble.
13. 7 novembre 2003, *Higher Order Pushdown Automata, the Caucal Hierarchy of Graphs and Parity Games*, Séminaire automates du LIAFA, Paris.
14. 4 novembre 2003, *Higher Order Pushdown Automata, the Caucal Hierarchy of Graphs and Parity Games*, Séminaire du LSV, Cachan.
15. 1er septembre 2003, *Higher Order Pushdown Automata, the Caucal Hierarchy of Graphs and Parity Games*, « short talk », GAMES Annual Meeting, Vienne, Autriche.
16. 22 avril 2003, *Jeux sur des graphes infinis*, Séminaire général, Institut Gaspard-Monge, Marne-la-Vallée.
17. 18 avril 2003, *Méthodes symboliques pour les jeux à pile*, Séminaire Verimag, Grenoble.
18. 7 mars 2003, *Different examples of game-reduction*, Groupe de Travail « Games », LIAFA, Paris.
19. 3 mars 2003, *Quelques aspects de la jeu-simulation*, Groupe de Travail « Graphes et Logique », LaBRI, Bordeaux.
20. 20 janvier 2003, *Jeux infinis, graphes infinis*, Séminaire d'Informatique, Université de Namur.
21. 15 février 2002, *Solving Pushdown Games with a Σ_3 -Winning Condition*, Amsterdam-Aachen Exchange, Amsterdam.

22. 30 novembre 2001, *Sur la puissance des langages rationnels à une lettre*, Séminaire automates du LIAFA, Paris.
23. 26 novembre 2001, *The power of one-letter rational languages*, Université de Turku (Finlande).
24. 2 octobre 2001, *Jeux sur les graphes des automates à pile*, Séminaire 68NQRT, IRISA, Rennes.
25. 21 septembre 2001, *Symbolic computation of strategies in pushdown games*, PROCOPE project Aachen-Bordeaux, fourth workshop, Bordeaux.
26. 27 avril 2001, *Computing winning strategies in pushdown games*, PROCOPE project Aachen-Bordeaux, third workshop, Aachen.
27. 29 septembre 2000, *The power of rational languages*, PROCOPE project Aachen-Bordeaux, second workshop, Aachen.

4 Enseignement

À Aix-la-Chapelle (2000–2003)

De septembre 2000 à août 2003 j’ai enseigné à l’Université d’Aix-la-Chapelle en tant qu’assistant dans le groupe de Wolfgang Thomas. Dans le système allemand traditionnel, les assistants dépendent d’un seul professeur, à la fois pour l’enseignement et pour la recherche. Typiquement les assistants encadrent les exercices (TD) des cours donnés par leur directeur de thèse. Ainsi les enseignements que j’ai réalisés à Aix-la-Chapelle sont assez proches de mon thème de recherche, mais j’ai choisi à l’ENS de Cachan en 2003–2004 de diversifier ces enseignements, et d’enseigner d’autres sujets.

En Allemagne après le « Vordiplom » (équivalent du DEUG), le cursus universitaire n’est pas divisé en années de licence, maîtrise, DEA, DESS, ... Cependant les cours de deuxième cycle ci-dessous peuvent être comparés à des cours de maîtrise (M1). Chaque enseignement a duré un semestre, a été effectué en allemand, et implique aussi la participation à l’organisation des examens et à leur correction.

Premier cycle :

- Exercices (TD) du cours « automates, langages et complexité » en filière technique (26 h). Introduction à l’informatique théorique : modèles d’automate (fini, à pile), théorie des langages, machines de Turing, calculabilité, complexité.
- Co-encadrement du séminaire « algorithmes efficaces » (26 h), basé sur l’ouvrage : Cormen, Leiserson, Rivest : Introduction to Algorithms.
- Co-encadrement du séminaire « Pattern-Matching et applications à la Bio-Informatique » (26 h), basé sur les ouvrages :
J. Setubal and J. Meidanis, Introduction to Computational Molecular Biology, PWS Publishing Company, 1997,
M. Crochemore und W. Rytter, Text Algorithms, Oxford University Press, 1994.
- Co-encadrement du stage logiciel « plate-forme pour les algorithmes de théorie des automates » en Java (20 heures). Implantation d’une interface graphique pour dessiner des automates, et des algorithmes de déterminisation, minimisation.

Deuxième cycle (et troisième cycle) :

- Exercices (TD) du cours « théorie des automates et applications » (26 h) et participation à la rédaction des notes de cours. Minimisation, bisimulation, logiques (LTL, FO), Model Checking, automates d’arbres, documents XML, systèmes à pile, automates communiquants, MSC, systèmes hiérarchiques, réseaux de Petri.
- Exercices (TD) du cours « automates et systèmes réactifs » (26 h). Omega-automates : Staiger-Wagner, Büchi, Street, Rabin, Muller ; jeux à deux joueurs : accessibilité, parité.
- Co-encadrement du séminaire « théorie des automates et applications : spécification et vérification de systèmes de transitions infinis » (26 h). Basé sur différents articles récents.

Dans un séminaire, chaque étudiant doit présenter un article de recherche ou un chapitre de livre.

À l'ENS de Cachan (2003–2004)

Les années indiquées sont comptées à partir de l'entrée des élèves à l'ENS. Les heures sont comptées en équivalent TD.

- TP Excel / Visual Basic, 1ère année d'économie-gestion (17 h). Initiation à l'algorithmique et à la manipulation d'Excel.
- TP C++, 1ère année de génie électrique (licence d'ingénierie électrique) (34 h) Initiation à l'algorithmique et à la programmation C++.
- TD analyse statique de programmes, 2ème année d'informatique (12 h)
- TD théorie des langages, 1ère année d'informatique (20 h). Langages réguliers, algébriques, analyse syntaxique.
- cours-TD logique et automates, 2ème année d'informatique (13 h). Logique Monadique du second ordre, arbres infinis, théorème de Rabin.

À l'université Paris Diderot – Paris 7 (2004–)

Les années sont indiquées dans le système LMD (L1, L2, L3, M1, M2). Les heures sont comptées en équivalent TD.

- « Introduction à l'informatique et à la programmation » L1. Langage Java. 39 h TP en 2004-05, 32 h cours-TD en 2005-06, 65 h cours-TD en 2006-07, 70 h cours-TD en 2007-08
- « Structures de données » L1. Langage Java, arbres, listes chaînées, objets statiques. 26 h TD et 21 h TP en 2004-05
- « Concepts informatiques » L1. Passages par valeur/par référence, récursivité, pile, numérisation, compression. 26 h TD en 2006-07, 28 h TD en 2007-08
- « Programmation réseaux » L3. En Java et C. 28 h TP 2007-08
- « Grammaire et Analyse » (2004-05) et « Théorie des langages » (2005-06) L3. Automates finis, expressions rationnelles, automates à pile, grammaires hors-contexte, analyse syntaxique. 28 h TD en 2004-05, 26 h TD en 2005-06
- « Algorithmique » L3. Algorithmes de tri, complexité, arbres, algorithmes gloutons, programmation dynamique. 28 h TD 2004-05, 26 h TD 2005-06
- « Logique » L3. Induction, systèmes de preuve, calcul des prédicats. 26 h TD en 2005-06, 28 h TD en 2007-08
- « Circuits et architecture » M1. Calcul booléen, portes logiques, registres, additionneur, mini-processeur, outil tkgate, assembleur SPARC et LC-3. 26 h TD/TP en 2005-06, 26 h TD/TP en 2006-07, 28 h TD/TP en 2007-08
- « Programmation pour les méthodes numériques » M1 en Bio-Informatique. Programmation en C. Pointeurs de fonctions, problèmes numériques, gestion des fichiers, notions de compilation. 14 h TD en 2004-05
- « Modèles de spécification et vérification de protocoles » M2 Pro. Logique LTL, réseaux de Petri, automates finis, model-checking, utilisation de SPIN. 28 h TD/TP en 2004-05
- « Model Checking » M2 Pro. BDD, utilisation de CADP, UPPALL, SPIN. 27 h TD/TP en 2005-06, 27 h TD/TP en 2006-07
- « Analyse Statique de Programmes » M2 Pro. Points fixes, variables vivantes, expressions disponibles, ... 27 h TD en 2005-06, 27 h TD/TP en 2006-07
- « jeux sur des automates à pile » M2 recherche (MPRI). 4h en 2004-05
- « Graphes rationnels et automates infinis » M2 recherche (MPRI). 18h en 2006-07, 18h en 2007-08

5 Tâches d'intérêt collectif

À Aix-la-Chapelle (2000-2003) : mise à jour des pages Web des TD et des cours, de la page Web qui recense les colloques.

Au LSV (2003-2004) : mise à jour des pages Web internes « Comment faire une page Web », mise à jour de l'annuaire interne.

Au LIAFA (2004-) : participation à l'organisation de la conférence Games'05 ; animation d'un atelier sur les jeux à la fête de la science les 13 et 14 octobre 2006 et les 12 et 13 octobre 2007 : *Le carré de chocolat était empoisonné...* ; commission Web (depuis janvier 2007).

Écriture d'un script php pour le site de l'Opération Postes :

<http://postes.smai.emath.fr/echanges/Cycles/>

6 Système allemand

J'attire votre attention sur le fait qu'en Allemagne, à l'université d'Aix-la-Chapelle (RWTH Aachen) les règles de composition du jury de thèse sont les suivantes :

- le premier rapporteur d'une thèse est toujours le directeur de thèse,
 - le deuxième rapporteur est souvent un professeur de la même université, avec un domaine de recherche voisin de celui de la thèse,
 - et les deux examinateurs sont des professeurs de la même université, dont le thème de recherche doit être différent de celui de la thèse, l'un des deux venant d'un domaine appliqué.
- De plus il n'existe pas de rapport de jury de thèse au sens français du terme.

Par ailleurs une co-tutelle entre la France et l'Allemagne n'a pas été possible, parce que j'avais à Aix-la-Chapelle un poste d'assistant à temps plein sous la forme d'un contrat de deux ans renouvelable. J'ai ainsi soutenu une thèse française à Rennes.

7 Thèse

Titre : Games on Pushdown Graphs and Extensions / Jeux sur des graphes d'automates à pile et leurs extensions

Jury (RWTH Aachen) :

rapporteurs : Wolfgang THOMAS (RWTH Aachen) et Didier CAUCAL (IRISA Rennes),
examineurs : Manfred NAGL et Peter ROSSMANITH (RWTH Aachen).

Jury (Rennes 1) :

rapporteurs : Damian NIWIŃSKI (Univ. Varsovie) et Igor WALUKIEWICZ (LaBRI Bordeaux),
examineurs : Didier CAUCAL (IRISA Rennes), Antoine PETIT (ENS Cachan), Jean-Claude RAOULT (Univ. Rennes 1) et Wolfgang THOMAS (RWTH Aachen).

Résumé :

On considère des jeux à deux joueurs sur des familles de graphes infinis avec différentes conditions de gain. Notre but est de déterminer le gagnant et de calculer une stratégie gagnante.

Dans le cas des graphes de transition des automates à pile, on donne une méthode symbolique basée sur la construction d'automates finis pour résoudre les jeux d'accessibilité, de récurrence (Büchi), et les jeux avec une condition Σ_3 . Avec cette méthode on construit un automate alternant qui reconnaît l'ensemble des configurations gagnantes d'un jeu : les configurations depuis lesquelles le joueur 0 dispose d'une stratégie gagnante. Cet automate permet aussi de calculer deux types de stratégies gagnantes : soit positionnelles soit à pile.

D'un autre côté les méthodes de « réduction » existantes ont été étendues d'abord aux jeux de parité sur les graphes préfixes reconnaissables puis aux graphes des automates à pile de pile et aux graphes de la hiérarchie définie par Caucal. Ces méthodes consistent, pour résoudre un jeu, à le simuler par un autre jeu plus petit ou plus simple, que l'on sait déjà résoudre. Cela a permis aussi d'obtenir une meilleure complexité pour la procédure de décision de la logique monadique du second ordre sur ces graphes.

8 Travaux effectués

Langages rationnels d'entiers

Mon stage de DEA a porté sur l'étude d'un problème combinatoire et algébrique sur les langages rationnels. Étant donné un langage rationnel $L \subseteq A^*$ on veut déterminer si sa puissance

$$\text{pow}(L) = \{u^k \mid u \in L, k \geq 0\}$$

est aussi rationnelle (ce n'est pas toujours le cas). Le problème général est resté ouvert depuis 1996 [Cal96, CN96], mais, dans le cas d'un alphabet A à une lettre, j'ai donné une procédure de décision [Cac02a]. Même dans ce cas restreint, qui se ramène aux langages rationnels d'entiers, la solution est assez complexe.

Jeux sur des graphes infinis

J'étais membre du projet "Game" de l'Union Européenne, *Games and Automata for Synthesis and Validation*, qui associait les universités suivantes : Aachen, Bordeaux I, Edinburgh, Paris 7, Rice, Uppsala, Vienna University of Technology, Warsaw, voir <http://www.games.rwth-aachen.de/>.

Pendant ma thèse j'ai étudié différents jeux sur des familles de graphes infinis. Ce sujet est motivé par des problèmes de vérification. J'ai considéré le paradigme des systèmes réactifs, où deux entités, un contrôleur et un environnement, interagissent. Le but du contrôleur est de répondre à chaque action possible de l'environnement de telle sorte que l'interaction produite satisfasse des conditions comme « aucun accident n'a lieu », « quelque chose est produit », ... D'un point de vue formel il est utile et naturel de voir ces systèmes comme des jeux à deux joueurs, où l'un est le contrôleur et l'autre l'environnement. Le contrôleur gagne une partie si et seulement si la condition requise est satisfaite.

Dans ma thèse je suis parti d'une situation où le problème est déjà modélisé comme un jeu à deux joueurs, et je me suis restreint aux jeux :

- alternés : les deux joueurs jouent à tour de rôle,
- à information parfaite : les deux joueurs connaissent parfaitement l'état actuel du jeu ainsi que les états antérieurs,
- sans hasard.

(Le jeu d'échec satisfait ces conditions, beaucoup de jeux de cartes utilisent le hasard et l'information incomplète, pierre-ciseaux-papier n'est pas alterné, ...) On s'intéresse aux jeux infinis : une partie est une suite de coups indexée par ω .

Notre but est de déterminer si le contrôleur peut gagner « à coup sûr » un jeu donné, quelque soit le comportement de l'environnement, et d'implémenter — si c'est possible — une stratégie gagnante pour le contrôleur. Dans le cas des graphes finis ces problèmes sont connus depuis longtemps, même si des questions de complexité et d'efficacité algorithmiques restent ouvertes. Dans ma thèse nous nous sommes attachés à quelques classes de graphes infinis. Ces graphes sont nécessaires pour modéliser des programmes avec des variables non bornées ou des appels récursifs de procédures.

Nous avons développé deux types de technique : une approche symbolique basée sur des automates finis qui reconnaissent des ensembles infinis de configurations et une *jeu-simulation* qui permet de réduire un jeu donné à un jeu plus simple et de le résoudre.

Méthode symbolique pour les jeux Pour manipuler les graphes de transition des automates à pile, qui sont en général infinis, l’approche symbolique utilise des automates finis, qui représentent des ensembles infinis de configurations. Ici une configuration est un mot pw où p est un état de contrôle et w est le mot de pile (un mot sur l’alphabet de pile). Des algorithmes sont déjà connus pour l’accessibilité et le model-checking des graphes d’automate à pile [BEM97, EHS00a]. Nous les avons réutilisés et étendus pour calculer aussi des stratégies gagnantes, sous deux formes : des stratégies positionnelles et des stratégies à pile [Cac02b]. De cette façon on est passé du model-checking à la synthèse de programmes.

Nous avons ensuite considéré des *jeux à pile* (jeux sur des graphes d’automate à pile) avec une condition de gain de type Büchi : le joueur 0 gagne si et seulement si l’ensemble « but » est visité infiniment souvent. Là aussi on peut calculer des stratégies gagnantes [Cac02b]. Enfin nous avons étudié une nouvelle condition de gain de type Σ_3 : le joueur 0 gagne si et seulement si il existe une configuration du jeu qui est visitée infiniment souvent. Comme le graphe est infini, cette quantification sur « une configuration » ne se réduit pas à une condition de Büchi. Elle permet aussi de résoudre le problème de « l’explosion de pile » : est-ce que le programme a besoin d’une mémoire non bornée ? Notre solution est une extension de l’algorithme pour les jeux de Büchi est peut être considérée comme un nouveau résultat de décidabilité [CDT02].

Jeu-simulation La jeu-simulation existe depuis longtemps en mathématiques, mais elle n’y est pas toujours utilisée de manière effective. L’idée est de réduire un jeu donné à un jeu plus simple que l’on sait résoudre, puis d’en déduire le gagnant ainsi qu’une stratégie gagnante dans le jeu initial. Ce type de méthode a déjà été utilisé [Wal01] pour résoudre les jeux de parité sur les graphes à pile en les réduisant à des jeux sur des graphes finis (exponentiellement plus grands). Pour faire face à l’infinité des contenus de pile accessibles, on force un des joueurs à deviner et annoncer ce qui arrivera plus tard si l’on dépile, et on peut condenser cette information. Nous avons réutilisé cette réduction et en avons donné une nouvelle preuve, plus directe et intuitive. De plus nous avons construit explicitement une stratégie à pile pour le jeu initial à partir du jeu réduit sur le graphe fini [Cac02d]. Jusqu’alors l’analyse était faite uniquement depuis une position initiale particulière (avec une pile vide). Nous l’avons étendue à une position initiale quelconque, ce qui donne une solution uniforme, et nous avons montré que la région gagnante d’un joueur est un ensemble rationnel de configurations. La région gagnante est ici l’ensemble des configurations depuis lesquelles ce joueur peut gagner à coup sûr.

Les résultats de [Var98, KV00] peuvent aussi être vus comme une jeu-simulation. Techniquement l’étape essentielle est la transformation des automates d’arbres bidirectionnels (boustrophédons) en unidirectionnels [Cac02c]. En effet, déterminer l’existence d’un calcul acceptant revient à déterminer le gagnant d’un jeu particulier, où un joueur veut montrer qu’il existe un tel calcul acceptant, et l’autre joueur veut le réfuter. Nous avons étendu cette réduction au cas des arbres à degré non borné ou infini. Cela a permis de résoudre les jeux de parité sur les graphes de la hiérarchie infinie de Caucal, puis les jeux de parité sur les graphes d’automates à pile d’ordre supérieur (pile de pile, ...) [Cac03]. Cela a permis aussi d’obtenir une meilleure complexité pour la procédure de décision de la logique monadique du second ordre sur ces graphes. Nous avons montré ensuite que la complexité de ces constructions était optimale [CW07].

Contrôle de systèmes temporisés / Information partielle

D’octobre 2003 à novembre 2006 j’ai participé aux travaux de l’ACI Cortos, *Control and Observation of Real-Time Open Systems*, avec les laboratoires IRCCyN (Nantes), LSV (Cachan), VERIMAG (Grenoble), voir <http://www.lsv.ens-cachan.fr/aci-cortos/>. Depuis novembre 2006 je suis membre du projet de recherche ANR “DOTS” *Distributed Open and Timed Systems*, avec les laboratoires IRCCyN (Nantes), IRISA (Rennes), LaBRI (Bordeaux), LAMSADE (Paris), LSV (Cachan), voir <http://www.lsv.ens-cachan.fr/anr-dots/>.

Le cadre des systèmes temporisés ajoute une dimension supplémentaire aux problèmes de jeu et de contrôle : les deux joueurs peuvent agir simultanément, et leurs actions interviennent à des dates qui ne sont pas connues à l’avance [ABC⁺05].

Une partie de mes travaux actuels concerne les jeux à information partielle : un des joueurs, le contrôleur, ne connaît pas parfaitement l’état du jeu, et il doit prendre une décision avec les informations dont il dispose. Cela peut apparaître en particulier dans le contexte de l’observation partielle, où le contrôleur n’a accès qu’à une partie des variables du système, ou quand les deux joueurs jouent simultanément.

Les automates ordinaux reconnaissent des suites indexées par des ordinaux. Ils permettent de modéliser des systèmes physiques qui ont des comportements Zénon. Demri et Nowak [DN05] ont défini une logique $LTL(\omega^k)$ pour exprimer des propriétés sur les suites de longueur ω^k , $k \geq 1$. Cependant on veut construire un contrôleur dont les exécutions sont des suites habituelles, indexées par ω . Le système physique réagit donc plus vite que le contrôleur, et le contrôleur ne peut pas observer toutes les actions intermédiaires du système. Nous avons résolu ce problème de contrôle à l’aide d’un jeu à information partielle [Cac06a].

Théories logiques des ordinaux

La décidabilité de la logique de Presburger (premier ordre avec addition) peut-être montrée à l’aide des automates finis. Un automate reconnaît le codage binaire des entiers qui satisfont une formule, la démonstration se fait par induction sur la structure de la formule grâce aux propriétés de clôture des automates finis [Kla04]. Nous avons étendu cette méthode à la décidabilité de la logique du premier ordre avec addition des ordinaux de type ω^{ω^i} [Cac06b]. Pour cela nous avons codé ces ordinaux avec des arbres binaires finis et utilisé des automates d’arbre. Cela donne une démonstration plus simple et une meilleure complexité que les résultats connus [Mau96].

Büchi a montré la décidabilité de la logique monadique du second ordre sur les entiers, ou $MSO(\omega, <)$, à l’aide des automates qui portent aujourd’hui son nom, puis il a étendu ce résultats aux ordinaux grâce aux automates ordinaux [Büc65]. Les tentatives pour améliorer la complexité de ces constructions ont conduit à des codages très complexes [Mau96]. Avec des arbres binaires infinis il est facile des représenter les sous-ensembles d’ordinaux inférieurs à ω^i , et avec les automates d’arbre de Rabin on obtient une nouvelle démonstration, simple, et avec une meilleure complexité [Cac06b].

Modélisation de la structure secondaire de l’ARN

Dans le cadre du projet Evolver qui associe le CNRS et l’Académie des Sciences de Russie, le LIAFA (Paris) avec l’IITP (Moscow).

Maintenant que l'on connaît le génome de nombreux organismes vivants, une question essentielle de la biologie est de savoir comment les gènes s'expriment. Pour qu'une séquence d'ADN s'exprime, une polymérase doit d'abord la transcrire en une séquence d'ARN, puis un ribosome doit traduire cet ARN en une chaîne d'acides aminés. Dans le cas des bactéries (sans noyau) les deux processus de transcription et traduction ont lieu simultanément et la séquence d'ARN intermédiaire peut se replier selon les bases qui sont complémentaires. D'après le modèle CAR, *Classical Attenuation Regulation* [LRSP06], la régulation de l'expression d'un gène est déterminée par cette structure secondaire qui influence la vitesse de la polymérase et peut même le "décrocher" de la séquence d'ADN. Dans [ACS⁺07] nous proposons d'utiliser la réécriture de termes pour prédire les probabilités de formation de ces structures secondaires de façon plus globale que la simulation Monte-Carlo de [LRSP06].

Références

- [ABC⁺05] Karine Altisen, Patricia Bouyer, Thierry Cachat, Franck Cassez, and Guillaume Gardey. Introduction au contrôle des systèmes temps-réel. In Hassane Alla and Éric Rutten, editors, *Actes du 5ème Colloque sur la Modélisation des Systèmes Réactifs (MSR'05)*, pages 367–380, Autrans, France, October 2005. Hermès.
- [ACS⁺07] Eugene Asarin, Thierry Cachat, Alexander Seliverstov, Tayssir Touili, and Vasily A. Lyubetsky. Attenuation regulation as a term rewriting system. In Hirokazu Anai, Katsuhisa Horimoto, and Temur Kutsia, editors, *Proceedings of the Second International Conference on Algebraic Biology, AB'07*, volume 4545 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2007.
- [BEM97] Ahmed Bouajjani, Javier Esparza, and Oded Maler. Reachability analysis of pushdown automata : Application to model-checking. In *Proceedings of the 8th International Conference on Concurrency Theory, CONCUR '97*, volume 1243 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 1997.
- [Büc65] J. Richard Büchi. Decision methods in the theory of ordinals. *Bull. Amer. Math. Soc.*, 71 :767–770, 1965.
- [Cac02a] Thierry Cachat. The power of one-letter rational languages. In *Proceedings of the 5th international conference Developments in Language Theory, DLT'01*, volume 2295 of *Lecture Notes in Computer Science*, pages 145–154. Springer, 2002.
- [Cac02b] Thierry Cachat. Symbolic strategy synthesis for games on pushdown graphs. In *Proceedings of the 29th International Colloquium on Automata, Languages, and Programming, ICALP'02*, volume 2380 of *Lecture Notes in Computer Science*, pages 704–715. Springer, 2002.
- [Cac02c] Thierry Cachat. Two-way tree automata solving pushdown games. In Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors, *Automata, Logics, and Infinite Games [GTW02]*, volume 2500 of *Lecture Notes in Computer Science*, pages 303–317. Springer, 2002.
- [Cac02d] Thierry Cachat. Uniform solution of parity games on prefix-recognizable graphs. In Antonin Kucera and Richard Mayr, editors, *Proceedings of the 4th International Workshop on Verification of Infinite-State Systems, INFINITY'02*, volume 68 (6) of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, 2002.
- [Cac03] Thierry Cachat. Higher order pushdown automata, the Caucal hierarchy of graphs and parity games. In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming, ICALP'03*, volume 2719 of *Lecture Notes in Computer Science*, pages 556–569. Springer, 2003.
- [Cac06a] Thierry Cachat. Controller synthesis and ordinal automata. In Susanne Graf and Wenhui Zhang, editors, *Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis, ATVA'06*, volume 4218 of *Lecture Notes in Computer Science*, pages 215–228. Springer, 2006.
- [Cac06b] Thierry Cachat. Tree automata make ordinal theory easy. In *Proceedings of the 26th International Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS'06*, volume 4337 of *Lecture Notes in Computer Science*, pages 285–296. Springer, 2006.

-
- [Cal96] Hugues Calbrix. *Mots ultimement périodiques des langages rationnels de mots infinis*. PhD thesis, université Denis Diderot–Paris VII, LITP Th96/03, 1996.
- [CDT02] Thierry Cachat, Jacques Duparc, and Wolfgang Thomas. Solving pushdown games with a Σ_3 winning condition. In *Proceedings of the 11th Annual Conference of the European Association for Computer Science Logic, CSL'02*, volume 2471 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2002.
- [CN96] Hugues Calbrix and Maurice Nivat. Prefix and period languages of rational ω -languages. In *Proceedings of the international conference Developments in Language Theory II, DLT'95*, pages 341–349. World Scientific, 1996.
- [CW07] Thierry Cachat and Igor Walukiewicz. The complexity of games on higher order pushdown automata. available at <http://hal.archives-ouvertes.fr/hal-00144226> and <http://fr.arxiv.org/abs/0705.0262>, May 2007.
- [DN05] Stéphane Demri and David Nowak. Reasoning about transfinite sequences (extended abstract). In Doron A. Peled and Yih-Kuen Tsay, editors, *Proceedings of the 3rd International Symposium on Automated Technology for Verification and Analysis (ATVA'05)*, volume 3707 of *Lecture Notes in Computer Science*, pages 248–262, Taipei, Taiwan, ROC, October 2005. Springer.
- [EHR00a] Javier Esparza, David Hansel, Peter Rossmanith, and Stefan Schwoon. Efficient algorithms for model checking pushdown systems. In *Proceedings of the 12th International Conference on Computer Aided Verification, CAV'00*, volume 1855 of *Lecture Notes in Computer Science*, pages 232–247. Springer, 2000. see also the full version [EHR00b].
- [EHR00b] Javier Esparza, David Hansel, Peter Rossmanith, and Stefan Schwoon. Efficient algorithms for model checking pushdown systems. Technical Report TUM-I0002, SFB-Bericht 342/1/00 A, Technische Universität München, February 2000. full version of [EHR00a].
- [GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games : A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.
- [Kla04] Felix Klaedtke. On the automata size for Presburger arithmetic. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS 2004)*, pages 110–119. IEEE Computer Society Press, 2004. A full version of the paper is available from the author's web page.
- [KV00] Orna Kupferman and Moshe Y. Vardi. An automata-theoretic approach to reasoning about infinite-state systems. In E. A. Emerson and A. P. Sistla, editors, *Proceedings of the 12th International Conference on Computer Aided Verification, CAV'00*, volume 1855 of *Lecture Notes in Computer Science*, pages 36–52. Springer, 2000.
- [LRSP06] Vassily Lyubetsky, Lev Rubanov, Alexander Seliverstov, and Sergey Pirogov. Model of gene expression regulation in bacteria via formation of RNA secondary structures. *Molecular Biology*, 40(3) :440–453, 2006.
- [Mau96] Françoise Maurin. Exact complexity bounds for ordinal addition. *Theoretical Computer Science*, 165(2) :247–273, 1996.

- [Var98] Moshe Y. Vardi. Reasoning about the past with two-way automata. In Kim Guldstrand Larsen, Sven Skyum, and Glynn Winskel, editors, *Proceedings of the 25th International Colloquium on Automata, Languages and Programming, ICALP'98*, volume 1443 of *Lecture Notes in Computer Science*, pages 628–641, Aalborg, Denmark, 1998. Springer.
- [Wal96] Igor Walukiewicz. Pushdown processes : Games and model checking. In *Proceedings of the 8th International Conference on Computer Aided Verification, CAV'96*, volume 1102 of *Lecture Notes in Computer Science*, pages 62–74. Springer, 1996. see also the full version [Wal01].
- [Wal01] Igor Walukiewicz. Pushdown processes : Games and model checking. *Information and Computation*, 164(2) :234–263, January 2001. Full version of [Wal96].