

Programme jeunes chercheuses et jeunes chercheurs 2005

B - Description du projet

Description du projet et résultats attendus

Parmi les structures mathématiques, les structures d'ordre ont reçu une attention particulière en informatique théorique. Dans le cadre de la théorie de la calculabilité et de la sémantique des langages de programmation, la théorie des domaines [28] repose de manière essentielle sur la notion d'ordre. La suite des mots possibles dans un système de réécriture donne lieu à une structure ordonnée qui a souvent de propriétés très intéressantes d'un point de vue algébrique [10, 9]. Plus généralement, la suite des événements possibles dans un système dynamique peut être analysé à l'aide des structures d'ordres dérivés [13]. Enfin, les notions de structures ordonnées interviennent dans la modélisation des systèmes concurrents [24, 11].

Cette présentation rapide sur l'utilisation des structures d'ordre en informatique montre l'existence de plusieurs thèmes de recherche qui ne partagent pas la même culture théorique mais qui étudient toutes ces structures d'ordre. Cela suggère des rapprochements entre ces thèmes qui peuvent être fructueux, d'autant plus qu'il n'y a eu jusqu'à présent que peu d'interaction entre les différentes communautés. Notre projet peut être vu comme un projet interdisciplinaire entre mathématiques et différentes branches de l'informatique, et son objectif est d'approfondir et rapprocher les connaissances théoriques portant sur les structures ordonnées. De plus il permettra d'orienter la recherche théorique vers des questions liées à des problèmes plus spécifiques de l'informatique proches de nos thèmes de recherche.

La recherche fondamentale

Structures d'ordre et systèmes concurrents. À un niveau abstrait, le comportement d'un système est décrit par un ensemble d'états et par une (ou plus) relation de transition entre les états. La relation d'accessibilité est la clôture réflexive et transitive de la relation de transition. En général cette relation est un ordre partiel sur l'ensemble des états et la question classique qu'on se pose alors est d'étudier cet ordre ¹. Dans le cas d'un système séquentiel, cet ordre est une arborescence. Il s'agit donc d'un ensemble ordonné très particulier qui, à notre avis, ne justifie pas le recours à la théorie des ensembles ordonnés dans sa généralité afin d'étudier les systèmes séquentiels. La question est plus compliquée si l'on considère des systèmes distribués, parallèles et concurrents (nous dirons concurrent dans la suite). En utilisant un mot emprunté à la théorie des groupes, les comportements de ces systèmes ont plus de "torsion". La relation d'accessibilité devient un ordre partiel assez complexe, ce qui rend prometteur l'étude de cette structure à l'aide du langage mathématique qui est lui propre. Il existe déjà en littérature [21] des caractérisations des structures d'ordre qui correspondent à un certain niveau de concurrence.

¹Nous rappelons que la structure d'ordre est souvent enrichie par des informations supplémentaires (comme des étiquettes d'action), afin de représenter fidèlement le comportement des systèmes.

Structures d'ordre infinies et leur présentation par méthodes finies. Un problème fondamental de l'informatique est de décrire des objets infinis par des méthodes finies : étant donné un langage de mots, on se demande s'il existe (et si l'on est capable de construire) un automate, satisfaisant certaines contraintes, reconnaissant ce langage ou bien, étant donné une fonction, on se demande s'il existe une machine de Turing qui calcule cette fonction tout en respectant certaines limites de temps et espace (polynomial par exemple). Quand on se place dans la théorie de la concurrence, le même problème se pose. Une fois que le comportement abstrait d'un système concurrent est assimilé à un ensemble ordonné éventuellement avec des étiquettes, le problème revient à décrire de manière finie de tels ensembles ordonnés infinis. Il existe en littérature plusieurs méthodes permettant de décrire des ordres infinis, nous pensons essentiellement aux d'automates avec relation de commutation implicite et/ou explicite et aux réseaux de Pétri. Il s'agit d'abord d'effectuer une étude complète sur les différents types d'automates, pour enfin pouvoir les comparer et les classer. Nous proposons aussi d'étudier des méthodes de présentation des ordres partiels infinis avec concurrence, ayant un caractère plus mathématique, c.-à-d. qui ne correspondent pas nécessairement à une notion d'automate. Des structures mathématiques et des concepts théoriques devraient permettre de surmonter cette difficulté. Nous pensons d'abord, aux structures d'ordre circulaires [22]. Ces structures ont récemment trouvé un regain d'intérêt dans le domaine de la logique mathématique [2, 25] et de la complexité en combinatoire [16]. Notre intérêt pour ces structures s'appuie sur l'analogie suivante : une permutation circulaire est naturellement associée à un ordre linéaire, comme un ordre circulaire fini est naturellement associé à un ordre fini. Comme un cycle donne lieu à un mot infini, un ordre circulaire donne lieu à un ordre infini dont les sommets sont étiquetés. Des autres outils mathématiques plus généraux peuvent intervenir, par exemple la théorie des espèces de structures [17], la théorie de solutions des systèmes d'équations [6, 3], et, pour finir, la théorie des co-algèbres de foncteurs [26].

Structures d'ordre finies : combinatoire et algèbre. L'étude du langage des structures ordonnées liées pour les systèmes infinis est un sujet prometteur à notre avis, mais il n'est pas encore possible d'estimer clairement quels seront les résultats pratiques qui en découleront. Par contre l'intérêt de cette étude dans le cas des systèmes finis est maintenant bien établie par ses applications dans le domaine de la concurrence. Même si de nombreux travaux ont été effectués, il reste encore matière à recherche, en particulier pour mieux comprendre pourquoi les techniques proposées fonctionnent bien.

La recherche portant sur la sûreté des logiciels utilise des méthodes qui modélisent un système réel par un modèle mathématique, usuellement un automate fini, sur lequel on peut effectuer un certain nombre de calculs de façon mécanique (comme par exemple, vérifier si ce modèle satisfait une propriété exprimée dans une logique appropriée, basée sur la relation d'accessibilité). Dans ce passage au modèle mathématique, le nombre d'états du système résultant est souvent trop grand pour pouvoir effectuer une exploration complète de l'ensemble des états. En effet, les systèmes concurrents obtenus par composition de systèmes simples donnent lieu à une explosion du nombre d'états liée à la sémantique de la concurrence. Celle-ci correspond aux entrelacements des exécutions des systèmes, avec synchronisation pour les actions communes, ce qui peut donner un nombre d'état exponentiel en le nombre de composants. Il faut donc concevoir des algorithmes qui exploitent les propriétés liées à la concurrence pour pouvoir calculer en pratique (ce qui revient à explorer efficacement le graphe des états). La concurrence intervient ici car la torsion de l'ordre associé aux système, la relation d'accessibilité, permet de visiter les états, l'un après l'autre, en réduisant

l'ensemble des chemins à considérer pour atteindre un état. En utilisant ces idées, et en exploitant les propriétés combinatoires des structures ordonnées associées à la relation d'accessibilité, on a abouti à des heuristiques très prometteuses [20, 7]. Le logiciel **ELSE**, en cours de développement dans l'équipe, nous permettra de tester différentes heuristiques sur des systèmes concurrents obtenus par composition de systèmes simples. Une meilleure compréhension des ordres utilisés permettra d'une part de mieux comprendre l'efficacité de ces heuristiques, et éventuellement de caractériser les systèmes sur lesquels elles sont efficaces. Par ailleurs nous espérons pouvoir étudier les notions concernant l'indépendance dynamique pour laquelle certaines actions commutent uniquement dans un certain contexte : par exemple deux actions a et b commutent uniquement dans un certain état du système. Différents axiomes ont été proposés pour modéliser cette notion et caractériser les bonnes propriétés nécessaires à une étude de ce type de système. Nous pensons qu'une approche cherchant à établir des propriétés des ordres correspondants peut permettre de mieux comprendre leur fonctionnement et de pouvoir proposer des méthodes algorithmiques pour traiter les systèmes concurrents ayant de l'indépendance dynamique.

D'un autre côté, l'étude approfondi de quelques systèmes de réécriture simples, notamment la réécriture des mots de Coxeter [10], et de certains systèmes dynamiques [13], a montré que les ordres partiels associés ont souvent des propriétés fort intéressantes d'un point de vue algébrique. Rappelons qu'un treillis est un ensemble ordonné pour lequel il existe un plus petit majorant et un plus grand minorant de tout sous ensemble fini non vide. Ces systèmes, de réécriture et dynamiques, donnent lieu à des ordres qui sont des treillis. En plus, ces treillis possèdent souvent des propriétés algébriques assez fortes, par exemple la semi-distributivité. Si ces propriétés algébriques sont bien connues dans le cadre classique de la modélisation des données [12], leurs interprétation dans le domaine de la concurrence nous semble être moins connue. Par exemple, il nous semble évident que l'absence de concurrence dans un système (ou de paires critiques dans un système de réécriture), correspond à la propriété de distributivité d'un treillis. Un travail récent [27] suggère qu'il est possible donner une généralisation de cette correspondance en termes de semi-distributivité.

Nous nous retrouvons donc avec deux domaines de recherche, l'un plus appliqué et l'autre plus théorique, où les structures d'ordre et les treillis jouent un rôle fondamentale. Nous proposons de travailler sur les deux aspects. Ce travail pourra suggérer des heuristiques fines pour l'exploration des états, et un enrichissement du cadre théorique dont l'intérêt est d'être extérieur à cette théorie.

Les outils à développer

AMSC : un analyseur marseillais de scénarios (responsable Rémi Morin)

Un « message sequence chart » est une description graphique d'un scénario de communication entre plusieurs composants d'un système. Il correspond à ce que l'on nomme en français les diagrammes de séquence. La norme Z.120 de l'*international telecommunication union* définit les conventions de représentation de ces diagrammes ainsi la syntaxe de leur description textuelle. En fait les MSC forment une classe particulière d'ordres partiels. Un MSC de haut-niveau (HMSC) est un automate fini dont les états (ou, de manière équivalente, les transitions) sont étiquetés par des MSC. Comme pour les MSC, les HMSC bénéficient d'une syntaxe formelle normalisée. Une exécution d'un HMSC est obtenue en juxtaposant les MSC rencontrées le long d'un chemin fini qui mène de l'état initial à un état final. Une fois un HMSC spécifié, l'outil **AMSC** permet de vérifier automatiquement certaines propriétés sur l'ensemble de ses exécutions. Une introduction à quelques tendances récentes de la

recherche sur le langage de spécification des MSC est proposée à [19].

La sémantique courante des MSC fait appel d'une façon naturelle aux structures d'ensembles d'ordre partiel. On peut penser aux HMSC comme à un formalisme permettant de définir des ordres partiels, éventuellement infinis. Pourtant, nous jugeons le développement de AMSC naturellement lié aux thématiques sur les structures d'ordre sur lesquelles nous voulons nous pencher.

Qu'est-ce que c'est que AMSC ?

- Une interface graphique Java pour la spécification de diagrammes de séquence (en anglais, *message sequence charts* ou MSC) selon la norme Z.120 de l'*international telecommunication union* (ITU).
- Une implémentation de quelques algorithmes de vérification développés au *Laboratoire d'Informatique Fondamentale de Marseille* dans l'équipe *Modélisation et vérification*.
- Un logiciel gratuit, évolutif et qui sera déposé à l'Agence pour la Protection des Programmes.

Ce projet est actuellement principalement développé par des étudiants de Licence et de Master de l'université de Provence. Une présentation de la version v1.0 développée en janvier et février 2005 par des étudiants de Master Professionnel est décrite très succinctement sur le site <http://www.cmi.univ-mrs.fr/~morin/AMSC>. L'éditeur de l'outil AMSC permet de spécifier des MSC par une interface graphique. Les spécifications produites peuvent être sauvegardées dans un fichier au format xml. L'éditeur de AMSC permet aussi de spécifier des MSC de haut-niveau également sauvegardés dans un format de fichier xml. La vérification de certaines propriétés de ces HMSC s'appuiera sur des algorithmes développés au LIF. Il s'agit notamment d'implémenter l'algorithme de détection de divergence de canaux présenté dans [18]. Ensuite nous programmerons la vérification de l'implémentabilité déterministe sans deadlock selon le critère établi dans [5]. Nous mettrons enfin en oeuvre un algorithme récemment développé au LIF pour la synthèse de systèmes communicants via un dépliage. Cette approche s'appuie sur le premier algorithme polynômial pour la construction d'automates asynchrones [4]. La recherche sur la théorie des MSC étant poursuivie actuellement, le projet AMSC vise évidemment à incorporer nos résultats futurs, notamment nos recherches sur l'implémentabilité faible sans deadlock.

ELSE : un vérificateur de modèles (responsable Peter Niebert)

L'outil ELSE [30] est un vérificateur de modèles qui peut analyser des systèmes dont la description est fournie dans un langage propre d'autres outils : SPIN, IF, et UPPAAL [14, 8, 23]. De la possibilité de traiter des systèmes décrits dans le langage de UPPAAL, on déduit une particularité de ELSE (par rapport à SPIN), la possibilité d'analyser des systèmes avec des contraintes de temps réel. Une deuxième particularité de ELSE est sa conception autour des techniques de réduction d'ordre partiel qui suggèrent comment visiter les états d'un système fortement concurrent de façon efficace. Pour cette raison cet outils se prête bien à une implantation immédiate des algorithmes que nous souhaitons développer dans notre travail théorique. Pour la même raison, ELSE est – en théorie – un produit prometteur pour l'analyse des systèmes fortement concurrent.

Nous disons « en théorie » car les heuristiques pour l'exploration des états, développées au LIF et implantées dans ELSE, ont été testées sur des machines avec des mémoires limitées. Ces limites ont empêché une pleine appréciation des euristiques. Naturellement, nous souhaitons disposer d'une machine performante, pour pouvoir évaluer en profondeur ces euristiques et celles que nous nous proposons de développer.

Comme partie du développement de l'outil ELSE, nous nous proposons aussi de l'intégrer avec l'outil AMSC. Il s'agit par exemple de définir une traduction des langages des MSC vers les structures de

données propres de ELSE, et d'implanter cette traduction.

Références

- [1] Samson Abramsky and Paul-André Melliès. Concurrent games and full completeness. In *14th Symposium on Logic in Computer Science (Trento, 1999)*, pages 431–442. IEEE Computer Soc., Los Alamitos, CA, 1999.
- [2] V. M. Abrusci and P. Ruet. Non-commutative logic I : the multiplicative fragment. *Annals of Pure and Applied Logic*, 101(1) :29–64, 2000.
- [3] A. Arnold and D. Niwinski. *Rudiments of μ -calculus*. Studies in Logic and the Foundations of Mathematics. 146. Amsterdam : Elsevier. xvii, 277 p. Dfl. 86.00 , 2001.
- [4] N. Baudru and R. Morin. Polynomial synthesis of asynchronous automata. Submitted, 35 pages, 2005.
- [5] Nicolas Baudru and Rémi Morin. Safe implementability of regular message sequence chart specifications. In Walter Dosch and Roger Y. Lee, editors, *SNPD*, pages 210–217. ACIS, 2003.
- [6] Stephen L. Bloom and Zoltán Ésik. *Iteration theories*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1993. The equational logic of iterative processes.
- [7] Sébastien Bornot, Rémi Morin, Peter Niebert, and Sarah Zennou. Black box unfolding with local first search. In Joost-Pieter Katoen and Perdita Stevens, editors, *TACAS*, volume 2280 of *Lecture Notes in Computer Science*, pages 386–400. Springer, 2002.
- [8] Marius Bozga, Susanne Graf, and Laurent Mounier. IF-2. 0 : A validation environment for component-based real-time systems. In *Brinksma, Ed (ed.) et al., Computer aided verification. 14th international conference, CAV 2002, Copenhagen, Denmark, July 27-31, 2002. Proceedings. Berlin : Springer. Lect. Notes Comput. Sci. 2404, 343-348 . 2002.*
- [9] Nathalie Caspard, Claude Le Conte de Poly-Barbut, and Michel Morvan. Cayley lattices of finite Coxeter groups are bounded. *Adv. in Appl. Math.*, 33(1) :71–94, 2004.
- [10] Patrick Dehornoy. On completeness of word reversing. *Discrete Math.*, 225(1-3) :93–119, 2000. Formal power series and algebraic combinatorics (Toronto, ON, 1998).
- [11] V. Diekert and G. Rozenberg, editors. *The book of traces*. World Scientific Publishing Co. Inc., River Edge, NJ, 1995.
- [12] Bernhard Ganter and Rudolf Wille. *Formal concept analysis*. Springer-Verlag, Berlin, 1999. Mathematical foundations, Translated from the 1996 German original by Cornelia Franzke.
- [13] Éric Goles, Matthieu Latapy, Clémence Magnien, Michel Morvan, and Ha Duong Phan. Sand-pile models and lattices : a comprehensive survey. *Theoret. Comput. Sci.*, 322(2) :383–407, 2004.
- [14] Jean-Charles Grégoire and Gerard J. Holzmann, editors. *The Spin verification system*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 32, Providence, RI, 1997. American Mathematical Society.
- [15] Michel Habib, Raoul Medina, Lhouari Nourine, and George Steiner. Efficient algorithms on distributive lattices. *Discrete Appl. Math.*, 110(2-3) :169–187, 2001.
- [16] P. Ille and P. Ruet. Cyclic extensions of order varieties. *Under review*, 2004.

- [17] André Joyal. Une théorie combinatoire des séries formelles. *Adv. in Math.*, 42(1) :1–82, 1981.
- [18] Rémi Morin. On regular message sequence chart languages and relationships to mazurkiewicz trace theory. In Furio Honsell and Marino Miculan, editors, *FoSSaCS*, volume 2030 of *Lecture Notes in Computer Science*, pages 332–346. Springer, 2001.
- [19] Rémi Morin. Raffinements au pays des MSC. Transparents, disponibles à <http://www.cmi.univ-mrs.fr/~morin/Talks/labri.pdf>, 2005.
- [20] Peter Niebert, Michaela Huhn, Sarah Zennou, and Denis Lugiez. Local first search—a new paradigm for partial order reductions. In *CONCUR 2001—concurrency theory (Aalborg)*, volume 2154 of *Lecture Notes in Comput. Sci.*, pages 396–410. Springer, Berlin, 2001.
- [21] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part i. *Theor. Comput. Sci.*, 13 :85–108, 1981.
- [22] Vítězslav Novák. Cyclically ordered sets. *Czechoslovak Math. J.*, 32(107)(3) :460–473, 1982.
- [23] Paul Pettersson and Kim G. Larsen. Uppaal2k. *Bulletin of the European Association for Theoretical Computer Science*, 70 :40–44, 2000.
- [24] Vaughan Pratt. Modeling concurrency with partial orders. *Internat. J. Parallel Programming*, 15(1) :33–71, 1986.
- [25] P. Ruet. Non-commutative logic II : sequent calculus and phase semantics. *Mathematical Structures in Computer Science*, 10(2) :277–312, 2000.
- [26] J. J. M. M. Rutten. Universal coalgebra : a theory of systems. *Theoret. Comput. Sci.*, 249(1) :3–80, 2000. *Modern algebra and its applications* (Nashville, TN, 1996).
- [27] Luigi Santocanale. Lattices of paths in higher dimension. Preprint, May 2005.
- [28] Dana S. Scott. Domains for denotational semantics. In *Automata, languages and programming (Aarhus, 1982)*, volume 140 of *Lecture Notes in Comput. Sci.*, pages 577–613. Springer, Berlin, 1982.
- [29] Glynn Winskel. Name generation and linearity. In *LICS 05*, 2005.
- [30] Sarah Zennou, Manuel Yguel, and Peter Niebert. Else : A new symbolic state generator for timed automata. In Kim Guldstrand Larsen and Peter Niebert, editors, *FORMATS*, volume 2791 of *Lecture Notes in Computer Science*, pages 273–280. Springer, 2003.

Méthodologie et objectifs

Notre recherche en équipe se déroulera principalement sous la forme d’un groupe de travail à échéance bihebdomadaire. Dans le cadre du groupe de travail nous inviterons des experts extérieurs à l’équipe, de France et de l’étranger. Les invités auront la possibilité de séjourner à Marseille pendant quelques jours, ce qui nous permettra d’établir des collaborations éventuelles avec eux.

Première année. Pendant une première période, que nous estimons de 8 mois, chaque membre de l’équipe exposera systématiquement ses compétences personnelles et des travaux qu’il juge pertinents avec les thématiques proposées. Après cette période, plus proprement d’étude que de recherche, il sera possible développer des nouvelles connaissances ; à ce fin, il faudra avoir atteint une bonne amalgame entre les différents éléments de l’équipe. Pendant la deuxième période les les objectifs de recherche proposés seront attaqués : on pourra par exemple chercher des caractérisations des

différentes modèles de calcul concurrent dans le langage de la théorie des ensembles ordonnés, ou immédiatement développer des nouveaux algorithmes pour l'exploration des états d'un système concurrent.

Il nous semble d'ailleurs important que les problématiques d'implantations des outils AMSC et ELSE soient bien comprises par toute l'équipe; nous n'excluons pas de dédier du temps à la conception du code.

Deuxième année. Nous réfléchirons dans notre groupe de travail sur les résultats obtenus jusqu'à ce point. Nous essayerons d'assimiler des idées et travaux portant sur la sémantique de la concurrence et sur l'algorithmique que nous ne pouvons pas au présent bien situer par rapport à notre proposition. Par exemple on s'intéressera aux travaux [29, 1] et [15].

L'équipe se confrontera avec des autres équipes françaises qui ont étudié les structures ordonnées avec des motivations différentes. A ce fin, nous nous proposons d'organiser un colloque, de portée nationale, portant sur les applications des structures ordonnées en informatique.

Troisième année. Nous allons employer la majorité de notre temps aux travail de recherche (comme oppose à l'approfondissement de connaissances et à l'assimilation d'autres thématiques). Comme conclusion à notre projet, nous envisageons organiser un colloque de portée international sur les thématiques proposés. Ce colloque nous donnera la possibilité de mettre en valeur les résultats obtenus.