

Développement prouvé d'un algorithme de vérification

Mots-clé : théorie des automates finis, assistant de preuve, programmation fonctionnelle

Laboratoires

LSV
Laboratoire Spécification et Vérification
Ecole Normale Supérieure de Cachan
61 avenue du Président Wilson
94235 Cachan cédex

LORIA
Laboratoire lorrain de recherche
en informatique et ses applications
Campus Scientifique, B.P. 239
54506 Vandœuvre-lès-Nancy cédex

Contexte

La théorie des automates finis joue un rôle éminent en informatique. Il existe en particulier une relation très forte entre les automates finis (opérant sur des objets finis ou infinis) et certaines théories logiques et arithmétiques. Cette relation est à la base d'une approche de la vérification algorithmique des systèmes appelée *model-checking* [1].

Dans cette approche, le système \mathcal{T} et la propriété attendue \mathcal{P} sont représentés par des automates, et la vérification est ramenée à la résolution d'un problème $\mathcal{L}(\mathcal{T}) \subseteq \mathcal{L}(\mathcal{P})$ d'inclusion de langages. Traditionnellement, ce problème est résolu par une réduction au problème $\mathcal{L}(\mathcal{T} \times \mathcal{P}_c) = \emptyset$ du vide, où \mathcal{P}_c désigne l'automate qui accepte le complément du langage de \mathcal{P} et \times le produit synchronisé d'automates. Plusieurs algorithmes efficaces ont été proposés dans la littérature pour décider ce problème ; néanmoins, en pratique les automates sont de grande taille, et le problème de l'explosion combinatoire reste un frein important pour l'application systématique des techniques du *model-checking*.

La technique des antichaînes a été introduite récemment pour s'attaquer à l'explosion combinatoire que l'on rencontre dans les problèmes de la théorie des automates finis. Par exemple, d'excellents résultats ont été obtenus pour le problème d'universalité des automates finis (qui demande si un automate fini donné accepte tous les mots finis) qui a une complexité exponentielle [2]. Cette technique consiste à identifier et exploiter la structure des graphes qu'on manipule. Ainsi, pour le problème d'universalité on utilise la construction des sous-ensembles pour déterminer les automates finis, et on peut se rendre compte que cette construction a des propriétés spécifiques exploitables algorithmiquement. En pratique, cette technique permet d'analyser des automates de 10.000 états contre une centaine d'états par les méthodes classiques.

Toujours dans le contexte de la vérification de systèmes, il est important de s'assurer de la correction de l'implémentation de l'algorithme de vérification afin de garantir la fiabilité du résultat. Ce problème devient non-trivial dès lors que les implémentations utilisent des structures de données compliquées ou mettent en œuvre des optimisations avancées, et des erreurs compromettant la correction ont été trouvées dans plusieurs outils de *model-checking*.

Sujet

Au cours de ce stage on s'intéressera à l'utilisation d'un assistant à la preuve comme Isabelle [3] pour générer une implémentation exécutable et digne de confiance pour l'algorithme basé sur les antichânes qui résout le problème d'universalité pour les automates finis. Dans un premier temps on formalisera les constructions élémentaires nécessaires sur les automates finis dans la logique de l'assistant à la preuve. Ensuite on décrira l'algorithme à travers des définitions qui ressemblent à un programme fonctionnel, et on démontrera la correction de cet algorithme. Isabelle dispose d'un mécanisme de génération de code à partir de telles définitions, et on évaluera l'efficacité du code généré. L'objectif est d'obtenir une implémentation prouvée dont l'efficacité est proche de celle d'une implémentation conventionnelle existante. Des travaux récents [4, 5] nous laissent penser que cet objectif peut être atteint.

Cadre du travail

Le sujet demande des solides connaissances de base en théorie d'automates, ainsi qu'un goût pour les techniques formelles de modélisation et de vérification. Une expérience avec l'utilisation d'un assistant interactif à la preuve (comme Coq, HOL, Isabelle ou PVS) sera appréciée.

Les personnes encadrant ce stage sont :

Laurent Doyen

Tél. : 01 47 40 77 06

doyen@lsv.ens-cachan.fr

<http://www.lsv.ens-cachan.fr/~doyen/>

Stephan Merz

Tél. : 03 54 95 84 78

Stephan.Merz@loria.fr

<http://www.loria.fr/~merz/>

Références

- [1] M. Vardi. *Verification of Concurrent Programs : The Automata-Theoretic Framework*. 2nd Symp. Logic in Computer Science, pp. 167–176. IEEE, 1987
- [2] Martin De Wulf, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. *Antichains : A New Algorithm for Checking Universality of Finite Automata*. 18th Intl. Conf. Computer-Aided Verification (CAV 2006), LNCS 4144, pp. 17–30. Springer, 2006.
- [3] Tobias Nipkow, Lawrence Paulson, and Markus Wenzel. *Isabelle/HOL. A Proof Assistant for Higher-Order Logic*. LNCS 2283. Springer, 2002.
- [4] Alexander Schimpf, Stephan Merz, and Jan-Georg Smaus. *Construction of Büchi Automata for LTL Model Checking Verified in Isabelle/HOL*. 22nd Intl. Conf. Theorem Proving in Higher-Order Logics (TPHOLs 2009), LNCS 5674, pp. 424–439. Springer, 2009.
- [5] Peter Lammich. *Tree Automata*. The Archive of Formal Proofs, <http://afp.sf.net/entries/Tree-Automata.shtml>, 2009.