View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by INRIA a CCSD electronic archive server

Approximations par réécriture pour deux problèmes indécidables

Roméo Courbis, Pierre-Cyrille Heam, Pierre Jourdan, Olga Kouchnarenko

To cite this version:

Roméo Courbis, Pierre-Cyrille Heam, Pierre Jourdan, Olga Kouchnarenko. Approximations par réécriture pour deux problèmes indécidables. AFADL, Jun 2010, Poitiers, France. pp.7. hal-00530341

HAL Id: hal-00530341

https://hal.archives-ouvertes.fr/hal-00530341

Submitted on 28 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Approximations par réécriture pour deux problèmes indécidables

Roméo Courbis*, Pierre-Cyrille Héam*, Pierre Jourdan**, Olga Kouchnarenko*

*INRIA/CASSIS et LIFC/Université de Franche-Comté
rcourbis,pcheam,okouchnarenko@lifc.univ-fcomte.fr
**France Telecom Groupe

Résumé. Pour vérifier des propriétés de sûreté ou de (non-)atteignabilité par model-checking régulier, on se concentre sur une modélisation des configurations accessibles du système par des langages réguliers et des relations d'évolution par des systèmes de réécriture. Le problème d'atteignabilité est indécidable dans de nombreux formalismes, et l'approche générale consiste à étudier et/ou à combiner des cas particuliers. Nous nous intéressons à l'approche par approximation pour semi-décider ce problème de vérification.

Dans ce papier, nous exploitons l'analyse d'atteignabilité pour deux problèmes indécidables pour les machines de Turing : vacuité d'un langage et appartenance d'un mot à un langage. Nous proposons une modélisation et montrons comment les approximations par réécriture permettent de semi-décider ces problèmes. Cette approche a été implantée et expérimentée.

1 Introduction

Concevoir des logiciels sûrs est difficile tant sur le plan pratique (les applications sont de taille importante), que sur le plan théorique (la plupart des problèmes de vérification sont indécidables). Dans ce cadre, il convient alors de faire des compromis, soit en limitant les propriétés vérifiées (model-checking, preuve automatique), soit en ne fournissant pas de garantie mathématique (test), soit en utilisant l'aide d'un expert (preuve).

Dans une démarche de vérification automatique, un moyen d'élargir le cadre possible des applications, consiste non plus à chercher des méthodes exactes mais des méthodes par approximations : l'algorithme de vérification devient alors un semi-algorithme et peut éventuellement répondre qu'il ne sait pas conclure. Dans le cadre du model-checking régulier, on a le problème suivant : soit l'ensemble des configurations accessibles $\mathcal{R}^*(D)$, où $\mathcal{R}^*$ est la clôture transitive et réflexive de la relation $\mathcal{R}$ et D un ensemble de configurations de départs ; soit A un ensemble de configurations d'arrivée dont on souhaite déterminer l'accessibilité. A-t-on $\mathcal{R}^*(D) \cap A = \emptyset$? Si oui, alors aucune configuration de A n'est atteignable, si oui alors certaines configurations de A le sont. Malheureusement, en général, le langage $\mathcal{R}^*(D)$ n'est pas régulier et son calcul peut ne pas terminer Genet (1998), Feuillade et al. (2004). Pour contourner ce problème, la solution proposée est de calculer une sur-approximation, ou une sous-approximation, de $\mathcal{R}^*(D)$ et de l'exploiter pour semi-décider certains problèmes de vérification.

En suivant cette approche, une technique fondée sur des approximations automatiques a été développée au LIFC et appliquée de façon efficace pour la vérification de protocoles de sécurité Boichut (2006). Des travaux récents nous ont permis d'apporter des améliorations théoriques à l'approche Boichut et al. (2008a), Boichut et al. (2008b). Par ailleurs, le nouveau moteur de l'outil présage une plus grande efficacité Balland et al. (2008).

Nous souhaiterions évaluer la technique sur différents problèmes indécidables ou de très forte complexité. Parmi les différents problèmes d'accessibilité indécidables, en relation avec des problématiques de vérification, nous nous intéressons au problème de vacuité du langage d'une machine de Turing et au problème d'appartenance d'un mot au langage d'une machine de Turing (section 2). A l'aide d'une modélisation par systèmes de réécriture et automates d'arbres, nous nous intéressons à l'approche par approximation pour semi-décider ces deux problèmes indécidables pour les machines Turing (section 3). Pour une classe de problèmes particuliers, nous développons une procédure qui transforme automatiquement une instance du problème en un problème d'accessibilité. Cette procédure utilise intensivement un outil de manipulation d'automates d'arbre, nommé Timbuk [1].

2 Deux problèmes sur les machines de Turing

Parmi les différents problèmes d'accessibilité indécidables, en relation avec des problématiques de vérification, nous nous intéressons au problème de vacuité du langage d'une machine de Turing et au problème d'appartenance d'un mot au langage d'une machine de Turing.

Une machine de Turing M, pour un alphabet Σ, définit un langage $\mathcal{L}(M)$ de mots sur Σ^*. Pour déterminer si un mot $w \in \Sigma^*$ est reconnu par M, la machine effectue un calcul depuis un état initial, modélisant w, pour atteindre un état d'acceptation si possible. Si un état d'acceptation est atteint alors $w \in \mathcal{L}(M)$, sinon $w \notin \mathcal{L}(M)$. On notera C_{accept} l'ensemble des états d'acceptation et C_{init} l'ensemble des états initiaux.

Dans le cadre des machines de Turing, nous avons les problèmes indécidables suivants :

Problème d'appartenance (PAML)
Input : M une machine de Turing, et $w \in \Sigma^*$. **Output :** Vrai si $w \in \mathcal{L}(M)$, faux sinon.

Problème de vacuité (PV)
Input : M une machine de Turing. **Output :** Vrai si $\mathcal{L}(M) = \emptyset$, faux sinon.

Dans notre étude en relation avec la problématique de vérification, nous nous intéressons aux machines de Turing déterministes.

3 Procédures de semi-décision pour résoudre PAML et PV

3.1 Problème d'atteignabilité

Le problème d'atteignabilité en réécriture est comme suit, où $\mathcal{R}$ est un système de réécriture et $\mathcal{T}(\mathcal{F})$ un ensemble de termes.

1. http ://www.irisa.fr/lande/genet/timbuk/

Problème d'atteignabilité
Input : $\mathcal{R}$, et deux termes $s, t \in \mathcal{T}(\mathcal{F})$. **Output :** Vrai si $s \rightarrow_{\mathcal{R}}^{*} t$, faux sinon.

Ce problème qui peut être facilement décidé pour les TRS terminant, est indécidable pour les TRS non terminant en général. Cependant, pour certaines classes de TRS ayant des restrictions syntaxiques le problème devient décidable comme on peut le voir dans (Feuillade et al., 2004, section 4). Pour des TRS autres que ceux des classes décidables, on peut prouver que $s \not\rightarrow_{\mathcal{R}}^{*} t$ en utilisant une sur-approximation de $\mathcal{R}^*(E)$ Jacquemard (1996), Feuillade et al. (2004) et en montrant que t n'appartient pas à cette approximation.

3.2 Décision du PV et de PAML par approximation d'accessibilité

Dans cette section on utilisera les notations suivantes : soit M une machine de Turing, soient C_{accept} et C_{init} les ensembles des états d'acceptation et initiaux de M. Soit $\mathcal{R}_M$ le système de réécriture modélisant les comportements de M, et soient $\mathcal{L}^M_{accept}$ et $\mathcal{L}^M_{init}$ des langages régulier d'arbre reconnaissant les termes des ensembles C_{accept} et C_{init}. On définit la fonction $\mathfrak{t}$ qui transforme un mot sur Σ^* en un terme sur $\mathcal{T}(\mathcal{F})$. Plus d'informations sur l'encodage d'une machine de Turing et des états/configurations sont disponibles dans Jourdan (2009).

Décision de PAML Pour décider le problème d'appartenance "$w \in L(M)$?" ($w \in \Sigma^*$), on peut se ramener au problème suivant : pour un état initial $c_{init} \in C_{init}$, est-ce qu'il existe $c_{accept} \in C_{accept}$ tel que c_{init} atteigne c_{accept} par un calcul de M ? Exprimé sous la forme de problème d'accessibilité, on a la proposition suivante :

Proposition 1 *(Décision de l'appartenance par analyse d'accessibilité) Soit le terme* $t_w = \mathfrak{t}(w)$*, on a :* $w \in \mathcal{L}(M) \Leftrightarrow \mathcal{R}^*_M(\{t_w\}) \cap \mathcal{L}^M_{accept} \neq \emptyset$ *et* $w \notin \mathcal{L}(M) \Leftrightarrow \mathcal{R}^*_M(\{t_w\}) \cap \mathcal{L}^M_{accept} = \emptyset$

Décision de PV Décider le problème de vacuité "$\mathcal{L}(M) = \emptyset$?" revient à savoir décider si $\forall c_{init} \in C_{init}$, on n'atteint aucun état d'acceptation. Exprimé sous la forme de problème d'accessibilité, on a la proposition suivante :

Proposition 2 *(Décision du vide par un calcul d'accessibilité) On a :*
$\mathcal{L}(M) = \emptyset \Leftrightarrow \mathcal{R}^*(\mathcal{L}^M_{init}) \cap \mathcal{L}^M_{accept} = \emptyset$ *et* $\mathcal{L}(M) \neq \emptyset \Leftrightarrow \mathcal{R}^*(\mathcal{L}^M_{init}) \cap \mathcal{L}^M_{accept} \neq \emptyset$

Les calculs de $\mathcal{R}^*(\{t_w\})$ et de $\mathcal{R}^*(\mathcal{L}^M_{init})$ ne terminant pas en général, nous nous tournons vers les Théorèmes 1 et 2 de Feuillade et al. (2004) car le TRS $\mathcal{R}_M$ est linéaire. Nous pouvons donc appliquer les algorithmes de Feuillade et al. (2004) pour calculer une sur-approximation, ou sous-approximation, de l'ensemble des termes atteignables par réécriture du TRS $\mathcal{R}^*_M$.

4 Conclusion et perspectives

Nous avons utilisé une modélisation des machines du Turing, sous forme de systèmes de réécriture linéaires et d'automates d'arbre, pour étudier deux problèmes généraux théoriques sur les machines de Turing : le problème de l'appartenance d'un mot au langage et le problème

de vacuité. Nous avons montré l'utilité des techniques d'approximation pour des problèmes indécidables. Nous avons automatisé cette technique en utilisant un outil existant de manipulation d'automates d'arbre, Timbuk. Il serait nécessaire maintenant de confirmer nos résultats par plus d'expérimentations, notamment pour d'autres instances de problèmes moins prévisibles. Par ailleurs, il serait intéressant d'expérimenter la complétion avec une nouvelle version de Timbuk utilisant des équations pour approximer.

Références

Balland, E., Y. Boichut, T. Genet, et P.-E. Moreau (2008). Towards an efficient implementation of tree automata completion. In J. Meseguer et G. Rosu (Eds.), *AMAST*, Volume 5140 of *Lecture Notes in Computer Science*, pp. 67–82. Springer.

Boichut, Y. (2006). *Approximations pour la vérification automatique de protocoles de sécurité*. Thèse de doctorat, Laboratoire Informatique de l'université de Franche-Comté, Université de Franche-Comté, Besançon, France.

Boichut, Y., R. Courbis, P.-C. Héam, et O. Kouchnarenko (2008a). Finer is better : Abstraction refinement for rewriting approximations. In A. Voronkov (Ed.), *RTA*, Volume 5117 of *Lecture Notes in Computer Science*, pp. 48–62. Springer.

Boichut, Y., R. Courbis, P.-C. Héam, et O. Kouchnarenko (2008b). Handling left-quadratic rules when completing tree automata. *Electr. Notes Theor. Comput. Sci 223*, 61–70.

Feuillade, G., T. Genet, et V. V. T. Tong (2004). Reachability analysis over term rewriting systems. *J. Autom. Reasoning 33*(3-4), 341–383.

Genet, T. (1998). Decidable approximations of sets of descendants and sets of normal forms. In T. Nipkow (Ed.), *Rewriting Techniques and Applications, 9th International Conference, RTA-98*, LNCS 1379, Tsukuba, Japan, pp. 151–165. Springer-Verlag.

Jacquemard, F. (1996). Decidable approximations of term rewriting systems. *Lecture Notes in Computer Science 1103*, 362– ? ?

Jourdan, P. (2009). Indécidabilité et approximations. *Mémoire de Master Recherche*, Université de Franche-Comté.

Summary

Safety properties verification–or reachability analysis–can be done using various representations of accessible terms. Unfortunately, the accessibility problem is undecidable in general, and the approaches to circumvent this consist in studying and blending specific cases. We focus on approximation based semi-decision procedures for two undecidable problems over Turing machines: emptiness of a language, and membership of a term to a language. An encoding of these problems, and semi-decision procedures using rewriting approximations are presented, showing how to semi-decide these problems.