Vérification paramétrée de programmes distribués asynchrones

Nathalie Bertrand Thierry Jéron

Inria - CNRS - Irisa, Rennes – Équipe SUMO (http://www.irisa.fr/sumo/)

Keywords: méthodes formelles, vérification paramétrée, programmes distribués asynchrones

Vérification paramétrée. On s'intéresse à la correction de modèles de programmes distribués asynchrones, par exemple de programmes écrits dans le modèle de programmation MPI, standard du monde HPC. En particulier, on se propose d'étudier la faisabilité de techniques de vérification paramétrée. L'objectif de ces dernières est de vérifier des propriétés sur toutes les instances d'un programme, donc sans fixer a priori le nombre de processus participants, contrairement aux techniques habituelles où il faut recommencer la vérification pour chaque instance d'intérêt, pour un nombre de participants fixé.

Dans le cas de processus distribués avec variables partagées, une approche de la vérification paramétrée consiste à transformer le modèle de programme (n processus modélisés par des automates identiques) en un modèle global du système représenté par un automate à compteurs : l'automate a la même structure que l'automate modélisant chaque processus, et est enrichi par des compteurs qui maintiennent dans chaque état de l'automate le nombre de processus qui y résident. La vérification de propriétés de sûreté s'opère alors sur l'automate à compteur. Les problèmes de vérification auxquels on s'intéresse se traduisent en des problèmes d'accessibilité (p. ex, est-ce qu'un état global d'erreur est atteignable ?), et restent décidables.

Objectifs. Une piste du stage consiste à développer ce type d'approches dans le cas où la communication est asynchrone par échange de messages à travers des canaux FIFO. Pour simplifier, on considérera dans un premier temps un modèle de programmes où chaque processus est modélisé par un automate fini, et communique avec ses voisins par envoi-réception de messages sur des canaux FIFO.

La transposition du cadre des variables partagées à celui de la communication asynchrone par canaux FIFO n'est cependant pas immédiate. En effet, les automates à compteurs ne suffisent plus à représenter la communication asynchrone FIFO. L'état global doit aussi représenter l'ensemble des contenus de l'ensemble (non borné) des canaux FIFO, c'est-à-dire, un ensemble de mots sur l'alphabet des messages. La sémantique doit aussi représenter l'évolution des FIFO au cours de l'exécution.

Le stage consistera donc à proposer des modèles adaptés, où on saurait raisonner de façon paramétrée, et où la décidabilité des problèmes d'accessibilité est préservée, au moins pour des sous-classes, i.e. quitte à poser des restrictions sur les comportements des canaux.

Profil recherché. Le stage s'adresse à un·e étudiant·e attiré·e par l'informatique théorique, les méthodes formelles en général, et en particulier les problèmes de modélisation et de vérification de systèmes informatiques.

Références.

1. Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, Josef Widder: Decidability of Parameterized Verification Synthesis Lectures on Distributed Computing Theory, September 2015, Vol. 6, No. 1 , pages 1-170.

2. Javier Esparza, Pierre Ganty, Rupak Majumdar: Parameterized Verification of Asynchronous Shared-Memory Systems. Journal of the ACM 63(1): 10:1-10:48 (2016).

3. Benedikt Bollig, Paul Gastin, Jana Schubert: Parameterized Verification of Communicating Automata under Context Bounds. 8th Workshop on Reachability Problems in Computational Models (RP'14), 2014, Oxford, United Kingdom. pages 45-57.

4. Ganesh Gopalakrishnan, Robert M. Kirby, Stephen F. Siegel, Rajeev Thakur, William Gropp, Ewing L. Lusk, Bronis R. de Supinski, Martin Schulz, Greg Bronevetsky: Formal analysis of MPI-based parallel programs. Communications of the ACM 54(12): 82-91 (2011)