Effective concurrent infinite state systems

Introduction to the scientific context

Multithreading is a central computing principle that allows a single process to launch multiple parallely-running threads. Although the threads are executed independently, they have access to the resources of the process that spawned them. The main benefit of a multithreaded program is that it can run faster on computer systems that possess multiple CPUs, multiple cores on single CPUs, or consist of a network of machines. The reason for this is that threads of the program naturally take advantage of their resources – the aim is that this leads to truly concurrent execution.

Although many different concrete formalisms for representing multithreaded programs exist in the literature [6, 7, 8, 9, 5, 3, 4, 1], they can essentially all be seen as subclasses or slight variants of the following model: *A finite automaton is used to communicate with finitely but potentially unboundedly many components, where each component is either a stack or a counter*. Let us denote this expressive model by *concurrent infinite state systems* for the rest of this research proposal.

The first thing that comes to mind is that that already for a particular subclass of concurrent infinite systems, namely the concurrent behavior of only two counter machines that both test for zero (a.k.a two-counter Minsky machines) the basic question of reachability is undecidable. A second famous subclass of concurrent infinite state systems is the concurrent behavior of a finite number n of counter machines that cannot test for zero (a.k.a n-dimensional vector addition systems with states, n-VASS for short) having a decidable reachability problem [10, 12, 11]. Therefore one carefully needs to restrict concurrent infinite systems in such a way that verification problems such as reachability, model checking or equivalence checking remain decidable.

Positioning of the project

This project is positioned in theoretical computer science, more specifically in the foundations of verification. The research commnunity is logic in computer science and formal languages/automata theory, typically centered around the following tier conferences: CONCUR, FOSSACS, FSTTCS, MFCS, ICALP, LICS, STACS.

The project suits a theoretically-minded student with some taste for theoretical and algorithmic constructions. The internship is an ideal opportunity for starting a PhD thesis (with the possible collaborations with

world-wide leading experts in the area of verification of infinite state systems.

Scientific objectives

This research proposal aims at providing a better foundational understanding of concurrent infinite state systems by analyzing subclasses of infinite state systems. More concretely, for a meaningful subclass $\mathcal{C}$ of concurrent infinite state systems this research proposal aims at providing answers to the following type of question:

> Does a given concurrent infinite state system S behave equivalently to some system S' from the class $\mathcal{C}$?

There are far more questions than answers on our central research question. We therefore focus on ones that we are convinced to be of central importance.

1. Is the language of a given determinstic n-VASS equivalent to a k-VASS for some $k < n$. When $k = 0$ this question is the well-known regularity problem. Decidability of this question is only known for $n \leq 2$. A good starting point is to determine the precise complexity of deciding if the language of a deterministic 2-VASS (whose reachability problem we have recently shown to be PSPACE-complete [2]) is already the language of a 1-VASS.

2. Is the reachability set of a given n-VASS semilinear? For $n \leq 2$ the answer is always yes, but there are 3-VASS for which this is not the case. We are convinced that the understanding of this question is decisive for improving the complexity of reachability of 3-VASS (which is not known to be better-behaved than reachability of general n-VASS). Moreover, when being sure that the reachability set is semilinear, one can apply acceleration techniques for the reachability problem. XXX citation

3. What is the complexity of deciding if the language of a deterministic pushdown automaton is regular? The best-known upper bound of 2EXPTIME dates back to the seventies [13], where the problem is only known to be hard for PTIME. Is it decidable if the language of a deterministic pushdown automaton is in fact the language of a 1-VASS? These are simply stated questions that have not been improved for a long time.

4. What is the maximal number of bisimulation classes of a pushdown automaton that is bisimilar to a finite graph? The understanding of

this question can very likely lead to a first complexity upper bound for bisimulation equivalence of pushdown automata (such an approach was successful for deciding bisimulation equivalence of one-counter automata [pushdown automata working over a single stack symbol]).

Particular conditions

Ideally, the candidate holds a Master degree in Computer Science (with courses in formal verification, theoretical computer science and mathematical structures for Computer Science) or equivalently is graduated from a Computer Science Engineering School with a strong background in theoretical computer science.

The candidate will be supervised by the following two researchers:

- Alain Finkel
  - *Telephone:* Alain Finkel : +33 (0) 1 47 40 75 69
  - *E-mail:* finkel@lsv.ens-cachan.fr
  - Laboratoire Spécification et Vérification
  - Ecole Normale Supérieure de Cachan 61, avenue du Président Wilson, 94235 Cachan CEDEX
- Stefan Göller
  - *Telephone:* Stefan Göller: +33 (0)1 47 40 74 07
  - *E-mail:* goeller@lsv.ens-cachan.fr
  - Laboratoire Spécification et Vérification
  - Ecole Normale Supérieure de Cachan 61, avenue du Président Wilson, 94235 Cachan CEDEX

This project is integrated into the research project VERICONISS (Verification of Concurrent Infinite State Systems) lead by Stefan Göller and carried out at Laboratoire Spécification et Vérificationin Cachan. We expert fruitful collaboration with the researchers Christoph Haase and Piotr Hofman that are currently employed by VERICONISS.

Methods and means

The student will begin to read and understand the main existing results on the subject. In particular, he/she will have to understand the very recent results on the decidability-complexity of reachability for VASS, 1-VASS, 2-VASS and 1-VASS with one stack. The student will attend different working

groups at LSV. We will propose to the student to participate to different meetings, schools as Highlights 2015, Workshop on parameterized verification at Concur, Young Researchers' Conference "Frontiers of Formal Methods",... Some particularly interesting algorithms could be implemented and tested.

Working environment

This internship will be supervised in the lab Laboratoire de Spécification et Vérification at the Ecole Normale Supérieure de Cachan.

References

[1] Mohamed Faouzi Atig, Ahmed Bouajjani, Michael Emmi, and Akash Lal. Detecting Fair Non-termination in Multithreaded Programs. In *Proc. of CAV*, volume 7358 of *Lecture Notes in Computer Science*, pages 210–226. Springer, 2012.

[2] Michael Blondin, Alain Finkel, Stefan Göller, Christoph Haase, and Pierre McKenzie. Reachability in Two-Dimensional Vector Addition Systems with States is PSPACE-complete. *CoRR*, abs/1412.4259, 2014.

[3] Ahmed Bouajjani and Michael Emmi. Analysis of recursively parallel programs. In *Proc. of POPL*, pages 203–214. ACM, 2012.

[4] Ahmed Bouajjani and Michael Emmi. Bounded Phase Analysis of Message-Passing Programs. In *Proc. of TACAS*, volume 7214 of *Lecture Notes in Computer Science*, pages 451–465. Springer, 2012.

[5] Ahmed Bouajjani, Michael Emmi, Constantin Enea, and Jad Hamza. Verifying Concurrent Programs against Sequential Specifications. In *Proc. of ESOP*, volume 7792 of *Lecture Notes in Computer Science*, pages 290–309. Springer, 2013.

[6] Ahmed Bouajjani, Michael Emmi, and Gennaro Parlato. On Sequentializing Concurrent Programs. In *Proc. of SAS*, volume 6887 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2011.

[7] Javier Esparza and Pierre Ganty. Complexity of pattern-based verification for multithreaded programs. In *Proc. of POPL*, pages 499–510. ACM, 2011.

[8] Javier Esparza, Pierre Ganty, and Rupak Majumdar. A Perfect Model for Bounded Verification. In *Proc. of LICS*, pages 285–294, 2012.

[9] Javier Esparza, Pierre Ganty, and Rupak Majumdar. Parameterized Verification of Asynchronous Shared-Memory Systems. In *Proc. of CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 124–140. Springer, 2013.

[10] S. R. Kosaraju. Decidability of reachability in vector addition systems. In *14th Annual Symposium on Theory of Computing*, pages 267–281. ACM Press, 1982.

[11] Jérôme Leroux. The General Vector Addition System Reachability Problem by Presburger Inductive Invariants. *Logical Methods in Computer Science*, 6(3), 2010.

[12] E. W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal on Computing*, 13:441–460, 1984.

[13] Leslie G. Valiant. Regularity and related problems for deterministic pushdown automata. *J. ACM*, 22(1):1–10, 1975.