Towards model-checking pointer systems*

Alain Finkel[1], Étienne Lozes[1], and Arnaud Sangnier[1,2]

[1] LSV, ENS Cachan & CNRS UMR 8643
61, av. Pdt Wilson 94235 Cachan Cedex, France
{finkel|lozes|sangnier}@lsv.ens-cachan.fr
[2] EDF R&D

The aim of this work is to investigate the possibility of using a counter machines model-checker, FAST [2] for instance, for programs working over singly-linked lists. We define a CTL* logic which may express quantitative properties and shape properties on the memory managed by pointer systems. For this temporal logic, we show that the model-checking problem reduces to the one for counter systems developed in [5], provided an adequate translation from pointers to counters is given. This result serves as a foundation for a two-steps analysis of pointer systems that consists in translating them into counter systems and then, with the help of FAST, to verify safety properties. It remains to provide a good translation of pointer systems into counter systems.

We obtain three categories of results concerning the translations:

First, we show that, from the experimental point of view, the translation defined in [3] allows us to verify all well-known singly-linked lists case studies, and also some new ones that are not immediately verifiable by the other tools and methodologies. Most of the case studies yield flat counter systems for which we know by advance that FAST will terminate.

Second, we propose a new translation of pointer systems into counter systems and we characterize several classes of pointer systems without destructive update for which our analysis terminates. We prove the decidability of the CTL* model-checking for flat pointer systems without destructive update and with an acyclic initial configuration; this extends the decidability results of [4], that were restricted to a special form of safety. Similarly, we show the decidability of safety for flat pointer systems without destructive update and without alias test.

Finally, we explore the limits of our analysis, and of the decidability of CTL* model-checking for classes of flat pointer systems. Indeed, we show that the model-checking of flat pointer systems without destructive update and alias test is undecidable if cyclic initial configurations are allowed. Conversely, we also prove that the safety problem becomes undecidable for flat pointer systems that keep their memory configurations acyclic, but can perform destructive updates, which answers some open problems asked by [4].

Details about this work can be found in [1].

* This work has been partially supported by contract 4300038040 between EDF R&D/LSV and by the AVERILES project.

References

1. Towards model-checking pointer systems (long version). http://www.lsv.ens-cachan.fr/∼sangnier/FLS07.pdf .
2. S. Bardin, A. Finkel, J. Leroux, and L. Petrucci. Fast: Fast acceleration of symbolic transition systems. In *CAV'03*, volume 2725 of *LNCS*, pages 118–121. Springer, 2003.
3. S. Bardin, A. Finkel, É. Lozes, and A. Sangnier. From pointer systems to counter systems using shape analysis. In *AVIS'06*, 2006.
4. M. Bozga and R. Iosif. On flat programs with lists. In *VMCAI 2007*, volume 4349 of *LNCS*, pages 122–136. Springer, 2007.
5. S. Demri, A. Finkel, V. Goranko, and G. van Drimmelen. Towards a model-checker for counter systems. In *ATVA'06*, volume 4218 of *LNCS*, pages 493–507. Springer, 2006.