Towards Formal Verification of Distributed Algorithms

Benedikt Bollig
LSV, ENS Cachan, CNRS & Inria
Email: bollig@lsv.ens-cachan.fr

Abstract

Model checking is an automatic verification technique, which provides an answer to the question whether a program, given as a state-transition system, satisfies its specification, given in terms of a temporal-logic formula. Model checking is very well studied as far as boolean finite-state programs are concerned [CGP99].

To take into account several sources of infinity such as an unknown number of processes or infinite data structures, the classical setting has been extended in several orthogonal directions. In the context of concurrent programs, one may ask whether a specification is satisfied independently of the number of participating processes. This question is referred to as *parameterized verification* (see [Esp14] for an overview). Second, a system may have to cope with variables ranging over an infinite domain such as the natural numbers or finite strings. Depending on the operations that are allowed on this domain, system executions can then be described as words over an infinite alphabet, possibly equipped with one or several binary relations such as equality or a total order. Those words are usually referred to as *data words* [BDM$^+$11], [BMSS09]. Many models and results from both areas, parameterized verification and data words, smoothly extend the classical finite-state approach and, in particular, provide decidable instances of the model-checking problem.

Our concern in this talk will be *distributed algorithms*, where an unknown number of (identical) processes cooperate to achieve a common goal. However, assuming perfectly identical processes, even simple tasks such as electing a leader cannot always be accomplished. One may, therefore, assume that every process is equipped with a unique process identifier from an unbounded domain, and that identifiers can be compared with one another wrt. a total order. Thus, when modeling distributed algorithms, one has to cope with both sources of infinity mentioned above: the number of processes and infinite data. This may be one reason why there have been only a few approaches to the formal verification of distributed algorithms [KVW12].

In this talk, we survey recent developments in the areas of parameterized verification and data words, and we demonstrate how they can be exploited towards a framework for the formal verification of distributed algorithms [ABG15].

References

[ABG15] C. Aiswarya, B. Bollig, and P. Gastin. An automata-theoretic approach to the verification of distributed algorithms. In *CONCUR'15*, Leibniz International Proceedings in Informatics. Leibniz-Zentrum für Informatik, 2015.

[BDM$^+$11] M. Bojanczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data words. *ACM Trans. Comput. Log.*, 12(4):27, 2011.

[BMSS09] M. Bojańczyk, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data trees and applications to XML reasoning. *Journal of the ACM*, 56(3), 2009.

[CGP99] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 1999.

[Esp14] J. Esparza. Keeping a crowd safe: On the complexity of parameterized verification. In *STACS'14*, volume 25 of *Leibniz International Proceedings in Informatics*, pages 1–10. Leibniz-Zentrum für Informatik, 2014.

[KVW12] I. Konnov, H. Veith, and J. Widder. Who is afraid of model checking distributed algorithms?, 2012.

1530-1311/15 $31.00 © 2015 IEEE
DOI 10.1109/TIME.2015.23