Automatic Synthesis of Distributed Systems

Alin Ştefănescu[1]

LFCS, Division of Informatics, University of Edinburgh,
JCMB, King's Buildings, Edinburgh EH9 3JZ

Abstract

The design of correct distributed systems is a difficult and error-prone task. This is due to the multitude of possible interactions between the concurrent components of the system. The *synthesis* approach directly constructs a *correct* implementation from a given specification, if possible. A complementary approach is *verification* where a given model is tested against a specification. Automatic synthesis has been less successful than verification so far, despite the fact that the synthesized model is correct 'ab initio' (so no need for verification) and that both approaches have similar computational complexity (in the size of specification). A possible explanation is that both approaches were mostly studied using temporal logics like LTL and CTL for the specification (see for example the seminal paper [EC82]). A shortcoming of these logics is the fact that they cannot express properties about casual independences between different actions of the system, and so these properties cannot play a role in the synthesis procedure.

Our research aims towards a new method of synthesis for distributed systems using Mazurkiewicz traces for specification and asynchronous automata for models. Mazurkiewicz trace languages (see their theory in [DR95]) are languages closed under an explicit independence relation between actions and therefore they are suitable to describe concurrent behaviour. Asynchronous automata were proposed by Zielonka [Zie87] as a natural generalization of finite state automata modeling concurrent systems and, loosely speaking, they are sequential automata communicating through rendez-vous. Zielonka solved a long-standing open problem: the construction of an asynchronous automaton able to accept a given *regular* trace language according to a distribution pattern. The procedure is computationally involved and there were attempts to simplify it, but state space explosion is still a stumbling block.

The main objectives of this work are: (a) to develop a specification language based on a distributed version of temporal logic on traces [TH98] that is able to express properties about the independence of actions; (b) to design a synthesis procedure based on improvements and heuristics of the algorithms for asynchronous automata; (c) to implement the new procedure efficiently (and so to turn the theory into a reliable tool that can be used in practice); (d) to apply it to case studies in areas like: small distributed algorithms (e.g. mutual exclusion, communication protocols) and asynchronous circuit design. The idea used for the core of synthesis procedure is that of *unfoldings*, a successful technique based on branching time partial order semantics [ERV96]. Promising preliminary results were obtained: we were able to automatically synthesize mutual exclusion algorithms from regular trace specifications.

References

[DR95] V. Diekert and G. Rozenberg (Eds.). *The Book of Traces*. World Scientific, 1995.

[EC82] E. A. Emerson and E. M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming* 2 (1982) 241-266.

[ERV96] J. Esparza, S. Römer, and W. Vogler. An improvement of McMillan's unfolding algorithm. In *Proc. TACAS'96*, LNCS 1055 (1996) 87-106.

[TH98] P.S. Thiagarajan and J.G. Henriksen. Distributed versions of linear time temporal logic: A trace perspective. In LNCS 1491 (1998) 643-681.

[Zie87] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. Inform. Théor. Appl.* 21 (1987) 99-135.

[1]this PhD research is carried out under the supervision of Professor Javier Esparza and is supported by an EPSRC UK Grant