A game approach to determinize timed automata

Nathalie Bertrand[1], Amélie Stainer[1], Thierry Jéron[1], Moez Krichen[2]

[1] INRIA Rennes - Bretagne Atlantique, Rennes, France
[2] Institute of Computer Science and Multimedia, Sfax, Tunisia

Determinization is a key issue for several problems such as implementability, diagnosis or test generation, where the underlying analyses depend on the observed behaviors. In the context of timed automata (TA), determinization is problematic for two reasons. First, determinizable timed automata form a strict subclass of timed automata [1]. Second, the problem of the determinizability of a timed automaton, is undecidable [4]. Therefore, in order to determinize timed automata, two alternatives have been investigated: either restricting to determinizable classes or choosing to ensure termination for all TA by allowing over-approximations. For the first approach, several classes of determinizable TA have been identified, such as strongly non-Zeno TA, event-clock TA, or TA with integer resets. A recent paper [2] proposes a procedure which does not terminate in general, but allows one to determinize TA in a class covering all the aforementioned determinizable classes. To our knowledge, the second approach has only been investigated in [5] where an algorithm that produces a deterministic over-approximation is proposed.

We propose a method combining techniques from [2] and [5], despite their notable differencies, and improving those two approaches. Our method is inspired by a game approach to decide the diagnosability of TA [3]. The resulting deterministic TA is given fixed resources (number of clocks and maximal constant) in order to simulate the original TA by a coding of relations between new clocks and original ones. The core principle is the construction of a finite turn-based safety game between two players, Spoiler and Determinizator, where Spoiler chooses an action and the region of its occurrence, while Determinizator chooses which set of clocks to reset. Our main result states that any winning strategy for Determinizator yields a deterministic timed automaton accepting exactly the same timed language as the initial automaton, while any strategy produces a deterministic over-approximation. Our approach is more general than the procedure of [2], thus allowing one to enlarge the set of timed automata that can be automatically determinized, thanks to an increased expressive power in the coding of relations between new and original clocks, and robustness to some language inclusions. It is also more precise than the algorithm of [5] in several respects: an adaptative and timed resetting policy, governed by a strategy, compared to a fixed untimed one and a more precise update of the relations between clocks, even for a fixed policy, allow our method to be exact on a larger class of TA. These observations illustrate the benefits of our approach compared to existing work.

References

1. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
2. C. Baier, N. Bertrand, P. Bouyer, and T. Brihaye. When are timed automata determinizable? In *Proc. of ICALP'09*, volume 5556 of *LNCS*, pages 43–54. Springer, 2009.
3. P. Bouyer, F. Chevalier, and D. D'Souza. Fault diagnosis using timed automata. In *Proc. of FOSSACS'05*, volume 3441 of *LNCS*, pages 219–233. Springer, 2005.
4. O. Finkel. Undecidable problems about timed automata. In *Proc. of FORMATS'06*, volume 4202 of *LNCS*, pages 187–199. Springer, 2006.
5. M. Krichen and S. Tripakis. Conformance testing for real-time systems. *Formal Methods in System Design*, 34(3):238–304, 2009.