Simplification of finite automata (ENS Project Proposal)

Lorenzo Clemente, University of Warsaw
clementelorenzo@gmail.com

November 14, 2016

Scope Nondeterministic finite automata (NFA) [7] can be seen as a data structure to represent a (possibly infinite) set of words. In this view, they can be seen as a basic component used in solving problems in different domains, e.g., decision procedures for logical theories (c.f. the MONA tool [4, 5]), program analysis [6], model checking [10], and various reachability problems. NFAs enjoy closure under boolean operations, but while union and intersection are relatively cheap to perform, complementation is exponential in the worst case. Classic decision problems on NFA are emptiness, universality, inclusion, and equivalence; however, they do not have the same complexity, and while emptiness LOGSPACE-complete, the others are already PSPACE-complete. Since procedures manipulating and testing automata might perform many boolean operations on them and expensive tests, it is of crucial importance to keep the size of intermediate automata as small as possible.

Simplification of automata An overview of simple yet effective simplification techniques can be found in [9], where they have been applied to automata over *infinite words* (i.e., the so called Buechi automata), which are of primary interest in the area of verification. The main theme is that exact minimization is not practical (yet) for automata with $>$20 states, and thus approximate minimization techniques based on removing *dead states*, *quotienting* (i.e., identifying equivalent states), and *transition pruning* (i.e., removing redundant transitions) are very fast, even on large automata, and they often lead to drammatic reduction in size.

Directions In this project we are interested in extending simplification techniques for finite automata. These techniques have been studied for automata over finite and infinite words [9] and finite trees [2] with very good practical results. It is reasonable that they extend to other fundamental computation models such as

1. Finite automata over infinite trees, such as Buechi and parity tree automata [1].
2. Register automata, which are used to model programs with infinite data [8].
3. Timed automata, which are used to model time aspects in computations [3].
4. Other models of computations, such as one-counter automata without zero test, and possibly more.

This is a flexible project and other classes of automata can be considered. The scope is both theoretical and practical. From a theoretical point of view, efficient (i.e, PTIME) simplification techniques should be defined, which might require to overcome computational or complexity bottlenecks on the models above (i.e., undecidability, or high complexity). From a practical point of view, experimental evaluation on some cases of interest should validate the usefulness of the proposed techniques w.r.t. both efficiency (they should be fast) and quality of reduction (they should produce small automata).

References

[1] AAVV. *Automata Logics, and Infinite Games - A Guide to Current Research.* Springer, 2002.

[2] Ricardo Almeida, Lukáš Holík, and Richard Mayr. Reduction of nondeterministic tree automata. In Marsha Chechik and Jean-François Raskin, editors, *In Proc. of TACAS'16*, pages 717–735, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[3] Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126:183–235, April 1994.

[4] Jacob Elgaard, Nils Klarlund, and Anders Møller. *MONA 1.x: New techniques for WS1S and WS2S*, pages 516–520. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[5] Tomas Fiedor, Lukas Holik, Petr Janku, Ondrej Lengal, and Tomas Vojnar. Lazy automata techniques for ws1s. Technical report, University of Brno, 2016.

[6] Seth Fogarty and Moshe Y Vardi. Buchi complementation and size-change termination. *Logical Methods in Computer Science*, pages 1–27, April 2010.

[7] John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 2000.

[8] Michael Kaminski and Nissim Francez. Finite-memory automata. *Theoretical Computer Science*, 134(2):329–363, 1994.

[9] Richard Mayr and Lorenzo Clemente. Advanced automata minimization. In *Proc. of POPL'13*, pages 63–74, New York, NY, USA, 2013. ACM.

[10] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification (preliminary report). In *LICS*, pages 332–344, 1986.