Automata on infinite words and Temporal Logic Normal Forms.

Alexander Bolotov
Department of Computing and Mathematics
Manchester Metropolitan University, Manchester M1 5GD, UK.
A.Bolotov@doc.mmu.ac.uk, www.doc.mmu.ac.uk/STAFF/A.Bolotov

We consider the relationship between non-deterministic automata on infinite words [1, 12] and alternating automata [10, 1, 12] and a specific logical formulation based on a normal form for temporal logic formulae, called $\mathrm{SNF}_{\mathrm{PLTL}}$ [6]. While this normal form was developed for use with clausal resolution in temporal logics [5, 8], we here show how it can represent, *syntactically*, these types of automata in a high-level way [2].

The general problem structure that we are trying to solve is the analisys of a system specification followed by some (formal) verification of its properties.

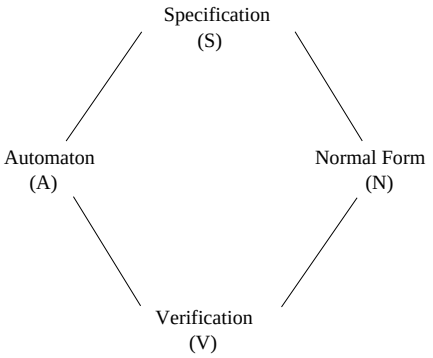

Figure 1: Specification-Verification Problem

If the specification (S) (see Fgure 1) is given in a high-level language, for example a logic, then the translation from (S) to an automaton (A) is exponential [12] in case of non-deterministic automata, for example, Büchi automaton. In contrast, the step from (A) to its verification (V) here is usually polynomial [4, 12], comprising an automaton emptiness check. Thus, if a model is given, we can see here the attraction of a *model-checking* approach. The situation is different when we carry out translation into alternating automata [1, 12], where, translation (S)$\rightarrow$(A) is polynomial while checking non-emptiness is hard. No direct methods of checking non-emptiness of alternating automata is known. The usual way is to simulate an alternating automaton by a standard non-deterministic automaton and then apply the emptiness check to the former [1, 12, 11].

If a specification (S) is given in a high-level language, for example in some logical language, then, aiming to apply as a verification (V) some form of efficient deduction, for example resolution, we carry out the translation from (S) to the Normal Form (N). Here the complexity is polynomial or often linear. In contrast, verification of formulae in the Normal Form is usually exponential since it involves

some form of proof (in our case, clausal resolution). However, we are often able to use either improved proof strategies [7] or restricted forms of the normal form [3] in order to improve the practical efficiency of such proof.

One particular concern is the relationship between the Normal Form (N) given as normal form for PLTL ($\mathrm{SNF}_{\mathrm{PLTL}}$) and the Automaton (A) in the diagram above. We first show [2] that $\mathrm{SNF}_{\mathrm{PLTL}}$ can represent a specific type of non-deterministic automata, Büchi automata. In translating a problem specification into our normal form, we actually derive clauses within a fragment of quantified propositional linear temporal logic (QPLTL) [9, 13] such that formulae within $\mathrm{SNF}_{\mathrm{PLTL}}$ are existentially quantified and then effectively skolemize the normal form producing temporal formulae without any quantification.

Having established this relationship between $\mathrm{SNF}_{\mathrm{PLTL}}$ and Büchi automata, we will be able to represent problem specification directly as a set of formulae in Normal Form and apply resolution based verification technique to the latter. Also, we believe that varying the formulation of acceptance conditions in our syntactic representation of Büchi automata, our approach allows us to specify other types of ω-automata such as Rabin or Street automata. Finally, note that structurally $\mathrm{SNF}_{\mathrm{PLTL}}$ is similar to alternating automata. Thus, if we can show that $\mathrm{SNF}_{\mathrm{PLTL}}$ can represent alternating automata, then clausal resolution method applicable to a set of $\mathrm{SNF}_{\mathrm{PLTL}}$ clauses can be considered as another non-direct method of checking emptiness for alternating automata. This the subject of the ongoing work.

References

[1] O. Bernholtz, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. In *Computer Aided Verification, Proc. 6th Int. Workshop*, volume 818 of *Lecture Notes in Computer Science*, pages 142–155, Stanford, California, June 1994. Springer-Verlag.

[2] A. Bolotov, M. Fisher, and C. Dixon. On the relationship between w-automata and temporal logic normal forms. In *To be published in Proc. of Advances in Modal Logic/International Conference on Temporal Logic 2000*, 2000.

[3] C. Dixon, M. Fisher, and M. Reynolds. Execution and Proof in Horn-Clause Temporal Logic. In *Proceedings the Second International Conference on Temporal Logic (ICTL)*, Manchester, July 1997. Kluwer.

[4] E. A. Emerson. Automated reasoning about reactive systems. In *Logics for Concurrency: Structures Versus Automata, Proc. of International Workshop*, volume 1043 of Lecture Notes in Computer Science. Springer-Verlag, 1996.

[5] M. Fisher. A Resolution Method for Temporal Logic. In *Proc. of the XII International Joint Conference on Artificial Intelligence (IJCAI)*, 1991.

[6] M. Fisher. A normal form for temporal logic and its application in theorem-proving and execution. *Journal of Logic and Computation*, 7(4), 1997.

[7] M. Fisher and C.Dixon. Guiding Clausal Temporal Resolution. In *Proceedings the Second International Conference on Temporal Logic (ICTL)*, Manchester, July 1997. Kluwer.

[8] M. Fisher, C. Dixon, and M. Peim. Clausal Temporal Resolution. To appear in ACM Transactions on Computational Logic (TOCL), 2000.

[9] Y. Kesten and A.Pnueli. A complete deductive system for QPTL. In *Proceedings of the 10th Annual IEEE Symposium of Logic in Computer Science*, 1995.

[10] D. E. Muller, A. Saoudi, and P. Schupp. Alternating automata, the weak monadic theory of the tree and its complexity.

[11] D. E. Muller and P. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of theorems of Rabin and McNaughton.

[12] M. Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for Concurrency: Structures Versus Automata, Proc. of International Workshop*, volume 1043 of Lecture Notes in Computer Science. Springer-Verlag, 1996.

[13] P. Wolper. Temporal logic can be more expressive. *Information and Control*, 56(1–2):72–99, 1983.