Büchi Complementation: A Forty-Year Saga

Moshe Y. Vardi*
Rice University

March 23, 2006

The complementation problem for nondeterministic word automata has numerous applications in formal verification. In order to check that the language of an automaton $\mathcal{A}_1$ is contained in the language of a second automaton $\mathcal{A}_2$, one checks that the intersection of $\mathcal{A}_1$ with an automaton that complements $\mathcal{A}_2$ is empty. Many problems in verification and design are reduced to language containment. In model checking, the automaton $\mathcal{A}_1$ corresponds to the system, and the automaton $\mathcal{A}_2$ corresponds to the property we wish to verify [13, 22]. While it is easy to complement properties given in terms of formulas in temporal logic, complementation of properties given in terms of automata is not simple. Indeed, a word w is rejected by a nondeterministic automaton $\mathcal{A}$ if all the runs of $\mathcal{A}$ on w rejects the word. Thus, the complementary automaton has to consider all possible runs, and complementation has the flavor of determinization.

For automata on finite words, determinization, and hence also complementation, is done via the subset construction [17]. Accordingly, if we start with a nondeterministic automaton with n states, the complementary automaton may have 2^n states. The exponential blow-up that is caused by the subset construction is justified by a tight lower bound: it is proved in [19] that for every $n > 1$, there exists a language L_n that is recognized by a nondeterministic automaton with n states, yet a nondeterministic automaton for the complement of L_n has at least 2^n states (see also [2]).

For Büchi automata on infinite words, which are required for the modeling of liveness properties, optimal complementation constructions are quite complicated, as the subset construction is not sufficient (but see erroneous claim in [16]). Due to the lack of a simple complementation construction, the user is typically required to specify the property by a deterministic Büchi automaton [13] (it is easy to complement a deterministic Büchi automaton), or to supply the automaton for the negation of the property [9]. Similarly, specification formalisms like ETL [23], which have automata within the logic, involve complementation of automata, and the difficulty of complementing Büchi automata is an obstacle to practical use [1]. In fact, even when the properties are specified in LTL, complementation is useful: the translators from LTL into automata have reached a remarkable level of sophistication (c.f., [4, 21, 6, 7]). Even though complementation of the automata is not explicitly

*Address: Department of Computer Science, Rice University, Houston, TX 77251-1892, U.S.A., Email: vardi@cs.rice.edu.

Supported in part by NSF grants CCR-9988322, CCR-0124077, CCR-0311326, IIS-9908435, IIS-9978135, EIA-0086264, and ANI-0216467, by BSF grant 9800096, by Texas ATP grant 003604-0058-2003, and by a grant from the Intel Corporation.

required, the translations are so involved that it is useful to checks their correctness, which involves complementation[1]. Complementation is interesting in practice also because it enables refinement and optimization techniques that are based on language containment rather than simulation[2]. Thus, an effective algorithm for the complementation of Büchi automata would be of significant practical value.

Efforts to develop simple complementation constructions for nondeterministic automata started early in the 60s, motivated by decision problems of second-order logics. Büchi suggested a complementation construction for nondeterministic Büchi automata that involved a complicated combinatorial argument and a doubly-exponential blow-up in the state space [3]. Thus, complementing an automaton with n states resulted in an automaton with $2^{2^{O(n)}}$ states. In [20], Sistla et al. suggested an improved implementation of Büchi's construction, with only $2^{O(n^2)}$ states, which is still, however, not optimal. Only in [18], Safra introduced a determinization construction, which also enabled a $2^{O(n \log n)}$ complementation construction, matching a lower bound described by Michel [15] (cf. [14]). Thus, from a theoretical point of view, some considered the problem solved since 1988.

A careful analysis, however, of the exact blow-up in Safra's and Michel's bounds reveals an exponential gap in the constants hiding in the $O()$ notations: while the upper bound on the number of states in the complementary automaton constructed by Safra is n^{2n}, Michel's lower bound involves only an $n!$ blow up, which is roughly $(n/e)^n$. The exponential gap exists also in more recent complementation constructions. In particular, the upper bound on the number of states in the complementation construction in [12], which avoids determinization, is $(6n)^n$. This is in contrast with the case of automata on finite words, where, as mentioned above, the upper and lower bounds coincide.

This compelxity gap motivated recent effort to close it. An improved complementation construction for nondeterministic Büchi automata is described in [5]. The construction is based on new observations on runs of nondeterministic Büchi automata: a run of a nondeterministic Büchi automaton $\mathcal{A}$ is accepting if it visits the set α of accepting states infinitely often. Accordingly, $\mathcal{A}$ rejects a word w if every run of $\mathcal{A}$ visits α only finitely often. The runs of $\mathcal{A}$ can be arranged in a DAG (directed acyclic graph). It is shown in [12] that $\mathcal{A}$ rejects w iff it is possible to label the vertices of the DAG by ranks in $0, \ldots, 2n$ so that some local conditions on the ranks of vertices and their successors are met. Intuitively, as in the progress measure of [11], the ranks measure the distance to a position from which no states in α are visited. It is shown in [5] that the ranks that label vertices of the same level in the DAG have an additional property: starting from some limit level $l_{lim} \geq 0$, if a vertex in level $l \geq l_{lim}$ is labeled by an odd rank j, then all the odd ranks in $1, \ldots, j$ label vertices in level l. It follows that the complementary automaton, which considers all the possible *level rankings* (i.e., ranks that vertices of some level in the DAG are labeled with), may restrict attention to a special class of level rankings. Using some estimates on the asymptotics of Stirling numbers of the Second

[1]For an LTL formula ψ, one typically checks that both the intersection of $\mathcal{A}_\psi$ with $\mathcal{A}_{\neg\psi}$ and the intersection of their complementary automata are empty.

[2]Since complementation of Büchi automata is complicated, current research is focused on ways in which fair simulation can approximate language containment [8], and ways in which the complementation construction can be circumvented by manually bridging the gap between fair simulation and language containment [10].

Kind it is shown in [5] that the complementary automaton has at most $(0.97n)^n$ states. A recent work by Yang [24] improves Michel's lower bound to $(0.76n)^n$, narrowing the gap considerably, but still leaving it exponentially wide.

References

[1] R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singerman, A. Tiemeyer, M.Y. Vardi, and Y. Zbar. The ForSpec temporal logic: A new temporal property-specification logic. In *Proc. 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 2280 of *Lecture Notes in Computer Science*, pages 296–211, Grenoble, France, April 2002. Springer-Verlag.

[2] J.C. Birget. Partial orders on words, minimal elements of regular languages, and state complexity. *Theoretical Computer Science*, 119:267–291, 1993.

[3] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. International Congress on Logic, Method, and Philosophy of Science. 1960*, pages 1–12, Stanford, 1962. Stanford University Press.

[4] N. Daniele, F. Guinchiglia, and M.Y. Vardi. Improved automata generation for linear temporal logic. In *Computer Aided Verification, Proc. 11th International Conference*, volume 1633 of *Lecture Notes in Computer Science*, pages 249–260. Springer-Verlag, 1999.

[5] E. Friedgut, O. Kupferman, and M.Y. Vardi. Büchi complementation made tighter. In *2nd International Symposium on Automated Technology for Verification and Analysis*, volume 3299 of *Lecture Notes in Computer Science*, pages 64–78. Springer-Verlag, 2004.

[6] P. Gastin and D. Oddoux. Fast LTL to büchi automata translation. In *Computer Aided Verification, Proc. 13th International Conference*, volume 2102 of *Lecture Notes in Computer Science*, pages 53–65. Springer-Verlag, 2001.

[7] S. Gurumurthy, R. Bloem, and F. Somenzi. Fair simulation minimization. In *Computer Aided Verification, Proc. 14th International Conference*, volume 2404 of *Lecture Notes in Computer Science*, pages 610–623. Springer-Verlag, 2002.

[8] T.A. Henzinger, O. Kupferman, and S. Rajamani. Fair simulation. *Information and Computation*, 173(1):64–81, 2002.

[9] G.J. Holzmann. The model checker SPIN. *IEEE Trans. on Software Engineering*, 23(5):279–295, May 1997. Special issue on Formal Methods in Software Practice.

[10] Y. Kesten, N. Piterman, and A. Pnueli. Bridging the gap between fair simulation and trace containment. In *Computer Aided Verification, Proc. 15th International Conference*, volume 2725 of *Lecture Notes in Computer Science*, pages 381–393. Springer-Verlag, 2003.

[11] N. Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proc. 32nd IEEE Symp. on Foundations of Computer Science*, pages 358–367, San Juan, October 1991.

[12] O. Kupferman and M.Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. on Computational Logic*, 2(2):408–429, July 2001.

[13] R.P. Kurshan. *Computer Aided Verification of Coordinating Processes*. Princeton Univ. Press, 1994.

[14] C. Löding. Optimal bounds for the transformation of omega-automata. In *Proc. 19th Conference on the Foundations of Software Technology and Theoretical Computer Science*, volume 1738 of *Lecture Notes in Computer Science*, pages 97–109, December 1999.

[15] M. Michel. Complementation is more difficult with automata on infinite words. CNET, Paris, 1988.

[16] D.E. Muller. Infinite sequences and finite machines. In *Proc. 4th IEEE Symp. on Switching Circuit Theory and Logical design*, pages 3–16, 1963.

[17] M.O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:115–125, 1959.

[18] S. Safra. On the complexity of ω-automata. In *Proc. 29th IEEE Symp. on Foundations of Computer Science*, pages 319–327, White Plains, October 1988.

[19] W. Sakoda and M. Sipser. Non-determinism and the size of two-way automata. In *Proc. 10th ACM Symp. on Theory of Computing*, pages 275–286, 1978.

[20] A.P. Sistla, M.Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. *Theoretical Computer Science*, 49:217–237, 1987.

[21] F. Somenzi and R. Bloem. Efficient Büchi automata from LTL formulae. In *Computer Aided Verification, Proc. 12th International Conference*, volume 1855 of *Lecture Notes in Computer Science*, pages 248–263. Springer-Verlag, 2000.

[22] M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, November 1994.

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

[24] Q. Yang. Lower bounds for transformations of ω-automata. Shanghai Jiao Tong University, 2005.