Counting with Automata

Orna Kupferman, Hebrew University[*]

Amnon Ta-Shma[†], UC Berkeley

Moshe Y. Vardi, Rice University[‡]

March 26, 2000

Classical models of computations, such as Turing machines and automata, have been enriched with existential and universal branching to capture concurrency. Unlike the models used in the study of real concurrent systems, in these types of branching no cooperation takes place between the spawned precesses, except when time comes to decide whether the input should be accepted. It turned out that this weak cooperation is sufficient to make nondeterministic and universal automata exponentially more succinct than deterministic automata, and to make their combination, namely alternating automata, doubly exponentially more succinct than deterministic automata [CKS81]. Nevertheless, as studied in [DH94], enriching automata with real concurrency, where the spawned processes can cooperate all along the computation, results in even more succinct automata. In particular, concurrent alternating automata are triply exponentially more succinct than deterministic automata.

One of the natural examples of the power of concurrency is the ability to count to n with $\log n$ concurrent processes, with the i'th processor being responsible to the value of the i'th bit in the boolean representation of n. In this work we study the cost of counting within the other types of concurrency.

We first study the number of states required for an automaton to count to n, namely to accept the unary language $\{1^n\}$. It is easy to see that nondeterministic automata need $O(n)$ states to count to n. For other types of automata, the problem was stated in [MF71], and was studied in [Bir93, Bir96] for alternating and two-way automata. Here we complete the picture with results about universal automata. We show that nondeterminism, universality, and alternation induce three different counting costs: like deterministic automata, nondeterministic automata need $\Theta(n)$ states to count to n. Then, universal automata need only $\Theta(\sqrt{n})$ states, and finally, alternating automata can count to n with $\Theta(\log n)$ states. These bounds are tight. Some of our results are based on known succinctness relations between the various types of automata with respect to unary languages [Lei81, Cho86, MP98], yet not all lower bounds that hold for general unary languages hold for $\{1^n\}$. For example, unlike the case for general unary languages, where nondeterministic automata are more succinct than one-way automata, here, nondeterminism does not help with counting.

[*]Address: Institute of Computer Science and Engineering, Jerusalem 91904, Israel. Email: orna@cs.huji.ac.il

[†]Address: Department of Computer Science, Berkeley, CA 94720-1770, U.S.A. Email: amnon@cs.berkeley.edu

[‡]Address: Department of Computer Science, Houston, TX 77251-1892, U.S.A. Email: vardi@cs.rice.edu

In many applications, the only task of concurrency is efficient counting. The above results show that weaker types of concurrency are sufficient for efficient counting, thus they question the necessity of full concurrency in such applications. To approve the necessity of full concurrency, we also discuss "purposeful counting", where the task of counting should be accomplished before the whole input is read. Formally, we say that a given type of automata purposefully counts to n with $f(n)$ states iff for every language L that is accepted by an automaton of this type with k states, the language $(0+1)^{n-1} \cdot 0 \cdot L$ is accepted by an automaton of this type with $k + f(n)$ states. We show that all types of bounded concurrency cannot purposefully count efficiently, thus even alternating automata purposefully count to n with $\Theta(n)$ (rather than $\Theta(\log n)$) states.

References

[Bir93] J.C. Birget. State-complexity of finite-state devices, state compressibility and incompressibility. *Mathematical Systems Theory*, 26(3):237–269, 1993.

[Bir96] J.C. Birget. Two-way automata and length-preserving homomorphisms. *Mathematical Systems Theory*, 29(3):191–226, 1996.

[BL80] J.A. Brzozowski and E. Leiss. Finite automata and sequential networks. *Theoretical Computer Science*, 10:19–35, 1980.

[Cho86] M. Chorbak. Finite automata and unary languages. *Theoretical Computer Science*, 47:149–158, 1986.

[CKS81] A.K. Chandra, D.C. Kozen, and L.J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, January 1981.

[CS76] A.K. Chandra and L.J. Stockmeyer. Alternation. In *Proc. 17th IEEE Symposium on Foundations of Computer Science*, pages 98–108, New York, 1976.

[DH94] D. Drusinsky and D. Harel. On the power of bounded concurrency I: Finite automata. *Journal of the ACM*, 41(3):517–539, 1994.

[Lei81] E. Leiss. Succinct representation of regular languages by boolean automata. *Theoretical computer science*, 13:323–330, 1981.

[MF71] A.R. Meyer and M.J. Fischer. Economy of description by automata, grammars, and formal systems. In *Proc. 12th IEEE Symp. on Switching and Automata Theory*, pages 188–191, 1971.

[MP98] C. Mereghetti and G. Pighizzini. Optimal simulations between unary automata. In *Proc. 15th Symp. on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, 1998.