Testing an automaton for synchronization in linear expected time

Pavel Ageev, Yekaterinburg

Results

- The first practically effective implementation of the algorithm for testing an automaton for synchronization in linear expected time
- Statistically accurate evaluation of the fraction of synchronizing automata amongst all automata with n states

Basic notation

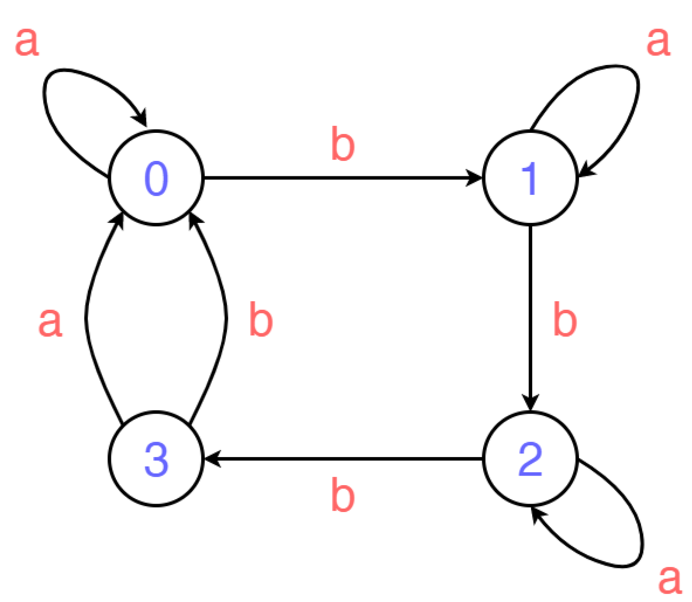
A deterministic finite automaton (DFA)

$$\mathcal{A} = (Q, \Sigma, \delta)$$

Q is a finite set of *states*
Σ stands for a finite *alphabet*
δ is a *transition function*

$$\delta : Q \times \Sigma \to Q$$

Σ^* is the set of all words over Σ



A DFA (Q, Σ, δ) is called *synchronizing* if there exists a word $w \in \Sigma^*$ that

$$\delta(q, w) = \delta(p, w) \text{ for each } q, p \in Q$$

Applications

- Part handling problems
- Biocomputing
- Coding theory

Many cute problems (partly open)

- Road coloring theorem
- Černý conjecture

The classic algorithm

Černý's synchronization criterion (1964)

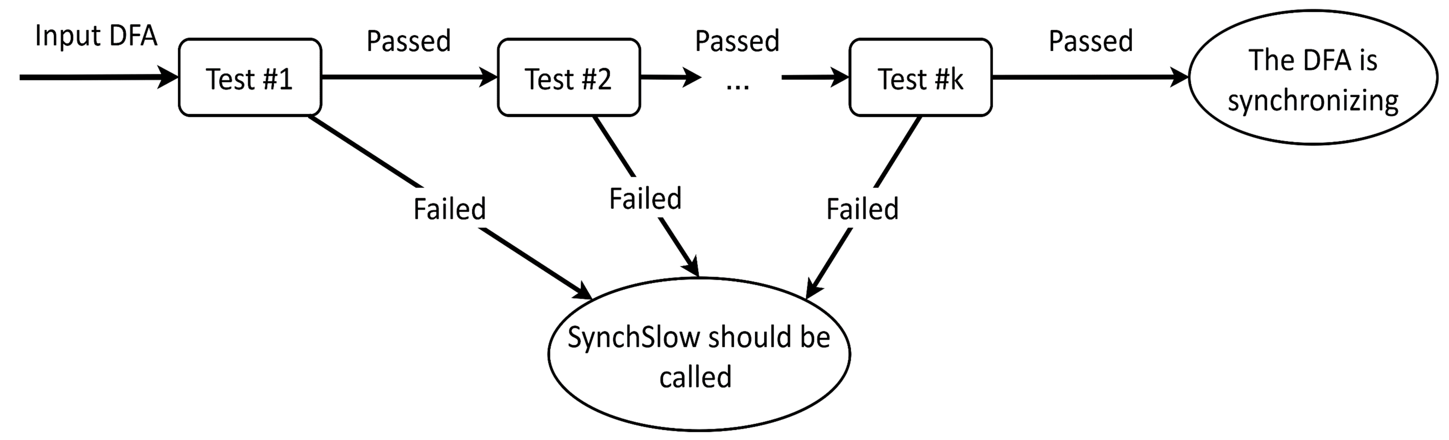
An automaton is synchronizing if and only if for each pair of states there exists a word mapping them to a single state.

The implied algorithm requires $\theta(|\Sigma||Q^2|)$ time.

The structure of the algorithm

The algorithm verifies certain conditions in a given automaton each of which has two features:

- it can be checked in $O(n)$ in every DFA with n states;
- the fraction of automata with n states not satisfying the condition is $O(1/n)$.



The expected time of the whole algorithm will be

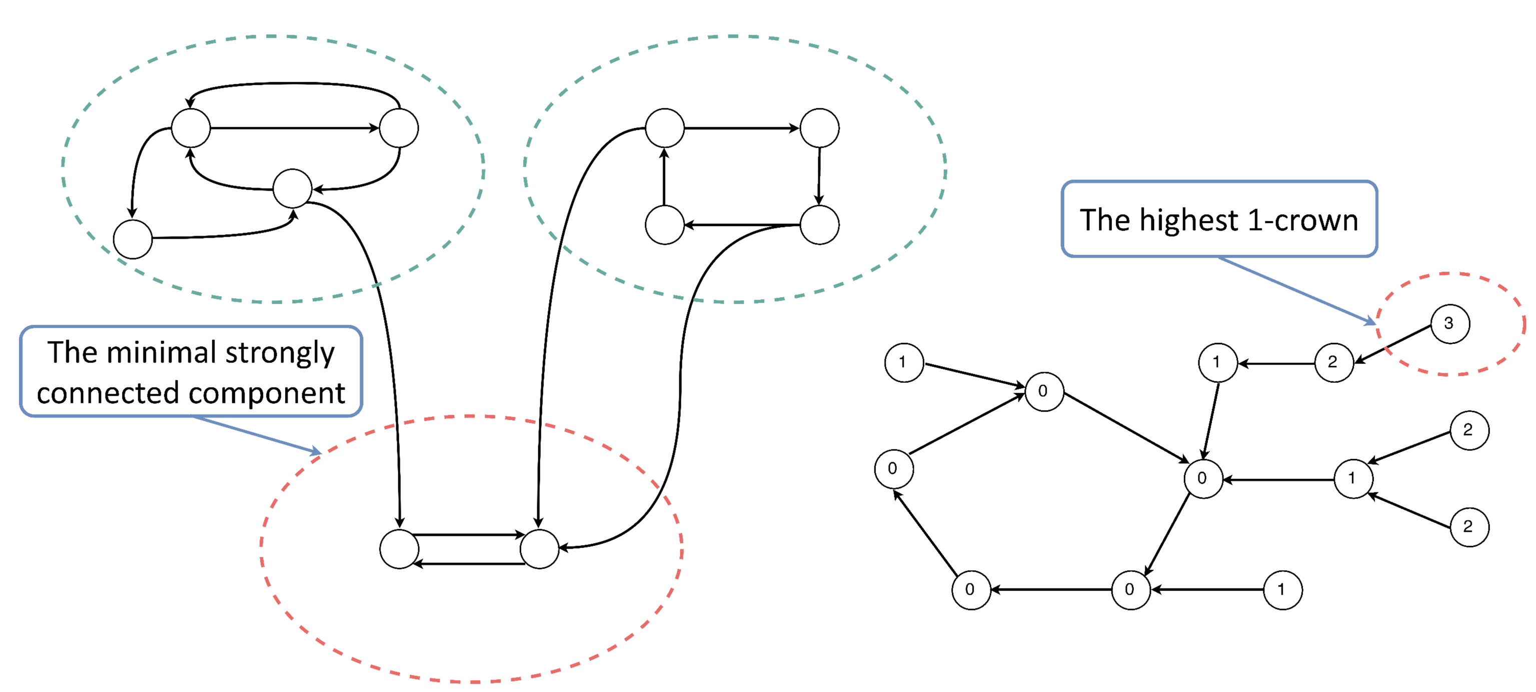
$$O(n)\left(1 - O\left(\frac{1}{n}\right)\right) + O(n^2)O\left(\frac{1}{n}\right) = O(n).$$

Our improvements

All the modifications consisted in

- omitting some of the conditions utilized in the original paper by Berlinkov
- checking a different condition instead.

An example of the improvement



The point is that, in all cases, it was the "new" condition that was implicitly used in the original algorithm while the role of the "old" condition was to ensure that the "new" one holds with high probability. Therefore checking the "new" condition instead of the "old" one straightens the algorithm and decreases the probability of invoking the slow algorithm.

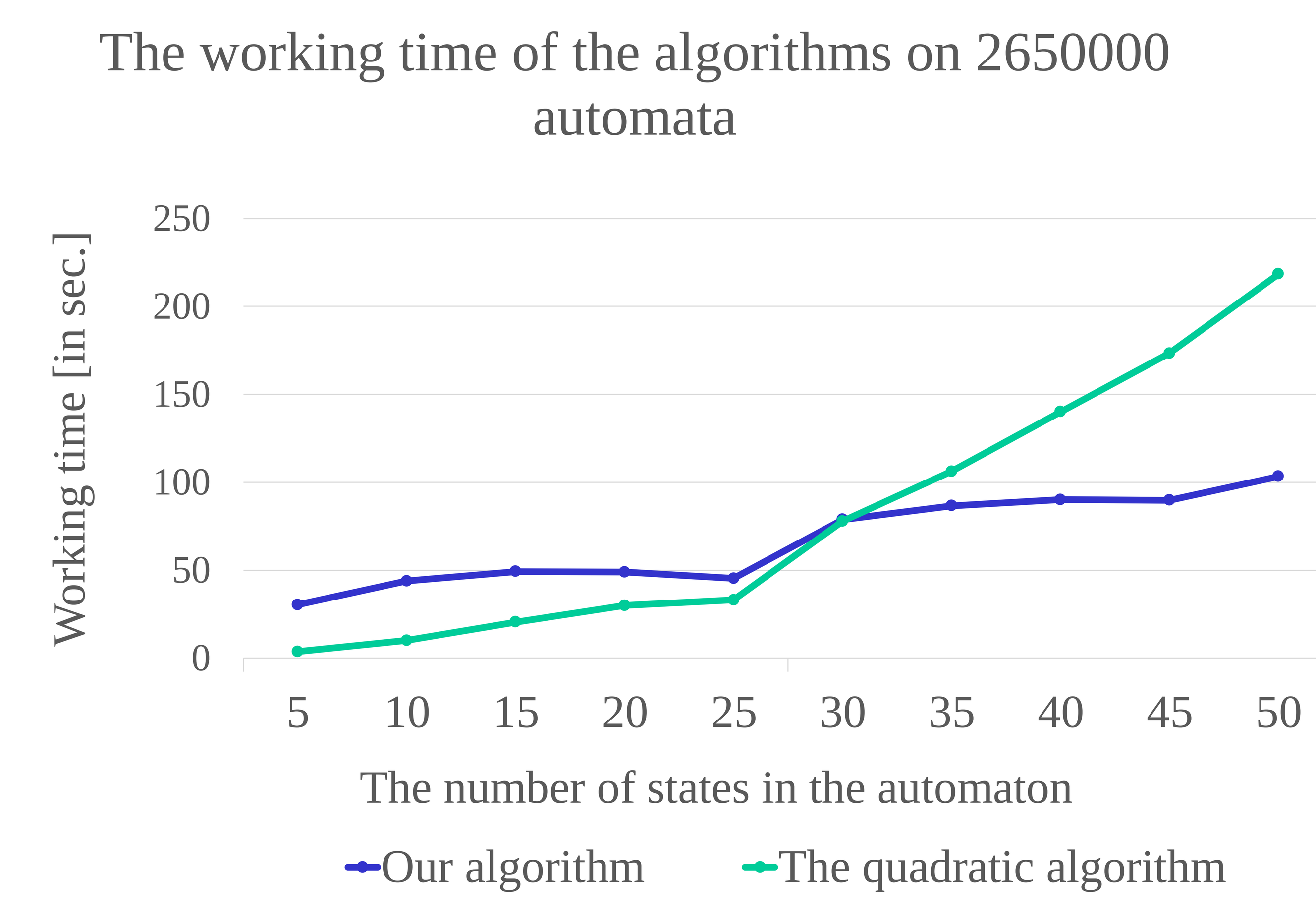
Probability of being synchronizable

The ratio of non-synchronizing automata among automata with n states and 2 input letters tends to C/n as $n \to \infty$ (Berlinkov). Our experiments provide a statistically accurate estimation of C in different models of random automata.

n	5	10	20	50	100	1000	10000
Estimate of C (uniform model)	3.57	4.6	4.8	4.33	3.79	5.09	5.32
Estimate of C (non-isomorphic model)	3.10	3.31	3.01	2.7	2.21	—	—

Comparison of performance

Our algorithm and the quadratic algorithm SynchSlow were run 2650000 times for all small n in order to find minimal $\boldsymbol{n_0}$ such that the average running time of SynchSlow becomes greater than the average running time of the main algorithm for automata with at least $\boldsymbol{n_0}$ states.



Acknowledgments

The author is grateful to his scientific advisor Professor Mikhail Volkov for helping with the article, to Mikhail Berlinkov for the explanation of his algorithm and to Mikhail Samoilenko for useful discussions.

References

Ageev, P.: Implementation of the algorithm for testing an automaton for synchronization in linear expected time, http://arxiv.org/abs/1710.00258 CoRR abs/1710.00258

Berlinkov, M.V.: On the probability of being synchronizable, pp. 73–84. LNCS, volume 9602 (2016). http://dx.doi.org/10.1007/978-3-319-29221-2_7