Journal of Automata, Languages and Combinatorics **24** (2019) 2–4, 139 – 152

IMPLEMENTATION OF THE ALGORITHM FOR TESTING AN AUTOMATON FOR SYNCHRONIZATION IN LINEAR EXPECTED TIME

Pavel Ageev[(A)]

Institute of Natural Sciences and Mathematics, Ural Federal University
Pr. Lenina 51, 620000 Ekaterinburg, Russia
pavel.ageev@urfu.ru

ABSTRACT

Berlinkov has suggested an algorithm that, given a deterministic finite automaton $\mathcal{A}$, verifies whether or not $\mathcal{A}$ is synchronizing in linear (of the number of states and letters) expected time. We present a modification of Berlinkov's algorithm which we have implemented and tested. Our experiments show that the implementation outperforms the standard quadratic algorithm even for automata of modest size and allow us to give a statistically accurate approximation of the ratio of non-synchronizing automata amongst all automata with a given number of states.

Keywords: synchronizing automata, Berlinkov's algorithm

1. Background and Motivation

A (deterministic finite) *automaton* (DFA, for short) is a triple $\mathcal{A} = (Q, \Sigma, \delta)$, where Q is a finite set of *states*, Σ stands for a finite *alphabet* and $\delta\colon Q \times \Sigma \to Q$ is a *transition function*. Let Σ^* be the set of all words over Σ, including the empty word ε. Each word $w \in \Sigma^*$ acts on Q via δ: namely, for each state $q \in Q$, we let[1]

$$q.w := \begin{cases} q, & \text{if } w = \varepsilon, \\ \delta(q.v, a), & \text{if } w = va \text{ for some } v \in \Sigma^* \text{ and } a \in \Sigma. \end{cases}$$

This action extends to subsets of Q: for $D \subseteq Q$, we define $D.w := \{\, q.w \mid q \in Q \,\}$.

An automaton $\mathcal{A} = (Q, \Sigma, \delta)$ is called *synchronizing* if there exists a word $w \in \Sigma^*$ whose action is a constant function, i. e., $q.w = p.w$ for all $p, q \in Q$. Every word with such a property is called *synchronizing*.

[(A)] The author acknowledges support by the Competitiveness Enhancement Program of Ural Federal University.

[1] Here and throughout expressions like $A := B$ mean that A is defined to be B.