Information and Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Complexity of a problem concerning reset words for Eulerian binary automata[☆]

Vojtěch Vorel

Faculty of Mathematics and Physics, Charles University Malostranské nám. 25, 118 00 Prague, Czech Republic

ARTICLE INFO

Article history:
Received 12 June 2014
Available online xxxx

ABSTRACT

A word is called a reset word for a deterministic finite automaton if it maps all the states of the automaton to a unique state. Deciding about the existence of a reset word of a given length for a given automaton is known to be an NP-complete problem. We prove that it remains NP-complete even if restricted to Eulerian automata with binary alphabets as it has been conjectured by Martyugin (2011).

1. Introduction and preliminaries

A *deterministic finite automaton* is a triple $A = (Q, X, \delta)$, where Q and X are finite sets and δ is an arbitrary mapping $Q \times X \to Q$. Elements of Q are called *states*, X is the *alphabet*. The *transition function* δ can be naturally extended to $Q \times X^\star \to Q$, still denoted by δ. We extend it also by defining

$$\delta(S, w) = \left\{\delta(s, w) \mid s \in S, w \in X^\star\right\}$$

for each $S \subseteq Q$. If the automaton is fixed, we write

$$r \xrightarrow{w} s$$

instead of $\delta(r, w) = s$.

For a given automaton $A = (Q, X, \delta)$, we call $w \in X^\star$ a *reset word* if

$$|\delta(Q, w)| = 1.$$

If such a word exists, we call the automaton *synchronizing*. Note that each word having a reset word as a factor is also a reset word.

A need for finding reset words appears in several fields of mathematics and engineering. Classical applications (see [14]) include model-based testing, robotic manipulation, and symbolic dynamics, but there are important connections also with information theory [13] and with formal models of biomolecular processes [2].

The *Černý Conjecture*, a longstanding open problem, claims that each synchronizing automaton has a reset word of length at most $(|Q| - 1)^2$. Though it still remains open, there are many weaker results in this field, see e.g. [11,7] for recent ones.[1]

☆ Research supported by the Czech Science Foundation grant GA14-10799S and the GAUK grant No. 52215.
E-mail address: vorel@ktiml.mff.cuni.cz.

[1] The result published by Trahtman [12] in 2011 was proved incorrectly, see [6].

http://dx.doi.org/10.1016/j.ic.2016.06.013
0890-5401/

Various computational problems arise from the study of synchronization:

- *Given an automaton, decide whether it is synchronizing.* Relatively simple algorithm, which could be traced back to [3], works in polynomial time.
- *Given a synchronizing automaton and a number d, decide whether d is the length of shortest reset words.* This has been shown to be both NP-hard [4] and coNP-hard. More precisely, it is DP-complete [10]. See also [1] and [5] for recent non-approximability results concerning an optimization setting.
- *Given a synchronizing automaton and a number d, decide whether there exists a reset word of length d.* This problem is of our interest. Lying in NP, it is not so computationally hard as the previous problem. However, it is proven to be NP-complete [4]. Following the notation of [9], we call it Syn. Assuming that $\mathcal{M}$ is a class of automata and membership in $\mathcal{M}$ is polynomially decidable, we define a restricted problem:

Syn($\mathcal{M}$)
Input: synchronizing automaton $A \in \mathcal{M}$, $d \in \mathbb{N}$
Output: does A have a reset word of length d?

An automaton $A = (Q, X, \delta)$ is *Eulerian* if

$$\sum_{x \in X} |\{r \in Q \mid \delta(r, x) = q\}| = |X|$$

for each $q \in Q$. Informally, there should be exactly $|X|$ transitions incoming to each state. An automaton is *binary* if $|X| = 2$. The classes of Eulerian and binary automata are denoted by $\mathcal{EU}$ and $\mathcal{AL}_2$ respectively.

Previous results about various restrictions of Syn can be found in [4,8,9]. Some of these problems turned out to be polynomially solvable, others are NP-complete. In [9] Martyugin conjectured that Syn($\mathcal{EU} \cap \mathcal{AL}_2$) is NP-complete. This conjecture is confirmed in the rest of the present paper.

2. Main result

2.1. Proof outline

We prove the NP-completeness of Syn($\mathcal{EU} \cap \mathcal{AL}_2$) by a polynomial reduction from 3-SAT. So, for an arbitrary propositional formula ϕ in 3-CNF we construct an Eulerian binary automaton A and a number d such that

$$\phi \text{ is satisfiable } \Leftrightarrow A \text{ has a reset word of length } d. \tag{1}$$

For the rest of the paper we fix a formula

$$\phi = \bigwedge_{i=1}^{m} \bigvee_{\lambda \in C_i} \lambda$$

on n variables, where each C_i is a three-element set of literals, i.e., a subset of

$$L_\phi = \{x_1, \dots, x_n, \neg x_1, \dots, \neg x_n\}.$$

We index the literals $\lambda \in L_\phi$ by the following mapping κ:

λ	x_1	x_2	$\dots$	x_n	$\neg x_1$	$\neg x_2$	$\dots$	$\neg x_n$
$\kappa(\lambda)$	0	1	$\dots$	$n-1$	n	$n+1$	$\dots$	$2n-1$

Let $A = (Q, X, \delta)$, $X = \{a, b\}$. Because the structure of the automaton A will be very heterogeneous, we use an unusual method of description. The basic principles of the method are:

- We describe the automaton A via a labeled directed multigraph G, representing the automaton in a standard way: edges of G are labeled by single letters a and b and carry the structure of the function δ. Paths in G are thus *labeled* by words from $\{a, b\}^\star$.
- There is a collection of labeled directed multigraphs called *templates*. The graph G is one of them. Another template is $\mathtt{SINGLE}$, which consists of one vertex and no edges.
- Each template $\mathtt{T} \neq \mathtt{SINGLE}$ is expressed in a fixed way as a disjoint union through the set $\mathrm{PARTS}_{\mathtt{T}}$ of its proper subgraphs (the *parts* of $\mathtt{T}$), extended by a set of additional edges (the *links* of $\mathtt{T}$). Each $H \in \mathrm{PARTS}_{\mathtt{T}}$ is isomorphic to some template $\mathtt{U}$. We say that H *is of type* $\mathtt{U}$.