Completely Reachable Automata

Eugenija Bondar, Mikhail Volkov

Completely Reachable Automata*

E. A. Bondar and M. V. Volkov

Institute of Mathematics and Computer Science
Ural Federal University, Lenina 51, 620000 Ekaterinburg, Russia
bondareug@gmail.com, mikhail.volkov@usu.ru

Abstract. We present a few results and several open problems concerning complete deterministic finite automata in which every non-empty subset of the state set occurs as the image of the whole state set under the action of a suitable input word.

Keywords: Deterministic finite automaton, Complete reachability, Transition monoid, Syntactic complexity, PSPACE-completeness

1 Background and overview

We consider the most classical species of finite automata, namely, complete deterministic automata. Recall that a *complete deterministic finite automaton* (DFA) is a triple $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$, where Q and Σ are finite sets called the *state set* and the *input alphabet* respectively, and $\delta\colon Q\times\Sigma \to Q$ is a totally defined map called the *transition function*. Let Σ^* stand for the collection of all finite words over the alphabet Σ, including the empty word. The function δ extends to a function $Q \times \Sigma^* \to Q$ (still denoted by δ) in the following natural way: for every $q \in Q$ and $w \in \Sigma^*$, we set $\delta(q, w) := q$ if w is empty and $\delta(q, w) := \delta(\delta(q, v), a)$ if $w = va$ for some word $v \in \Sigma^*$ and some letter $a \in \Sigma$. Thus, via δ, every word $w \in \Sigma^*$ induces a transformation of the set Q.

Let $\mathcal{P}(Q)$ stand for the set of all non-empty subsets of the set Q. The function δ can be further extended to a function $\mathcal{P}(Q) \times \Sigma^* \to \mathcal{P}(Q)$ (again denoted by δ) by letting $\delta(P, w) := \{\delta(q, w) \mid q \in P\}$ for every non-empty subset $P \subseteq Q$. Thus, the triple $\mathcal{P}(\mathscr{A}) := \langle\mathcal{P}(Q), \Sigma, \delta\rangle$ is a DFA again; this DFA is referred to as the *powerset automaton* of $\mathscr{A}$.

* Supported by the Russian Foundation for Basic Research, grant no. 16-01-00795, the Ministry of Education and Science of the Russian Federation, project no. 1.1999.2014/K, and the Competitiveness Program of Ural Federal University. The paper was written during the second author's stay at Hunter College of the City University of New York as Ada Peluso Visiting Professor of Mathematics and Statistics with a generous support from the Ada Peluso Endowment.

Whenever we deal with a fixed DFA, we simplify our notation by suppressing the sign of the transition function; this means that we may introduce the DFA as the pair $\langle Q, \Sigma\rangle$ rather than the triple $\langle Q, \Sigma, \delta\rangle$ and may write $q\,.\,w$ for $\delta(q, w)$ and $P\,.\,w$ for $\delta(P, w)$.

Given a DFA $\mathscr{A} = \langle Q, \Sigma\rangle$, we say that a non-empty subset $P \subseteq Q$ is *reachable* in $\mathscr{A}$ if $P = Q\,.\,w$ for some word $w \in \Sigma^*$. A DFA is called *completely reachable* if every non-empty subset of its state set is reachable.

Let us start with an example that served as a first spark which ignited our interest in completely reachable automata. A DFA $\mathscr{A} = \langle Q, \Sigma\rangle$ is called *synchronizing* if it has a reachable singleton, that is, $Q\,.\,w$ is a singleton for some word $w \in \Sigma^*$. Any such word w is said to be a *reset word* for the DFA. The minimum length of reset words for $\mathscr{A}$ is called the *reset threshold* of $\mathscr{A}$. In 1964 Černý [8] constructed for each $n > 1$ a synchronizing automaton $\mathscr{C}_n$ with n states, 2 input letters, and reset threshold $(n-1)^2$. Recall the definition of $\mathscr{C}_n$. If we denote the states of $\mathscr{C}_n$ by $1, 2, \ldots, n$ and the input letters by a and b, the actions of the letters are as follows:

$$i\,.\,a := \begin{cases} i & \text{if } i < n,\\ 1 & \text{if } i = n; \end{cases} \qquad i\,.\,b := \begin{cases} i+1 & \text{if } i < n,\\ 1 & \text{if } i = n. \end{cases}$$

The automaton $\mathscr{C}_n$ is shown in Fig. 1.

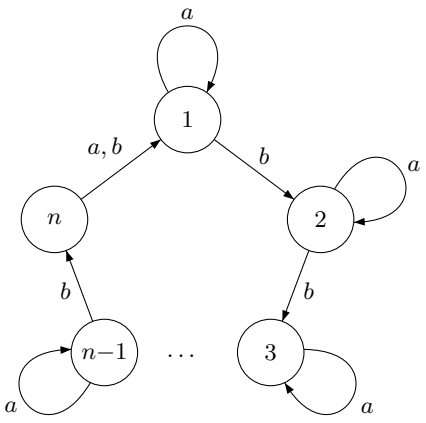

Fig. 1. The automaton $\mathscr{C}_n$

The automata in the Černý series are well-known in the connection with the famous Černý conjecture about the maximum reset threshold for synchronizing automata with n states, see [18]. The automata $\mathscr{C}_n$

provide the lower bound $(n-1)^2$ for this maximum, and the conjecture claims that these automata represent the worst possible case since it has been conjectured that every synchronizing automaton with n states can be reset by a word of length $(n-1)^2$. The automata $\mathscr{C}_n$ also have other interesting properties, including the one registered here:

Example 1. Each automaton $\mathscr{C}_n$, $n > 1$, is completely reachable.

The result of Example 1 was first observed by Maslennikova [15, Proposition 2], see also [16], in the course of her study of the so-called reset complexity of regular ideal languages. Later, Don [9, Theorem 1] found a sufficient condition for complete reachability that applies to the automata $\mathscr{C}_n$. In Section 2 we present another sufficient condition that both simplifies and generalizes Don's one. We provide an example showing that our condition is not necessary but we conjecture that it may be necessary for a stronger version of complete reachability.

In Section 3 we discuss the problem of recognizing completely reachable automata. We show PSPACE-completeness of the following decision problem: given a DFA $\mathscr{A} = \langle Q, \Sigma \rangle$ and a subset $P \subseteq Q$, decide whether or not P is reachable in $\mathscr{A}$. We also outline a polynomial algorithm that recognizes completely reachable automata with 2 input letters modulo the conjecture from Section 2.

Given a DFA $\mathscr{A} = \langle Q, \Sigma \rangle$, its *transition monoid* $M(\mathscr{A})$ is the monoid of all transformations of the set Q induced by the words in Σ^*. By the *syntactic complexity* of $\mathscr{A}$ we mean the size of $M(\mathscr{A})$. Clearly, the syntactic complexity of a completely reachable automaton $\mathscr{A}$ with n states cannot be less than $2^n - 1$ since, for each non-empty subset P of the state set, the transition monoid of $\mathscr{A}$ must contain a transformation whose image is P. In Section 4 we address the question of the existence and classification of *minimal* completely reachable automata, i.e., completely reachable automata with minimum possible syntactic complexity. This question has been recently investigated in the realm of transformation monoids by the first author [3, 4]; here we translate her results into the language of automata theory and augment them by determining the input alphabet size of minimal completely reachable automata.

The present paper is in fact a work-in-progress report, and therefore, each of Sections 2–4 includes some open questions. Several additional open questions form Section 5; they mostly deal with synchronization properties of completely reachable automata.

We assume the reader's acquaintance with some basic concepts of graph theory, monoid theory, and computational complexity.

2 A Sufficient Condition

If Q is a finite set, we denote by $T(Q)$ the *full transformation monoid* on Q, i.e., the monoid consisting of all transformations $\varphi\colon Q \to Q$. For $\varphi \in T(Q)$, its *defect* is defined as the size of the set $Q \setminus Q\varphi$. Observe that the defect of a product of transformations is greater than or equal to the defect of any of the factors and is equal to the defect of a factor whenever the other factors are permutations of Q. In particular, if a product of transformations has defect 1, then one of the factors must have defect 1.

Let $\mathscr{A} = \langle Q, \Sigma\rangle$ be a DFA. The defect of a word $w \in \Sigma^*$ with respect to $\mathscr{A}$ is the defect of transformation induced by w. Consider a word w of defect 1. For such a word, the set $Q \setminus Q \,.\, w$ consists of a unique state, which is called the *excluded state* for w and is denoted by $\mathrm{excl}(w)$. Further, the set $Q \,.\, w$ contains a unique state p such that $p = q_1 \,.\, w = q_2 \,.\, w$ for some $q_1 \neq q_2$; this state p is called the *duplicate state* for w and is denoted by $\mathrm{dupl}(w)$. Let $D_1(\mathscr{A})$ stand for the set of all words of defect 1 with respect to $\mathscr{A}$, and let $\Gamma_1(\mathscr{A})$ denote the directed graph having Q as the vertex set and the set

$$E_1 := \{(\mathrm{excl}(w), \mathrm{dupl}(w)) \mid w \in D_1(\mathscr{A})\}$$

as the edge set. Since we consider only directed graphs in this paper, we call them just graphs in the sequel. Recall that a graph is *strongly connected* if for every pair of its vertices, there exists a directed path from the first vertex to the second.

Theorem 1. *If a DFA $\mathscr{A} = \langle Q, \Sigma\rangle$ is such that the graph $\Gamma_1(\mathscr{A})$ is strongly connected, then $\mathscr{A}$ is completely reachable.*

Proof. Take an arbitrary non-empty subset $P \subseteq Q$. We prove that P is reachable in $\mathscr{A}$ by induction on $k := |Q \setminus P|$. If $k = 0$, then $P = Q$ and nothing is to prove as Q is reachable via the empty word. Now let $k > 0$ so that P is a proper subset of Q. Since the graph $\Gamma_1(\mathscr{A})$ is strongly connected, there exists an edge $(q, p) \in E_1$ that connects $Q \setminus P$ and P in the sense that $q \in Q \setminus P$ while $p \in P$. By the definition of E_1, there exists a word w of defect 1 with respect to $\mathscr{A}$ for which q is the excluded state and p is the duplicate state. By the definition of the duplicate state, $p = q_1 \,.\, w = q_2 \,.\, w$ for some $q_1 \neq q_2$, and since the excluded state q for w does not belong to P, for each state $r \in P \setminus \{p\}$, there exists a unique state $r' \in Q$ such that $r = r' \,.\, w$. Now letting $R := \{q_1, q_2\} \cup \{r' \mid r \in P \setminus \{p\}\}$, we conclude that $P = R \,.\, w$ and $|R| = |P| + 1$. Then $|Q \setminus R| = k - 1$, and the induction assumption applies to the subset R whence $R = Q \,.\, v$ for some word $v \in \Sigma^*$. Then $P = Q \,.\, vw$ so that P is reachable as required.

Don [9] has formulated a sufficient condition for complete reachability in the terms of what he called a state map. Consider a DFA $\mathscr{A} = \langle Q, \Sigma \rangle$ with n states in which every subset of size $n-1$ is reachable. Let W be a set of n words of defect 1 with respect to $\mathscr{A}$ such that for every subset $P \subset Q$ with $|P| = n-1$ there is a unique word $w \in W$ with $P = Q\,.\,w$. (Such a set is termed a 1-*contracting collection* in [9]). The *state map* $\sigma_W \colon Q \to Q$ induced by W is defined by

$$q\sigma_W := \mathrm{dupl}(w) \quad \text{for } w \in W \text{ such that } \quad q = \mathrm{excl}(w).$$

The following is one of the main results in [9]:

Theorem 2. *A DFA $\mathscr{A}$ is completely reachable if it admits a 1-contracting collection such that the induced state map is a cyclic permutation of the state set of $\mathscr{A}$.*

Even though Theorem 2 is stated in different terms, it is easily seen to constitute a special case of Theorem 1. Indeed, if W is a 1-contracting collection and σ_W is the corresponding state map, then each pair $(q, q\sigma_W)$ can be treated as an edge in E_1. Therefore, if σ_W is a cyclic permutation of Q, then the set of edges $\{(q, q\sigma_W) \mid q \in Q\}$ forms a directed Hamiltonian cycle in the graph $\Gamma_1(\mathscr{A})$ whence the latter is strongly connected.

We believe that Theorem 1 may have strongly wider application range than Theorem 2 even though at the moment we do not have any example confirming this conjecture. If the conditions of the two theorems were equivalent, every strongly connected graph of the form $\Gamma_1(\mathscr{A})$ would possess a directed Hamiltonian cycle, and this does not seem to be likely.

Now we demonstrate that the condition of Theorem 1 is not necessary.

Example 2. Consider the DFA $\mathscr{E}_3$ with the state set $\{1, 2, 3\}$ and the input letters $a_{[1]}, a_{[2]}, a_{[3]}, a_{[1,2]}$ that act as follows:

$$i\,.\,a_{[1]} := \begin{cases} 2 & \text{if } i = 1, 2, \\ 3 & \text{if } i = 3; \end{cases} \qquad i\,.\,a_{[2]} := \begin{cases} 1 & \text{if } i = 1, 2, \\ 3 & \text{if } i = 3; \end{cases}$$

$$i\,.\,a_{[3]} := \begin{cases} 1 & \text{if } i = 1, 2, \\ 2 & \text{if } i = 3; \end{cases} \qquad i\,.\,a_{[1,2]} := 3 \quad \text{for all} \quad i = 1, 2, 3.$$

The automaton $\mathscr{E}_3$ is shown in Fig. 2 on the left. The graph $\Gamma_1(\mathscr{E}_3)$ is shown in Fig. 2 on the right; it is not strongly connected. However, it can be checked by a straightforward computation that the automaton $\mathscr{E}_3$ is completely reachable.

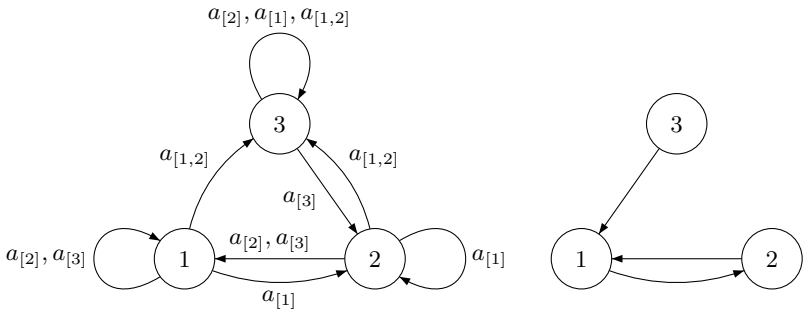

Fig. 2. The automaton $\mathscr{E}_3$ and the graph $\Gamma_1(\mathscr{E}_3)$

The reason of why the converse of Theorem 1 fails becomes obvious if one analyzes the above proof. In fact, we have proved more than we have formulated, namely, our proof shows that if a DFA $\mathscr{A}$ is such that the graph $\Gamma_1(\mathscr{A})$ is strongly connected, then *every proper non-empty subset of the state set of* $\mathscr{A}$ *is reachable via a product of words of defect* 1. Of course, this stronger property has no reason to hold in an arbitrary completely reachable automaton. For instance, in the automaton $\mathscr{E}_3$ of Example 2 the singleton $\{3\}$ is not an image of any product of words of defect 1. On the other hand, for the stronger property italicized above, the condition of Theorem 1 may be not only sufficient but also necessary. We formulate this guess as a conjecture.

Conjecture 1. If for every proper non-empty subset P of the state set of a DFA $\mathscr{A}$ there is a product w of words of defect 1 with respect to $\mathscr{A}$ such that $P = Q\,.\,w$, the graph $\Gamma_1(\mathscr{A})$ is strongly connected.

One can formulate further sufficient conditions for complete reachability in terms of strong connectivity of certain *hypergraphs* related to words of defect 2.

3 Complexity of Deciding Reachability

Given a DFA, one can easily decide whether or not it is completely reachable considering its powerset automaton: a DFA $\mathscr{A} = \langle Q, \Sigma\rangle$ is completely reachable if and only if Q is connected with every its non-empty subset by a directed path in the powerset automaton $\mathcal{P}(\mathscr{A})$, and the latter property can be recognized by breadth-first search on $\mathcal{P}(\mathscr{A})$ starting at Q. This algorithm is however exponential with respect to the size of $\mathscr{A}$, and it is natural to ask whether or not complete reachability can be decided in polynomial time. First, consider the following decision problem:

Reachable Subset: *Given a DFA* $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *and a non-empty subset* $P \subseteq Q$*, is it true that* P *is reachable in* $\mathscr{A}$*?*

Theorem 3. *The problem* Reachable Subset *is* PSPACE*-complete.*

Proof. The fact that Reachable Subset is in the class PSPACE is easy and known, see, e.g., [5, Lemma 6, item 1].

To prove PSPACE-hardness of Reachable Subset, we reduce to it in logarithmic space the well-known PSPACE-complete problem FAI (Finite Automata Intersection, see [14]). Recall that an instance of FAI consists of k DFAs $\mathscr{A}_j = \langle Q_j, \Sigma, \delta_j \rangle$, $j = 1, \ldots, k$, with disjoint state sets and a common input alphabet. In each DFA $\mathscr{A}_j$ an *initial state* $s_j \in Q_j$ and a *final state* $t_j \in Q_j$ are specified; a word $w \in \Sigma^*$ is said to be *accepted* by $\mathscr{A}_j$ if $\delta_j(s_j, w) = t_j$. The question of FAI asks whether or not there exists a word $w \in \Sigma^*$ which is simultaneously accepted by all automata $\mathscr{A}_1, \ldots, \mathscr{A}_k$.

Now, given an instance of FAI as above, we construct the following instance $(\mathscr{A}, P)$ of Reachable Subset. The state set of the DFA $\mathscr{A}$ is $Q := \bigcup_{j=1}^{k} Q_j$; the input alphabet of $\mathscr{A}$ is Σ with one extra letter ρ added. The transition function $\delta\colon Q \times (\Sigma \cup \{\rho\}) \to Q$ is defined by the rule

$$\delta(q, a) := \begin{cases} \delta_j(q, a) & \text{if } a \in \Sigma \text{ and } q \in Q_j, \\ s_j & \text{if } a = \rho \text{ and } q \in Q_j. \end{cases} \tag{1}$$

Expressing this rule less formally, it says that, given a state $q \in Q$, one first should find the index $j \in \{1, \ldots, k\}$ such that q belongs to Q_j; then every letter $a \in \Sigma$ acts on q in the same way as it does in the automaton $\mathscr{A}_j$ while the added letter ρ sends q to the initial state s_j of $\mathscr{A}_j$ (so ρ artificially 'initializes' each $\mathscr{A}_j$). Observe that each set Q_j is closed under the action of each letter in $\Sigma \cup \{\rho\}$. Finally, we set $P := \{t_1, \ldots, t_k\}$, that is, P consists of the final states of $\mathscr{A}_1, \ldots, \mathscr{A}_k$.

We claim that the subset P is reachable in $\mathscr{A}$ if and only if there exists a word $w \in \Sigma^*$ which is simultaneously accepted by all automata $\mathscr{A}_1, \ldots, \mathscr{A}_k$. Indeed, if such a word w exists, then $\delta(Q, \rho w) = P$ since we have $\delta(Q, \rho) = \{s_1, \ldots, s_k\}$ by (1) and $\delta(s_j, w) = \delta_j(s_j, w) = t_j$ for each $j = 1, \ldots, k$ by the choice of w. Conversely, suppose that P is reachable in $\mathscr{A}$, that is, $\delta(Q, u) = P$ for some word $u \in (\Sigma \cup \{\rho\})^*$. Then we must have $\delta_j(Q_j, u) = \{t_j\}$ for each $j = 1, \ldots, k$. If the word u has no occurrence of the letter ρ, then $u \in \Sigma^*$ and $\delta_j(s_j, u) = \{t_j\}$ for each $j = 1, \ldots, k$ so that u is simultaneously accepted by all automata $\mathscr{A}_1, \ldots, \mathscr{A}_k$. Otherwise we fix the rightmost occurrence of ρ in u and denote by w the suffix of u following

this occurrence so that $w \in \Sigma^*$ and $u = v\rho w$ for some $v \in (\Sigma \cup \{\rho\})^*$. Then $\delta_j(Q_j, v\rho) = \{s_j\}$ and $\delta_j(s, w) = \delta(Q_j, v\rho w) = \{t_j\}$ for each $j = 1, \ldots, k$. We conclude that w is simultaneously accepted by all automata $\mathscr{A}_1, \ldots, \mathscr{A}_k$. This completes the proof of our claim and establishes the reduction which obviously can be implemented in logarithmic space.

The reduction used in the above proof is an adaptation of a slightly more involved log-space reduction used by Brandl and Simon [5, Section 3] to show PSPACE-hardness of a natural problem about transformation monoids presented by a bunch of generating transformations.

In connection with Theorem 3, an interesting result by Goralčík and Koubek [13, Theorem 1] is worth being mentioned. If stated in the language adopted in the present paper, their result says that, given a DFA $\mathscr{A} = \langle Q, \Sigma \rangle$ with $|Q| = n$, $|\Sigma| = m$ and a subset $P \subseteq Q$ with $|P| = k$, one can decide in $O\big((k+1)n^{k+1}m\big)$ time whether or not there exists a word $w \in \Sigma^*$ such that $P = Q \,.\, w = P \,.\, w$. (The difference from our definition of reachability is that here one looks for a word not only having the subset P as its image but also acting on P as a permutation.) Thus, if the size of the target set P is treated as a parameter, the algorithm from [13] becomes polynomial. One can ask if a similar result holds for the parameterized version of Reachable Subset formulated as follows:

Reachable Subset$_k$: *Given a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ and a non-empty subset $P \subseteq Q$ of size k, is it true that P is reachable in $\mathscr{A}$?*

For $k = 1$, the cited result by Goralčík and Koubek applies since, for P being a singleton, any word $w \in \Sigma^*$ such that $P = Q \,.\, w$ automatically satisfies the additional condition $P \,.\, w = P$. For $k > 1$, the question about the complexity of Reachable Subset$_k$ is open. The reduction from the proof of Theorem 3 cannot help here because the size k of the subset P in this reduction is equal to the number of DFAs in the instance of FAI from which we depart, and for each fixed k, there is a polynomial algorithm that decides on all instances of FAI with k automata.

Now we return to the question of whether or not complete reachability can be decided in polynomial time. It should be noted that Theorem 3 does not imply any hardness conclusion here: while checking reachability of individual subsets is PSPACE-complete, checking reachability of all non-empty subsets may still be polynomial even though the latter problem consists of exponentially many individual problems! One can illustrate this phenomenon of 'simplification due to collectivization' with the following example. If is known [14] that the following *membership problem* for transition monoids of DFAs is PSPACE-complete: given a

DFA $\mathscr{A} = \langle Q, \Sigma \rangle$ and a transformation $\varphi\colon Q \to Q$, does φ belongs to the transition monoid $M(\mathscr{A})$, i.e., is there a word $w \in \Sigma^*$ such that $q\varphi = q \,.\, w$ for all $q \in Q$? On the other hand, one can decide in polynomial time whether or not *every* transformation of the state set belongs to the transition monoid of a given DFA. Indeed, given a DFA $\mathscr{A} = \langle Q, \Sigma \rangle$, we partition the alphabet Σ as $\Sigma = \Pi \cup \Delta$, where Π consists of all letters that act on Q as permutations and Δ contains all letters with non-zero defect. First we inspect Δ: if no letter in Δ has defect 1, then it is clear that the monoid $M(\mathscr{A})$ contains no transformation of defect 1 (see the observation registered at the beginning of Section 2). Further, we invoke twice the polynomial algorithm by Furst, Hopcroft and Luks [11] for the membership problem in permutation groups: we fix a cyclic permutation and a transposition of Q and check if they belong to the permutation group on Q generated by the permutations induced by the letters in Π. If the answers to all these queries are affirmative, then $M(\mathscr{A})$ contains a cyclic permutation, a transposition, and a transformation of defect 1, and it is well-known that any such trio of transformations generates the full transformation monoid $T(Q)$, see, e.g., [12, Theorem 3.1.3].

Thus, the complexity of deciding complete reachability for a given DFA remains unknown so far. We expect this problem to be computationally hard for automata over unrestricted alphabets while for automata with a fixed number of letters a polynomial algorithm may exist. For instance, if Conjecture 1 holds true, there exists a polynomial algorithm that recognizes completely reachable automata among DFAs with 2 input letters. Indeed, let $\mathscr{A} = \langle Q, \{a, b\} \rangle$ be a DFA with n states, $n > 1$. Every subset of the form $Q \,.\, w$, where w is a non-empty word over $\{a, b\}$, is contained in either $Q \,.\, a$ or $Q \,.\, b$. At least one of the letters must have defect 1 since no subset of size $n - 1$ is reachable otherwise, and if the other letter has defect greater than 1, only one subset of size $n - 1$ is reachable. Hence, if $\mathscr{A}$ is a completely reachable automaton, one of its letters has defect 1 while the other has defect at most 1. Therefore for each proper reachable subset $P \subset Q$, there is a product w of words of defect 1 with respect to $\mathscr{A}$ such that $P = Q \,.\, w$. In view of Theorem 1, if Conjecture 1 holds true, then complete reachability of $\mathscr{A}$ is equivalent to strong connectivity of the graph $\Gamma_1(\mathscr{A})$. It remains to show that for automata with 2 input letters, the latter condition can be verified in polynomial time.

Once the graph $\Gamma_1(\mathscr{A})$ is constructed, checking its strong connectivity in polynomial time makes no difficulty. However, it is far from being obvious that $\Gamma_1(\mathscr{A})$, even though it definitely has polynomial size, can

always be constructed in polynomial time. Indeed, by the definition, the edges of $\Gamma_1(\mathscr{A})$ arise from transformations of defect 1 in the transition monoid of $\mathscr{A}$, and for an automaton with n states, the number of transformations of defect 1 in $M(\mathscr{A})$ may reach $n!\binom{n}{2}$. Our algorithm depends on some peculiarities of automata with 2 input letters. It incrementally appends edges to a spanning subgraph of $\Gamma_1(\mathscr{A})$ in a way such that one can reach a conclusion about strong connectivity of $\Gamma_1(\mathscr{A})$ by examining only polynomially many transformations of defect 1. In the following brief description of the algorithm, we use the notation introduced in Section 2 in the course of defining the graph $\Gamma_1(\mathscr{A})$.

Thus, again, let $\mathscr{A} = \langle Q, \{a, b\}\rangle$ be a DFA with n states, $n > 1$. For certainty, let a stand for the letter of defect 1. If b also has defect 1, then at most two subsets of size $n-1$ are reachable (namely, $Q\,.\,a$ and $Q\,.\,b$), and $\mathscr{A}$ can only be completely reachable provided that $n = 2$. The automaton $\mathscr{A}$ is then nothing but the classical flip-flop, see Fig. 3.

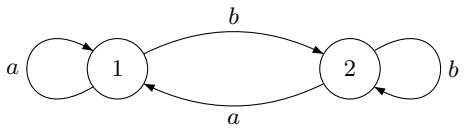

Fig. 3. Filp-flop

Beyond this trivial case, b must be a permutation of Q whence b^n acts on Q as the identity transformation. Then the set $\{\mathrm{excl}(w) \mid w \in D_1(\mathscr{A})\}$ of the states at which edges of $\Gamma_1(\mathscr{A})$ may originate is easily seen to coincide with the set $\{\mathrm{excl}(a), \mathrm{excl}(ab), \ldots, \mathrm{excl}(ab^{n-1})\}$. For $\Gamma_1(\mathscr{A})$ to be strongly connected, it is necessary that every vertex is an origin of an edge whence the latter set must be equal to Q. Taking into account that $\mathrm{excl}(ab^k) = \mathrm{excl}(a)\,.\,b^k$ for each $k = 1, \ldots, n-1$, we conclude that b must be a cyclic permutation of Q. It is easy to show that $\mathrm{excl}(w)\,.\,b = \mathrm{excl}(wb)$ and $\mathrm{dupl}(w)\,.\,b = \mathrm{dupl}(wb)$ for every word w of defect 1, and therefore, b acts as a permutation on the edge set E_1 of $\Gamma_1(\mathscr{A})$.

The set E_1 contains the edges

$$(\mathrm{excl}(a), \mathrm{dupl}(a)), \ldots, (\mathrm{excl}(ab^{n-1}), \mathrm{dupl}(ab^{n-1})). \tag{2}$$

Since $\mathrm{dupl}(ab^k) = \mathrm{dupl}(a)\,.\,b^k$ for each $k = 1, \ldots, n-1$, the edges in (2) are the 'translates' of the edge $(\mathrm{excl}(a), \mathrm{dupl}(a))$. Any two edges in (2) start at different vertices and end at at different vertices, whence for

some d such that $d < n$ and d divides n, the edges in (2) form d directed cycles, each of size $\frac{n}{d}$. If $d = 1$, we can already conclude that the graph $\Gamma_1(\mathscr{A})$ is strongly connected. If $d > 1$, denote the cycles by $C_1, \ldots, C_d$ and consider the words $a^2, aba, \ldots, ab^{n-1}a$. It can be easily shown that exactly two of them have defect 1; let us denote these two words by w_1 and w_2. Since w_1 and w_2 end with a, we have $Q \,.\, w_1 = Q \,.\, w_2 = Q \,.\, a$ whence $\mathrm{excl}(w_1) = \mathrm{excl}(w_2) = \mathrm{excl}(a)$. Thus, the edges $(\mathrm{excl}(w_1), \mathrm{dupl}(w_1))$ and $(\mathrm{excl}(w_2), \mathrm{dupl}(w_2))$ start at the vertex $\mathrm{excl}(a)$ which can be assumed to belong to the cycle C_1. If also the ends of these edges lie in C_1, one can show that no further edge in E_1 can connect C_1 with another cycle whence C_1 forms a strongly connected component of $\Gamma_1(\mathscr{A})$. We then conclude that $\Gamma_1(\mathscr{A})$ is not strongly connected.

Now suppose that the edge $(\mathrm{excl}(w_i), \mathrm{dupl}(w_i))$ where $i = 1$ or $i = 2$ connects the vertex $\mathrm{excl}(a)$ with a vertex from the cycle C_j where $j > 1$. Then we append the edge and all its translates $(\mathrm{excl}(w_i b^k), \mathrm{dupl}(w_i b^k))$, $k = 1, \ldots, n-1$, to $C_1, \ldots, C_d$; in the case where both $(\mathrm{excl}(w_1), \mathrm{dupl}(w_1))$ and $(\mathrm{excl}(w_2), \mathrm{dupl}(w_2))$ leave C_1, we append both these edges and all their translates. After that, we get larger strongly connected subgraphs $D_1, \ldots, D_\ell$ isomorphic to each other, where $\ell < d$ and ℓ divides d. If $\ell = 1$, then the graph $\Gamma_1(\mathscr{A})$ is strongly connected. If $\ell > 1$, we iterate by considering the words $w_i a, w_i ba, \ldots, w_i b^{n-1}a$. Eventually, either we reach a strongly connected spanning subgraph of $\Gamma_1(\mathscr{A})$, and then the graph $\Gamma_1(\mathscr{A})$ is strongly connected as well, or on some step the process gets stacked, which means that $\Gamma_1(\mathscr{A})$ has a proper strongly connected component, and therefore, is not strongly connected.

The described process branches, and in the worst case the number of words of defect 1 to be analyzed doubles at each step. On the other hand, since the steps are indexed by a chain of divisors of n, the number of steps does not exceed $\log_2 n + 1$. Thus, executing the algorithm, we have to analyze at most

$$1 + 2 + 4 + \cdots + 2^{\lceil \log_2 n \rceil + 1} = O(n)$$

words of maximum length $O(n \log_2 n)$, and therefore, the algorithm can be implemented in polynomial time.

4 Minimal Completely Reachable Automata

Syntactic complexity of a regular language is a well established concept that has attracted much attention lately, see, e.g., [6, 7]. It can be defined as the size of the transition monoid of the minimal DFA recognizing the

language. It appears to be worthwhile to extend this concept to automata by defining the *syntactic complexity* of an arbitrary DFA $\mathscr{A}$ as the size of its transition monoid $M(\mathscr{A})$. In fact, if one thinks of a DFA as a computational device rather than acceptor, its transition monoid can be thought of as the device's 'software library' since the monoid contains exactly all programs (transformations) that the automaton can execute. From this viewpoint, measuring the complexity of an automaton by the size of its 'software library' is fairly natural.

As already mentioned in Section 1, the syntactic complexity of a completely reachable automaton with n states cannot be less than $2^n - 1$. It turns out that this lower bound is tight if one considers automata over unrestricted alphabet. We present now a construction for completely reachable automata with n states and syntactic complexity $2^n - 1$; for short, we call them *minimal completely reachable automata*.

Our construction produces minimal completely reachable automata from full binary trees satisfying certain subordination conditions. Recall that a binary tree is said to be *full* if each its vertex v either is a leaf or has exactly two children that we refer to as the *left child* or the *son* of v and the *right child* or the *daughter* of v. (Thus, all vertices except the root have a gender.) It is well known (and easy to verify) that a full binary tree with n leaves has $2n - 1$ vertices. As full binary trees are the only trees occurring in this paper, we call them just trees in the sequel.

If Γ is a tree and v is a vertex in Γ, we denote by Γ_v the subtree of Γ rooted at v. The *span* of v, denoted $\mathrm{span}(v)$, is the number of leaves in the subtree Γ_v. Fig. 4 shows a tree with vertices labelled by their spans.

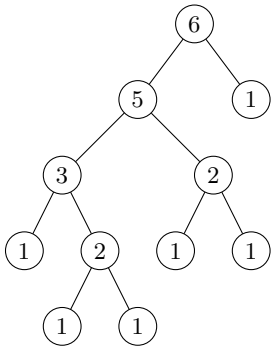

Fig. 4. An example of a tree with spans of its vertices shown

By a *homomorphism* between two trees Γ_1 and Γ_2 we mean a map from the vertex set of Γ_1 into the vertex set of Γ_2 that sends the root of Γ_1 to the root of Γ_2 and preserves the parent–child relation. Given two trees Γ_1 and Γ_2, we say that Γ_1 *subordinates* Γ_2 if there exists a 1-1 homomorphism $\xi\colon \Gamma_1 \to \Gamma_2$ such that for $\mathrm{span}(v) \le \mathrm{span}(v\xi)$ for every vertex v of Γ_1. If u and v are two vertices of the same tree Γ, we say that u *subordinates* v if the subtree Γ_u subordinates the subtree Γ_v. A tree is said to be *respectful* if it satisfies two conditions:

(S1) if a male vertex has a nephew, the nephew subordinates his uncle;
(S2) if a female vertex has a niece, the niece subordinates her aunt.

For an illustration, the tree shown in Fig. 4 satisfies (S1) but fails to satisfy (S2): the daughter of the root has a niece but this niece does not subordinates her aunt. On the other hand, the tree shown in Fig. 5 is respectful. (In order to ease the inspection of this claim, we have shown the uncle–nephew and the aunt–niece relations in this tree with dotted and dashed arrows respectively.)

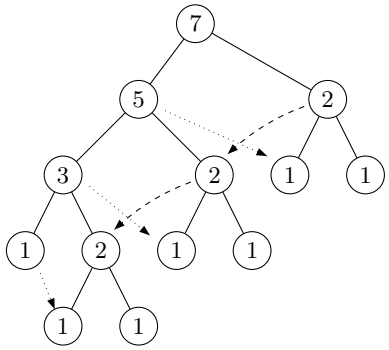

Fig. 5. An example of a respectful tree

It is easy to show that there exist respectful trees with any number of leaves. In the following table (borrowed from [4]) we present the numbers of respectful trees with up to 10 leaves.

Number of leaves	1	2	3	4	5	6	7	8	9	10
Number of respectful trees	1	1	2	3	6	10	18	32	58	101

We are not aware of any closed formula for the number of respectful trees with a given number of leaves.

In our construction, we use certain markings of trees by intervals of the set $\mathbb{N}$ of positive integers considered as a chain under the usual order:

$$1 < 2 < \cdots < n < \ldots .$$

If $i, j \in \mathbb{N}$ and $i \leq j$, the *interval* $[i, j]$ is the set $\{k \in X_n \mid i \leq k \leq j\}$. We write $[i]$ instead of $[i, i]$. By the *span* of an interval we mean the number of its elements. Now, a *faithful interval marking* of a tree Γ is a map μ from the vertex set of Γ into the set of all intervals in $\mathbb{N}$ such that for each vertex v,

- the span of the interval $v\mu$ is equal to $\mathrm{span}(v)$;
- if $v\mu = [i, j]$ and s and d are respectively the son and the daughter of v, then $s\mu = [i, k]$ and $d\mu = [k + 1, j]$ for some k such that $i \leq k < j$.

It easy to see that every tree Γ admits a faithful interval marking which is unique up to an additive translation: given any two markings μ, μ' of Γ, there is an integer m such that $v\mu = v\mu' + m$ for every vertex v. Observe that if μ is a faithful interval marking of a tree Γ and v is a vertex of Γ, then the restriction of μ to the subtree Γ_v is a faithful interval marking of the latter. Fig. 6 demonstrates a faithful interval marking of the tree from Fig. 5.

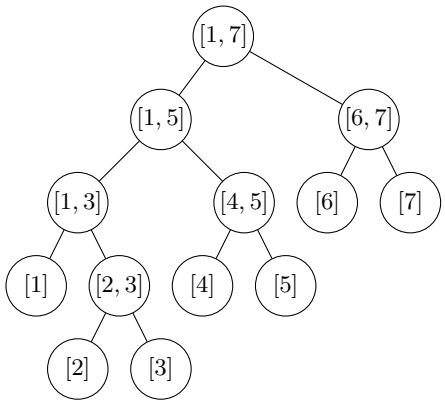

Fig. 6. A faithful interval marking of the tree from Fig. 5

We have prepared everything and can now present our construction.

Construction T2A (trees to automata). For each respectful tree Γ with n leaves and each its faithful interval marking μ, we construct an automaton denoted by $\mathscr{A}_\mu(\Gamma)$. The states of $\mathscr{A}_\mu(\Gamma)$ are the elements of

the interval $r\mu$, where r stands for the root of Γ, and the input alphabet of $\mathscr{A}_\mu(\Gamma)$ consists of $2n-2$ letters a_v, one for each non-root vertex v of Γ. To define the action of the letters, we proceed by induction on n. For $n=1$, that is, for the trivial tree Γ with one vertex r and no edges, $\mathscr{A}_\mu(\Gamma)$ is the trivial automaton with one state and no transitions, so that nothing has to be defined.

Now suppose that $n>1$. Take any non-root vertex v of Γ; we have to define the action of the letter a_v on the elements of the interval $r\mu$. If s and d are respectively the son and the daughter of r, the interval $r\mu$ is the disjoint union of $s\mu$ and $d\mu$. If $v\neq s$ and $v\neq d$, then v is a non-root vertex in one of the subtrees Γ_s or Γ_d. These two cases are symmetric, so that we may assume that v belongs to Γ_s. By the induction assumption applied to Γ_s and its marking induced by μ, the action of a_v is already defined on the states from the interval $s\mu$; we extend this action to the whole interval $r\mu$ by setting $y\,.\,a_v := y$ for each $y\in d\mu$.

It remains to define the action of the letters a_s and a_d. Again, by symmetry, it suffices to handle one of these cases, so that we define that action of a_s. If s has no nephew in Γ, then d is a leaf and $d\mu=[m]$ for some $m\in\mathbb{N}$. Then we let $x\,.\,a_s := m$ for each $x\in r\mu$. Otherwise let t be the nephew of s. The subordination condition (S1) implies that $\mathrm{span}(t)\leq\mathrm{span}(s)$. Let $s\mu=[i,j]$, $t\mu=[k,\ell]$. Then $h=(j-i)-(\ell-k)\geq 0$, and we define the action of a_s on $s\mu$ as follows:

$$\begin{gathered} i\,.\,a_s=(i+1)\,.\,a_s=\cdots=(i+h)\,.\,a_s := k,\\ (i+h+1)\,.\,a_s := k+1,\ \ldots,\ j\,.\,a_s := \ell. \end{gathered}$$

By the induction assumption applied to the subtree Γ_d and its marking induced by μ, the action of the letter a_t is already defined on the states from the interval $d\mu$; now we define the action of a_s on $d\mu$ by setting $y\,.\,a_s := y\,.\,a_t$ for all $y\in d\mu$. This completes our construction.

The reader may find it instructive to work out Construction T2A on a concrete example. For the tree from Fig. 5 and 6 used for illustrations above, computing all 12 input letters of the corresponding automaton would be rather cumbersome but one can check, for instance, that the letters a_s and a_d act on the set $[1,7]$ as follows:

$$a_s=\begin{pmatrix}1&2&3&4&5&6&7\\6&6&6&6&6&6&7\end{pmatrix}\quad a_d=\begin{pmatrix}1&2&3&4&5&6&7\\1&1&1&2&3&4&5\end{pmatrix}.$$

Those who prefer a complete example can look at the DFA $\mathscr{E}_3$ from Example 2: the automaton was in fact derived by Construction T2A from

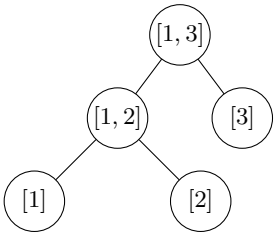

Fig. 7. The tree behind the automaton $\mathscr{E}_3$

the respectful tree with 3 leaves shown in Fig. 7. In particular, this explains our choice of notation for the input letters of $\mathscr{E}_3$ that perhaps had slightly puzzled the reader when she or he encountered this automaton in Section 2. By the way, the flip-flop in Fig. 3 also can be obtained by Construction T2A (from the unique tree with 2 leaves).

Observe that all automata constructed from different markings of the same respectful tree are isomorphic since passing to another marking only results in a change of the state names. Taking this into account, we omit the reference to μ in the notation and denote the automaton derived from any marking of a given respectful tree Γ simply by $\mathscr{A}(\Gamma)$.

We say that two DFAs $\mathscr{A} = \langle Q, \Sigma, \delta\rangle$ and $\mathscr{B} = \langle Q, \Delta, \zeta\rangle$ are *syntactically equivalent* if their transition monoids coincide. Now we are ready for the main result of this section.

Theorem 4. 1. *For each respectful tree Γ, the automaton $\mathscr{A}(\Gamma)$ is a minimal completely reachable automaton.*

2. *Every minimal completely reachable automaton is syntactically equivalent to an automaton of the form $\mathscr{A}(\Gamma)$ for a suitable respectful tree Γ.*

3. *Every minimal completely reachable automaton with n states has at least $2n - 2$ input letters.*

Claims 1 and 2 in Theorem 4 are essentially equivalent to the main results of the papers [3, 4] by the first author who has used a slightly different construction expressed in the language of transformation monoids: given a marking of a respectful tree Γ she constructs the transition monoid of $\mathscr{A}(\Gamma)$ rather than the automaton itself. Claim 3 is new but we have not included its proof here due to the space limitations because the only proof we have at the moment requires reproducing several concepts and results from [3, 4] and restating them in the language adopted in the present paper. It is very tempting to invent a direct proof of this claim that would bypass rather bulky considerations from [3, 4].

Theorem 4 leaves widely open the question about lower bounds for syntactic complexity of completely reachable automata with restricted alphabet. In particular, the case of completely reachable automata with 2 input letters both is of interest and seems to be tractable. The latter conclusion follows from our analysis of completely reachable automata with 2 input letters at the end of Section 3 which demonstrates that such DFAs have rather a specific structure.

We say that a DFA $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ *induces* a DFA $\mathscr{B} = \langle Q, \Delta, \zeta \rangle$ on the same state set if the transition monoid of $\mathscr{A}$ contains that of $\mathscr{B}$. Equivalently, this means that for every letter $b \in \Delta$, there exists a word $w \in \Sigma^*$ such that $\zeta(q, b) = \delta(q, w)$ for every $q \in Q$. This relation between automata plays an essential role in the theory of synchronizing automata, see, e.g., [2]. With respect to completely reachable automata, the following question is of interest: is it true that every completely reachable automaton induces a minimal completely reachable automaton? In other words, is it true that an automaton of the form $\mathscr{A}(\Gamma)$ 'hides' within every completely reachable automaton?

5 More Open Questions

Since completely reachable automata are synchronizing, it is natural to ask what is the maximum reset threshold for completely reachable automata with n states. In view of Example 1, the lower bound $(n-1)^2$ for this maximum is provided by the Černý automata $\mathscr{C}_n$. For completely reachable automata with 2 input letters this bound is tight because, except for the flip-flop, such automata have a letter that acts as a cyclic permutation of the state set, and therefore, Dubuc's result [10] applies to them. Some partial results about synchronization of completely reachable automata can be found in [9], but the general problem of finding the maximum reset threshold for completely reachable automata with n states and unrestricted alphabet remains open.

The problem discussed in the previous paragraph basically asks what is the minimum length of a word that reaches a singleton. For completely reachable automata, a similar question makes sense for an arbitrary nonempty subset. Thus, we suggest to investigate the minimum length of a word that reaches a subset with m element in a completely reachable automaton with n states as a function of n and m. Don [9, Conjecture 2] has formulated a very strong conjecture that implies the upper bound $n(n-m)$ on this length. Observe that if this upper bound indeed holds, then completely reachable automata satisfy the Černý conjecture. To see

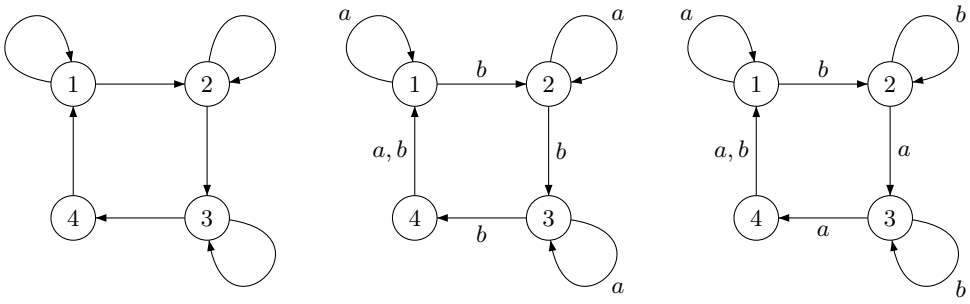

Fig. 8. A graph and two of its colorings

this, take a completely reachable automaton $\mathscr{A} = \langle Q, \Sigma \rangle$ with n states; it should possess a letter $a \in \Sigma$ such that $q \,.\, a = q' \,.\, a$ for two different states $q, q' \in Q$. If a word $w \in \Sigma^*$ of length at most $n(n-2)$ is such that $Q \,.\, w = \{q, q'\}$, the word wa is a reset word for $\mathscr{A}$ and has length at most $n(n-2) + 1 = (n-1)^2$.

Another intriguing problem about completely reachable automata suggested by the theory of synchronizing automata is a variant of the Road Coloring Problem. We recall notions involved there. A *road coloring* of a finite graph Γ consists in assigning non-empty sets of labels (colors) from some alphabet Σ to edges of Γ such that the label sets assigned to the outgoing edges of each vertex form a partition of Σ. Colored this way, Γ becomes a DFA over Σ; every such DFA is called a *coloring* of Γ. Fig. 8 shows a graph and two of its colorings by $\Sigma = \{a, b\}$, one of which is the Černý automaton $\mathscr{C}_4$. The Road Coloring Problem, recently solved by Trahtman [17], had asked which strongly connected graphs admit *synchronizing colorings*, i.e., colorings that are synchronizing automata. It turns out that, as it was conjectured in [1], the necessary and sufficient condition for a scn to possess a synchronizing coloring is that the greatest common divisor of lengths of all directed cycles in the graph should be equal to 1. The latter property is called *aperiodicity* or *primitivity*.

An analogous question makes sense for completely reachable automata. Namely, call a coloring of a graph *completely reachable* if it yields a completely reachable automaton. Our problem then consists in characterising graphs that admit completely reachable colorings. Such graphs must be strongly connected and primitive since every completely reachable automaton is strongly connected and synchronizing. However, it is easy to produce an example of a strongly connected primitive graph that has no completely reachable coloring; such a graph is shown in Fig. 9 on the left. Moreover, there are interesting phenomena that have no parallel in the theory of synchronizing automata; for instance, there exist graphs that

have no completely reachable coloring with 2 letters but admit such a coloring with 3 letters; an example of such a graph is presented in the center of Fig. 9 while the corresponding coloring is shown on the right.

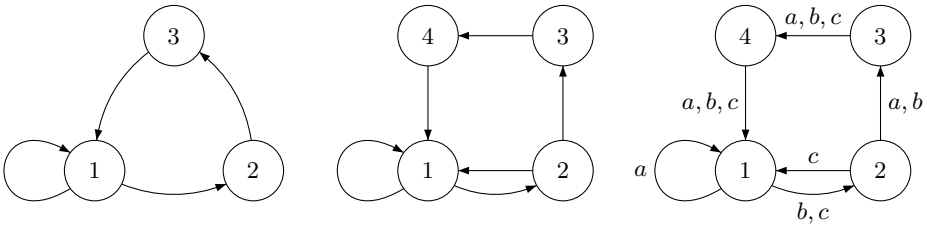

Fig. 9. The left graph has no completely reachable coloring; the central graph has no completely reachable coloring with 2 letters but has a completely reachable coloring with 3 letters shown in the right

References

1. Adler, R.L., Goodwyn, L.W., Weiss, B.: Equivalence of topological Markov shifts. Israel J. Math. 27, 49–63 (1977)
2. Ananichev, D.S., Gusev, V.V., Volkov, M.V.: Primitive digraphs with large exponents and slowly synchronizing automata. J. Math. Sci. 192(3), 263–278 (2013)
3. Bondar, E.: $\mathcal{L}$-cross-sections of the finite symmetric semigroup. Algebra and Discrete Math. 18(1), 27–41 (2014)
4. Bondar, E.: Classification of $\mathcal{L}$-cross-sections of $\mathcal{T}_n$. Algebra and Discrete Math. 21(1), 1–17 (2016)
5. Brandl, Ch., Simon, H.U., Complexity analysis: transformation monoids of finite automata. In: I. Potapov (ed.), Developments in Language Theory—19th Int. Conf., DLT 2015. Lect. Notes Comput. Sci., vol. 9168, pp. 143–154. Springer, Heidelberg (2015)
6. Brzozowski, J.A., Li, B.: Syntactic complexity of $\mathcal{R}$ and $\mathcal{J}$-trivial regular languages. Int. J. Found. Comput. Sci. 25(7): 807–822 (2014)
7. Brzozowski, J.A., Szykuła, M.: Upper bound on syntactic complexity of suffix-free languages. In: J. Shallit, A. Okhotin (eds.), Descriptional Complexity of Formal Systems—17th Int. Workshop, DCFS 2015. Lect. Notes Comput. Sci., vol. 9118, pp. 33–45. Springer, Heidelberg (2015)
8. Černý, J.: Poznámka k homogénnym eksperimentom s konečnými automatami. Matematicko-fyzikalny Časopis Slovensk. Akad. Vied 14(3), 208–216 (1964) (in Slovak)
9. Don, H.: The Černý conjecture and 1-contracting automata. CoRR, abs/1507.06070 (2015) `http://arxiv.org/abs/1507.06070`
10. Dubuc, L.: Sur les automates circulaires et la conjecture de Černý. RAIRO Inform. Théor. Appl. 32(1-3), 21–34 (1998) (in French)

11. Furst, M.L., Hopcroft, J.E., Luks, E.M.: Polynomial-time algorithms for permutation groups. In: 21st Annual Symp. on Foundations of Comput. Sci., pp. 36–41. IEEE Computer Society, Washington (1980)
12. Ganyushkin, O., Mazorchuk, V. Classical Finite Transformation Semigroups: An Introduction. Springer, Heidelberg (2009)
13. Goralčík, P., Koubek, V.: Rank problems for composite transformations. Int. J. Algebra Comput. 5(3), 309–316 (1995)
14. Kozen, D.: Lower bounds for natural proof systems. In: 18th Annual Symp. on Foundations of Comput. Sci., pp. 254–266. IEEE Computer Society, Washington (1977)
15. Maslennikova, M.I.: Reset complexity of ideal languages. In: M. Bieliková, G. Friedrich, G. Gottlob, S. Katzenbeisser, R. Špánek, G. Turán (eds.), 38th Int. Conf. on Current Trends in Theory and Practice of Comput. Sci., SOFSEM 2012. Vol. II, pp. 33–44. Inst. Comp. Sci. Acad. Sci. Czech Republic, Prague (2012)
16. Maslennikova, M.I.: Reset complexity of ideal languages. CoRR, abs/1404.2816 (2014) http://arxiv.org/abs/1404.2816
17. Trahtman, A.N.: The road coloring problem. Israel J. Math. 172, 51–60 (2009)
18. Volkov, M.V.: Synchronizing automata and the Černý conjecture. In: C. Martín-Vide, F. Otto, H. Fernau (eds.), Languages and Automata Theory and Applications—2nd Int. Conf., LATA 2008. Lect. Notes Comput. Sci., vol. 5196, pp. 11–27. Springer, Heidelberg (2008)