Subset Synchronization of Transitive Automata[*]

Vojtěch Vorel
Charles University in Prague, Czech Republic
vorel@ktiml.mff.cuni.cz

We consider the following generalized notion of synchronization: A word is called a reset word of a subset of states of a deterministic finite automaton if it maps all states of the set to a unique state. It is known that the minimum length of such words is superpolynomial in worst cases, namely in a series of substantially nontransitive automata. We present a series of transitive binary automata with a strongly exponential minimum length. This also constitutes a progress in the research of composition sequences initiated by Arto Salomaa, because reset words of subsets are just a special case of composition sequences. Deciding about the existence of a reset word for given automaton and subset is known to be a PSPACE-complete problem, we prove that this holds even if we restrict the problem to transitive binary automata.

1 Introduction

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

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

for each $S \subseteq Q$. An automaton (Q, X, δ) is said to be *transitive* if

$$(\forall r, s \in Q)\,(\exists w \in X^{\star})\,\delta(r,w) = s.$$

A state $s \in Q$ is a *sink state* if

$$(\forall x \in X)\,\delta(s,x) = s.$$

Clearly, if a nontrivial automaton has some sink state, it is impossible for the automaton to be transitive. 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.

The *Černý conjecture*, a longstanding open problem, claims that each synchronizing automaton has a reset word of length at most $(|Q|-1)^2$. There is a series of automata due to Černý that reaches this bound [3], but all known upper bounds lie in $\Omega\left(|Q|^3\right)$, see [15] for the best one[1]. A tight bound has been established for various special classes of automata, see a survey in [23] or some recent advances e.g. in [1, 6, 8, 20].

[*]Research supported by the Czech Science Foundation grant GA14-10799S.
[1]An improved bound published by Trakhtman [22] in 2011 has turned out to be proved incorrectly.

1.1 Synchronization of Subsets

Even if an automaton is not synchronizing, there could be various subsets $S \subseteq Q$ such that $|\delta(S,w)| = 1$ for some word $w \in X^{\star}$. We say that such S is *synchronizable* in A and in the opposite case we say it is *blind* in A. The word w is called a *reset word of* S in A. Such words are of our interest. They lack some of elegant properties of classical reset words (i.e. reset words of all Q), particularly a word w having a factor v which is a reset word of S need not to be itself a reset word of S. In fact, if we choose a subset S and a word w, it is possible for the set $\delta(S,w)$ to be blind even if the set S was synchronizable.

Suppose $A = (Q,X,\delta)$ and $S \subseteq Q$. We denote by $\mathrm{CS}(A,S)$ the length of the shortest reset word of S in A. If S is blind, we set $\mathrm{CS}(A,S) = 0$. Let $\mathcal{M}$ be a class of automata. For each n let $\mathcal{M}_{\leq n}$ be the class of all automata lying in $\mathcal{M}$ and having at most n states. We denote

$$\mathrm{CS}_n^{\mathcal{M}} \quad = \max_{\substack{A \in \mathcal{M}_{\leq n} \\ S \subseteq Q}} \mathrm{CS}(A,S).$$

If $\mathcal{M}$ is the class of *all* automata, we write just CS_n instead of $\mathrm{CS}_n^{\mathcal{M}}$.

Such values we informally call *subset synchronization thresholds*. The class of all transitive automata and the class of all automata with a k-letter alphabet are denoted by $\mathcal{TR}$ and $\mathcal{AL}_k$ respectively. Automata from $\mathcal{AL}_2$ are called *binary*.

As we describe below, it was proven independently by [10] and [17] that $\mathrm{CS}_n \geq (\sqrt[3]{n})!$, and a construction from [12] implies that $\mathrm{CS}_n \geq 2^{\Omega(n)}$, but the proofs use automata with multiple sink states and growing alphabets. Use of sink states is a very strong tool for designing automata having given properties, but in practice such automata seem very special. They represent unstable systems balancing between different deadlocks. The very opposite are the transitive automata. Does the threshold remain so high if we consider only transitive automata? Unfortunately, we show below that it does, even if we restrict the alphabet size to a constant. We prove that

$$\mathrm{CS}_n^{\mathcal{AL}_2 \cap \mathcal{TR}} = 2^{\Omega(n)},$$

which substantially raises also the general lower bounds of each $\mathrm{CS}_n^{\mathcal{AL}_k}$, because their former lower bounds (following from [12]) lie in $2^{o(n)}$. The new bound is tight since $\mathrm{CS}_n = 2^{\mathcal{O}(n)}$.

1.2 Minimum Length of Compositions

It has been repeatedly pointed out by Arto Salomaa [17, 18] that very little is known about minimum length of a composition needed to generate a function by a given set of generators. To be more precise, let us adopt and slightly extend the notation used in [17, 18]. We denote by $\mathcal{T}_n$ the semigroup of all functions from $\{1,\ldots,n\}$ to itself. Given $\mathbf{G} \subseteq \mathcal{T}_n$, we denote by $\langle \mathbf{G} \rangle$ the subsemigroup generated by $\mathbf{G}$. Given $\mathbf{F} \subseteq \mathcal{T}_n$ we denote by $\mathrm{D}(\mathbf{G},\mathbf{F})$ the length k of a shortest sequence $g_1,\ldots,g_k$ of functions from $\mathbf{G}$ such that $g_1 \ldots g_k \in \mathbf{F}$. Finally, denote

$$\mathrm{D}_n = \max_{\overline{n} \leq n} \max_{\substack{\mathbf{F},\mathbf{G} \subseteq \mathcal{T}_{\overline{\mathbf{n}}} \\ \mathbf{F} \cap \langle \mathbf{G} \rangle \neq \emptyset}} \mathrm{D}(\mathbf{G},\mathbf{F}). \tag{1}$$

From the well-known connection between automata and transformation semigroups it follows that the value CS_n could be also defined by (1) if we just restrict $\mathbf{F}$ to be some of the sets

$$\mathbf{F}_S = \{f \in \mathcal{T}_n \mid (\forall r,s \in S)\, f(r) = f(s)\}$$

for $S \subseteq \{1,\ldots,n\}$. Therefore it holds trivially that

$$\mathrm{D}_n \geq \mathrm{CS}_n\,.$$

Arto Salomaa refers to a single nontrivial bound of D_n, namely $\mathrm{D}_n \geq (\sqrt[3]{n})!$, which is a consequence of the above-mentioned variant for CS_n. In fact he omits a much older construction of Kozen [9, Theorem 3.2.7] which deals with lengths of *proofs* rather than compositions but witnesses easily that $\mathrm{D}_n = 2^{\Omega\left(\frac{n}{\log n}\right)}$. Since 2013 it follows from [12] that $\mathrm{D}_n = 2^{\Omega(n)}$. Our result shows that this lower bound holds also if we restrict $\mathbf{G}$ to any nontrivial fixed size.

In Group Theory, thresholds like D_n are studied in the scope of permutations, see [7].

2 Lower Bounds of Subset Synchronization Thresholds

We first formulate the two former lower bounds of CS_n. Let p_i stand for the i-th prime.

Theorem 1 ([10, 17]). *For each k there is an automaton $A_k = (Q_k, \{a,b\}, \delta_k)$ and a subset S_k such that $|Q_k| = 2 + \sum_{i=1}^{k} p_i$ and $\mathrm{CS}(A_k,S_k) = \prod_{i=1}^{k} p_i$.*

The proof of the theorem uses an automaton A_k that consists of k cyclic parts of prime sizes and two sink states, so it is essentially non-transitive. The theorem implies that $\mathrm{CS}_n \geq (\sqrt[3]{n})!$, because $\prod_{i=1}^{k} p_i \geq k!$ and $|Q_k| \leq k^3$, using the estimation $p_i \leq i^2$. By the terminology of [10] such bound is *exponential*, but using canonical estimations of p_i it is not hard to show that the bound is exceeded by $n \mapsto \varepsilon^n$ for any $\varepsilon > 1$.

Theorem 2 ([12]). *It holds that $\mathrm{CS}_n = 2^{\Omega(n)}$ and $\mathrm{CS}_n^{\mathcal{AL}_2} = 2^{\Omega\left(\frac{n}{\log n}\right)}$.*

The paper [12] studies *careful synchronization* of partial automata, but the lower bounds can be adapted for our setting. The proofs of [12, Theorem 1] and [12, Theorem 3] can be modified (by adding one state) so that the constructed automata have sink states. Then we can add another sink state D which becomes the target of all undefined transitions. Then all reset words for the subset $Q \backslash \{\mathrm{D}\}$ are careful reset words of the original partial automaton and we can use the corresponding lower bounds.

Let us introduce three key methods used in the present paper. The first is quite simple and has been already used in the literature [2]. It modifies an automaton in order to decrease the alphabet size with preserving high synchronization thresholds:

Lemma 3. *For each automaton $A = (Q,X,\delta)$ and $S \subseteq Q$ there is an automaton $A' = (Q',X',\delta')$ and $S' \subseteq Q'$ such that*

1. *S is synchronizable in $A \Rightarrow S'$ is synchronizable in A'*
2. $\mathrm{CS}(A',S') \geq \mathrm{CS}(A,S)$
3. $|Q'| = |Q| \cdot |X|$
4. $|X'| = 2$
5. *A' and A have equal number of strongly connected components*

Proof. Suppose that $X = \{a_0,\ldots,a_m\}$. We set $Q' = Q \times X$, $X' = \{\alpha,\beta\}$,

$$\begin{aligned}\delta'((s,a_i),\alpha) &= (\delta(s,a_i),a_0)\\ \delta'((s,a_i),\beta) &= (s,a_{i+1 \bmod m})\,.\end{aligned}$$

Informally, a transition $r \xrightarrow{a_i} s$ of A is simulated in A' by

$$(r,a_0) \xrightarrow{\beta} (r,a_1) \xrightarrow{\beta} \dots \xrightarrow{\beta} (r,a_i) \xrightarrow{\alpha} (s,a_0)\,.$$

If we set $S' = S \times \{a_0\}$, it is not hard to see that any reset word of S' in A' have to be of the form $\left(\beta^{i_1}\alpha\right)\dots\left(\beta^{i_d}\alpha\right)$ for some $w = a_{i_1}\dots a_{i_d}$ which is a reset word of S in A. ☐

The second method is original and is intended for modifying an automaton to be transitive, again with high synchronization thresholds preserved. It relies on the following concept:

Definition 4. Let $A = (Q,X,\delta)$ be an automaton and let $\rho \subseteq Q^2$ be a congruence, i.e. equivalence relation satisfying $r\rho s \Rightarrow \delta(r,x)\,\rho\,\delta(s,x)$ for each $x \in X$. We say that ρ is a *swap congruence* if, for each equivalence class C of ρ and each letter $x \in X$, the restricted function $\delta(\_,x) : C \to Q$ is injective.

Let us express the key feature of swap congruences and use it in the construction.

Lemma 5. *Let $A = (Q,X,\delta)$ be an automaton, let $\rho \subseteq Q^2$ be a swap congruence and take any $S \subseteq Q$. If there are any $r,s \in S$ with $r \neq s$ and $r\rho s$, the set S is blind.*

Proof. Because r and s lie in a common equivalence class of ρ, by the definition of swap congruence we have $\delta(r,x) \neq \delta(s,x)$ for any $x \in X$. It follows that each set $\delta(S,w)$ for $w \in X^\star$ is of size at least 2. ☐

Lemma 6. *For each automaton $A = (Q,X,\delta)$ and $S \subseteq Q$ there is an automaton $A' = (Q',X',\delta')$ and $S' \subseteq Q'$ such that*

1. *S is synchronizable in $A \Rightarrow S'$ is synchronizable in A'*
2. $\mathrm{CS}(A',S') \geq \mathrm{CS}(A,S)$
3. *A' is transitive*
4. $|Q'| = 4|Q| + 2$
5. $|X'| = |X| + 2c$

where c is the number of strongly connected components of A.

Proof. Let $C_1,\dots,C_c$ be the strongly connected components of A. Fix some $q_i \in C_i$ for each i. We set

$$\begin{aligned} Q' &= \left\{E,\overline{E}\right\} \cup \left(\left\{1,\overline{1},2,\overline{2}\right\} \times Q\right) \\ X' &= X \cup \{a_i,b_i \mid i = 1,\dots,c\} \end{aligned}$$

and define the transition function δ' as follows. If we omit all the letters a_s and b_s from the alphabet X', we find the states $E,\overline{E}$ isolated (i.e. $E \xrightarrow{x} E, \overline{E} \xrightarrow{x} \overline{E}$ for $x \in X$) and the rest of A' consisting just of four copies of A:

$$(N,s) \quad \xrightarrow{x} \quad (N,\delta(s,x))$$

for $N \in \left\{1,\overline{1},2,\overline{2}\right\}, s \in Q, x \in X$. Let us introduce the additional letters. For any $i \in 1,\dots,c$ and $s \in Q$ such that $s \neq q_i$ we set

$$\begin{aligned} (1,s) &\xrightarrow{a_i,b_i} E \qquad & \left(\overline{1},s\right) &\xrightarrow{a_i,b_i} \overline{E} \\ (2,s) &\xrightarrow{a_i,b_i} E \qquad & \left(\overline{2},s\right) &\xrightarrow{a_i,b_i} \overline{E} \end{aligned}$$

and it remains to see Figure 1, which describes for each $i \in 1,\dots,c$ the action of a_i and b_i on the six states $E, \overline{E}, (1,q_i), (\overline{1},q_i), (2,q_i), (\overline{2},q_i)$.

Observe that the equivalence ρ having the classes $\{E,\overline{E}\}$ and $\{(1,s),(\overline{1},s)\}, \{(2,s),(\overline{2},s)\}$ for each $s \in Q$ is a swap congruence of A'. We claim that the automaton A' and the set

$$S' = \{1\} \times S \cup \{\overline{2}\} \times S$$

fulfill our requirements on synchronizability, the synchronization threshold and transitivity:

- If the set S is synchronizable in A, there are $r \in Q$ and $w \in X^\star$ such that $\delta(S,w) = \{r\}$. The state r lies in some C_i, so there is a word $u \in X^\star$ such that $\delta(r,u) = q_i$. We claim that the word wua_i synchronizes S' in A'. Indeed, we have $\delta'(S',w) = \{(1,r),(\overline{2},r)\}$ and therefore $\delta'(S',wu) = \{(1,q_i),(\overline{2},q_i)\}$ and $\delta'(S',wua_i) = \{(1,q_i)\}$.
- Let S' be synchronized in A' by a word $w \in (X')^\star$. There must occur some letter a_r or b_r in w, because S' contains states from two different copies of A. Thus we can write

$$w = uxv$$

for some $u \in X^\star$, $i \in 1,\dots,c$, $x \in \{a_i,b_i\}$ and $v \in (X')^\star$. If $\delta'(S',u)$ contains unique state from $\{1\} \times Q$, the word u synchronizes S in A, we are done. Otherwise there is some state $(1,s) \in \delta'(S',u)$ such that $s \neq q_i$. Because $u \in X^\star$, it holds also that $(\overline{2},s) \in \delta'(S',u)$. But

$$(1,s) \xrightarrow{a_i,b_i} E \qquad (\overline{2},s) \xrightarrow{a_i,b_i} \overline{E},$$

so the blind subset $\{E,\overline{E}\}$ is contained in $\delta'(S',ux)$, which is a contradiction.
- In order to verify that A' is transitive we first find a path between any pair of distinct states (N,s),(N,r) from a common copy of A. Let $r \in C_i$ and $\delta(q_i,u) = r$. If $s = q_i$, the path is labeled by u. Otherwise we have:

$$\begin{array}{llll} (1,s) \xrightarrow{a_i} E \xrightarrow{a_i} (1,q_i) \xrightarrow{u} (1,r) & & (\overline{1},s) \xrightarrow{a_i} \overline{E} \xrightarrow{a_i} (\overline{1},q_i) \xrightarrow{u} (\overline{1},r) \\ (2,s) \xrightarrow{a_i} E \xrightarrow{b_i} (2,q_i) \xrightarrow{u} (2,r) & & (\overline{2},s) \xrightarrow{a_i} \overline{E} \xrightarrow{b_i} (\overline{2},q_i) \xrightarrow{u} (\overline{2},r). \end{array}$$

The paths above also guarantee that there are no more than two strongly connected components:

$$C = \{E\} \cup \{1,2\} \times Q, \qquad \overline{C} = \{\overline{E}\} \cup \{\overline{1},\overline{2}\} \times Q.$$

It remains to connect C with $\overline{C}$: For any i we have $(1,q_i) \xrightarrow{b_i} (\overline{2},q_i) \xrightarrow{a_i} (1,q_i)$.

□

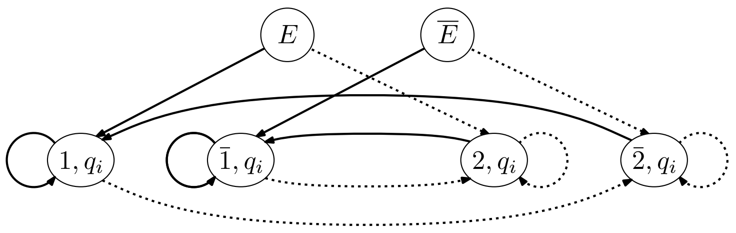
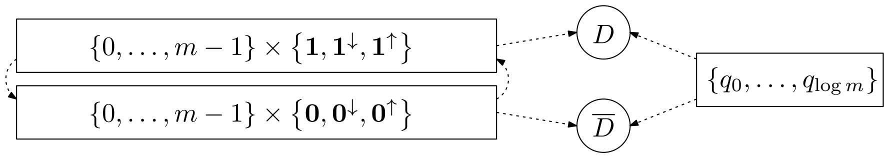


Figure 1: Action of the letters a_i (solid arrows) and b_i (dotted arrows) on certain states of A'.

Let us present the main construction of the present paper, a series of automata with strictly exponential subset synchronization threshold, constant alphabet size and constant number of strongly connected components. We use some informal principles that occur in [12] as well.



$\{0,\dots,m-1\}\times\{\mathbf{1},\mathbf{1}^{\downarrow},\mathbf{1}^{\uparrow}\}$
$\{0,\dots,m-1\}\times\{\mathbf{0},\mathbf{0}^{\downarrow},\mathbf{0}^{\uparrow}\}$
D
$\overline{D}$
$\{q_0,\dots,q_{\log m}\}$

Figure 2: Main parts of A. The arrows depict the connectivity pattern of A.

Lemma 7. *For infinitely many $m\in\mathbb{N}$ there is an automaton $A=(Q,X,\delta)$ and $S\subseteq Q$ such that*

1. $\mathrm{CS}(A,S)=2^m(\log m+1)+1$
2. $|Q|=6m+\log m+3$
3. $|X|=4$
4. A has 4 strongly connected components.

Proof. Suppose $m=2^k$. For each $t\in 0,\dots,m-1$ we denote by $\tau=\mathrm{bin}(t)$ the standard k-digit binary representation of t. By a classical result proved in [5] there is a *De Bruijn sequence* $\xi=\xi_0\dots\xi_{m-1}$ consisting of binary letters $\xi_i\in\{\mathbf{0},\mathbf{1}\}$ such that each word $\tau\in\{\mathbf{0},\mathbf{1}\}^k$ appears exactly once as a cyclic factor of ξ (i.e. it is a factor or begins by a suffix of ξ and continues by a prefix of ξ). Let us fix such ξ. By $\pi(i)$ we denote the number t, whose binary representation $\mathrm{bin}(t)$ starts in ξ from the i-th position. Note that π is a permutation of $\{0,\dots,m-1\}$. Set

$$\begin{aligned} Q &= \left(\{0,\dots,m-1\}\times\left\{\mathbf{0},\mathbf{0}^{\downarrow},\mathbf{0}^{\uparrow},\mathbf{1},\mathbf{1}^{\downarrow},\mathbf{1}^{\uparrow}\right\}\right)\cup\{q_0,\dots,q_{\log m},D,\overline{D}\}\\ X &= \{\mathbf{0},\mathbf{1},\kappa,\omega\}\\ S &= (\{0,\dots,m-1\}\times\{\mathbf{0}\})\cup\{q_0,D\}. \end{aligned}$$

Figure 2 visually distinguishes main parts of the automaton. The states D and $\overline{D}$ are sinks. Together with $D\in S$ it implies that any reset word of S takes the states of S to D and that the state $\overline{D}$ must not become active during the synchronization (i.e. lie in $\delta(S,v)$ for a prefix v of a reset word). The states $\{q_0,\dots,q_{\log m}\}$ guarantee that any reset word of S lies in

$$\left(\{\mathbf{0},\mathbf{1}\}^k\kappa\right)^{\star}\omega X^{\star}. \tag{2}$$

Indeed, as defined by Figure 3, any other word takes q_0 to $\overline{D}$. Let the letter ω act as follows:

$$\begin{aligned} \{0,\dots,m-1\}\times\{\mathbf{1}\}\ ,\ q_0\ ,\ D &\xrightarrow{\omega} D\\ \{0,\dots,m-1\}\times\left\{\mathbf{0},\mathbf{0}^{\downarrow},\mathbf{0}^{\uparrow},\mathbf{1}^{\downarrow},\mathbf{1}^{\uparrow}\right\}\ ,\ q_1,\dots,q_{\log m}\ ,\ \overline{D} &\xrightarrow{\omega} \overline{D}. \end{aligned}$$

We see that ω maps each state to D or $\overline{D}$. This implies that once ω occurs in a reset word of S, it must complete the synchronization. In order to map q_0 to D, the letter ω *must* occur, so any shortest reset word of S is exactly of the form

$$w=(\tau_1\kappa)\dots(\tau_d\kappa)\,\omega, \tag{3}$$

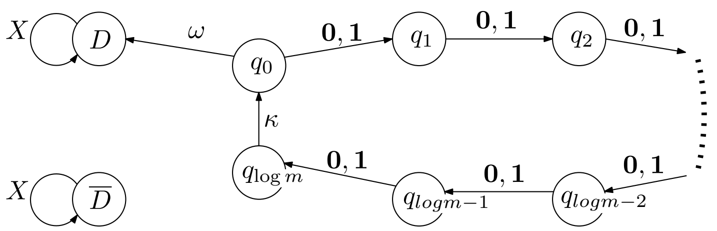


Figure 3: A part of A. Except for the depicted ones, all transitions here lead to $\overline{D}$.

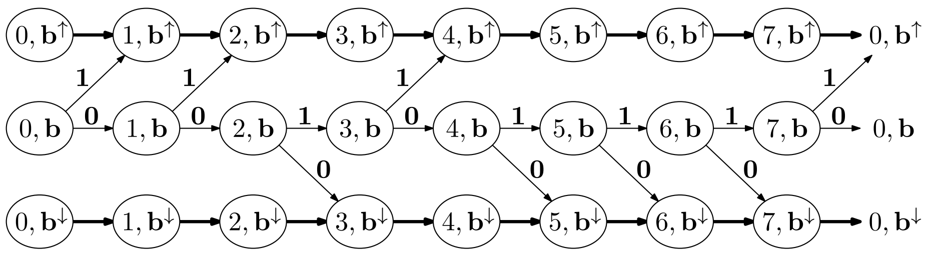
where $\tau_j \in \{\mathbf{0},\mathbf{1}\}^k$ for each j.

The two biggest parts depicted by Figure 2 contain $3m$ states each and are the same up to the letters κ and ω. On both of them (take $\mathbf{b} \in \{\mathbf{0},\mathbf{1}\}$) let the letters $\mathbf{0}$ and $\mathbf{1}$ act as follows:

$$(i,\mathbf{b}) \xrightarrow{\mathbf{0}} \begin{cases} (i+1,\mathbf{b}) & \text{if } \xi_i = \mathbf{0} \\ (i+1,\mathbf{b}^{\downarrow}) & \text{if } \xi_i = \mathbf{1} \end{cases} \qquad (i,\mathbf{b}) \xrightarrow{\mathbf{1}} \begin{cases} (i+1,\mathbf{b}^{\uparrow}) & \text{if } \xi_i = \mathbf{0} \\ (i+1,\mathbf{b}) & \text{if } \xi_i = \mathbf{1} \end{cases}$$

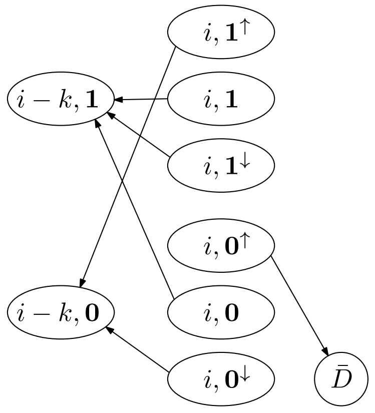
$$\left(i,\mathbf{b}^{\uparrow}\right) \xrightarrow{\mathbf{0},\mathbf{1}} \left(i+1,\mathbf{b}^{\uparrow}\right) \qquad \left(i,\mathbf{b}^{\downarrow}\right) \xrightarrow{\mathbf{0},\mathbf{1}} \left(i+1,\mathbf{b}^{\downarrow}\right)$$

where we perform the addition modulo m. For example, Figure 4 depicts such part of A for $m = 8$ and a particular De Bruijn sequence ξ. Figure 5 defines the action of κ on the states $\{i\} \times \left\{\mathbf{0},\mathbf{0}^{\downarrow},\mathbf{0}^{\uparrow},\mathbf{1},\mathbf{1}^{\downarrow},\mathbf{1}^{\uparrow}\right\}$ for any i, so the automaton A is completely defined.



$0,\mathbf{b}^{\uparrow}$
$1,\mathbf{b}^{\uparrow}$
$2,\mathbf{b}^{\uparrow}$
$3,\mathbf{b}^{\uparrow}$
$4,\mathbf{b}^{\uparrow}$
$5,\mathbf{b}^{\uparrow}$
$6,\mathbf{b}^{\uparrow}$
$7,\mathbf{b}^{\uparrow}$
$0,\mathbf{b}^{\uparrow}$
$0,\mathbf{b}$
$1,\mathbf{b}$
$2,\mathbf{b}$
$3,\mathbf{b}$
$4,\mathbf{b}$
$5,\mathbf{b}$
$6,\mathbf{b}$
$7,\mathbf{b}$
$0,\mathbf{b}$
$0,\mathbf{b}^{\downarrow}$
$1,\mathbf{b}^{\downarrow}$
$2,\mathbf{b}^{\downarrow}$
$3,\mathbf{b}^{\downarrow}$
$4,\mathbf{b}^{\downarrow}$
$5,\mathbf{b}^{\downarrow}$
$6,\mathbf{b}^{\downarrow}$
$7,\mathbf{b}^{\downarrow}$
$0,\mathbf{b}^{\downarrow}$

Figure 4: A part of A assuming $m = 8$ and $\xi = \mathbf{00101110}$. Bold arrows represent both $\mathbf{0},\mathbf{1}$.



$i,\mathbf{1}^{\uparrow}$
$i-k,\mathbf{1}$
$i,\mathbf{1}$
$i,\mathbf{1}^{\downarrow}$
$i,\mathbf{0}^{\uparrow}$
$i-k,\mathbf{0}$
$i,\mathbf{0}$
$i,\mathbf{0}^{\downarrow}$
$\bar{D}$

Figure 5: Action of the letter κ. The subtraction is modulo m.

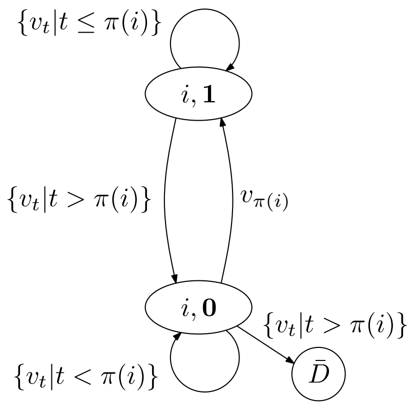


Figure 6: Action of the words $v_0,\ldots,v_{m-1}$ on the i-th switch.

Let w be a shortest reset word of S in A. It is necessarily of the form (3), so it makes sense to denote $v_t = \mathrm{bin}(t)\,\kappa$ and treat w as a word

$$w = v_{t_1} \ldots v_{t_d} \omega \in \{v_0, \ldots, v_{m-1}, \omega\}^{\star}. \tag{4}$$

The action of each v_t is depicted by Figure 6. It is a key step of the entire proof to confirm that Figure 6 is correct. Indeed:

- Starting from a state $(i, \mathbf{0})$, a word $\mathrm{bin}(t)$ takes us through kind of decision tree to one of the states $\left(i+k, \mathbf{0}^{\downarrow}\right), (i+k, \mathbf{0}), \left(i+k, \mathbf{0}^{\uparrow}\right)$, depending on whether t is lesser, equal, or greater than $\pi(i)$ respectively. This is guaranteed by wiring the sequence ξ into the transition function, see Figure 4. The letter κ then take us back to $\{i\} \times \{\ldots\}$, namely to $(i, \mathbf{0})$ or $(i, \mathbf{1})$, or we fall to $\overline{D}$ (respectively).
- Starting from a state $(i, \mathbf{1})$, we proceed similarly and end up in $(i, \mathbf{0})$ or $(i, \mathbf{1})$ depending on whether t is greater than $\pi(i)$ or not.

It follows that after applying any prefix $v_{t_1} \ldots v_{t_j}$ of w exactly one of the states $(i, \mathbf{0}), (i, \mathbf{1})$ is active for each i. We say that *the i-th switch is set to* $\mathbf{0}$ *or* $\mathbf{1}$ *in time* j. Observe that in time d all the switches are set to $\mathbf{1}$, because otherwise the state $\overline{D}$ would become active by the application of ω. On the other hand, in time 0 all the switches are set to $\mathbf{0}$. We are going to show that in fact during the synchronization of S the switches together perform a binary counting from 0 (all the switches set to $\mathbf{0}$) to $2^m - 1$ (all the switches set to $\mathbf{1}$). For each i the significance of i-th switch is given by the value $\pi(i)$. So the $\pi^{-1}(m-1)$-th switch carries the most significant digit, the $\pi^{-1}(0)$-th switch carries the least significant digit and so on. The number represented in this manner by the switches in time j is denoted by $\mathfrak{b}_j \in \{0, \ldots, 2^m - 1\}$. We claim that $\mathfrak{b}_j = j$ for each j. Indeed:

- In time 0, all the switches are set to $\mathbf{0}$, we have $\mathfrak{b}_0 = 0$.
- Suppose that $\mathfrak{b}_{j'} = j'$ for each $j' \leq j-1$. We denote

$$\overline{t_j} = \min\{\pi(i) \mid i\text{-th switch is set to } \mathbf{0} \text{ in time } j-1\} \tag{5}$$

  and claim that $t_j = \overline{t_j}$. Note that $\overline{t_j}$ is defined to be the least significance level at which there occurs a $\mathbf{0}$ in the binary representation of $\mathfrak{b}_{j-1}$. Suppose for a contradiction that $t_j > \overline{t_j}$. By the definition of $\overline{t_j}$ the state $\left(\pi^{-1}(\overline{t_j}), \mathbf{0}\right)$ lies in $\delta\left(S, v_{t_1} \ldots v_{t_{j-1}}\right)$. But v_{t_j} takes this state to $\overline{D}$, which is a contradiction. Now suppose that $t_j < \overline{t_j}$. In such case the application of v_{t_j} does not turn any switch from $\mathbf{0}$ to $\mathbf{1}$, so $\mathfrak{b}_j \leq \mathfrak{b}_{j-1}$ and thus in time j the configuration of switches is the same at it was in time $\mathfrak{b}_j$. This contradicts the assumption that w is a shortest reset word. We have proved that $t_j = \overline{t_j}$ and it remains only to show that the application of v_{t_j} performs an addition of 1 and so makes the switches represent the value $\mathfrak{b}_{j-1} + 1$.

  - Consider an i-th switch with $\pi(i) < t_j$. By the definition of $\overline{t_j}$ it is set to $\mathbf{1}$ in time $j-1$ and the word v_{t_j} set it to $\mathbf{0}$ in time j. This is what we need because such switches represent a continuous segment of $\mathbf{1}$s at the least significant positions of the binary representation of $\mathfrak{b}_{j-1}$.
  - The $\pi^{-1}(t_j)$-th switch is set from $\mathbf{0}$ to $\mathbf{1}$ by the word v_{t_j}.
  - Consider an i-th switch with $\pi(i) > t_j$. The switch represents a digit of $\mathfrak{b}_{j-1}$ which is more significant than the $\overline{t_j}$-th digit. As we expect, the word v_{t_j} leave such switch unchanged.

Because $\mathfrak{b}_d = 2^m$, we deduce that $d = 2^m$ and thus $|w| = 2^m(\log m + 1) + 1$ if such (shortest) reset word exists. But in fact we have also shown that there is only one possibility for such w and that it is a true reset word for S: The unique w is of the form (4), where t_j is the position of the least significant $\mathbf{0}$ in the binary representation of $j - 1$. □

Now it remains to put the three lemmas together and so construct a binary transitive automaton with a strictly exponential subset synchronization threshold.

Theorem 8. *It holds that* $\mathrm{CS}_n^{\mathcal{TR}\cap\mathcal{AL}_2} = 2^{\Omega(n)}$.

Proof. The series CS_n is non-decreasing, so it is enough to work with some infinitely many values of n. Let us take any $m \in \mathbb{N}$ and use it to build the automaton $A = (Q, X, \delta)$ and the subset S as described by Lemma 7. We apply Lemma 6 to get transitive $A' = (Q', X', \delta')$ and S' with

$$\begin{aligned} |Q'| &= 24m + 4\log m + 14 \\ |X'| &= 12 \\ \mathrm{CS}(A', S') &\geq 2^m \end{aligned}$$

and then apply Lemma 3 to get transitive $A'' = (Q'', X'', \delta'')$ and S'' with

$$\begin{aligned} |Q''| &= 288m + 48\log m + 168 \\ |X'| &= 2 \\ \mathrm{CS}(A'', S'') &\geq 2^m \end{aligned}$$

Denoting $n = 288m + 48\log m + 168$ we get that

$$\mathrm{CS}_n^{\mathcal{TR}\cap\mathcal{AL}_2} = \Omega\left(2^{\frac{n}{289}}\right).$$

□

Simpler variants of the constructions imply some more subtle results for less restricted classes:

Theorem 9. *It holds that*

1. $\mathrm{CS}_n = \Omega\left(2^{\frac{n}{2}}\right)$
2. $\mathrm{CS}_n^{\mathcal{TR}} = \Omega\left(2^{\frac{n}{4}}\right)$
3. $\mathrm{CS}_n^{\mathcal{AL}_4} = \Omega\left(2^{\frac{n}{7}}\right)$
4. $\mathrm{CS}_n^{\mathcal{AL}_2} = \Omega\left(2^{\frac{n}{25}}\right)$

A series witnessing the first claim arises from the proof of Lemma 7 if we just consider actual alphabet consisting of the letters $\{\omega, v_0, \ldots, v_{m-1}\}$ and realize the idea of Figure 6 so there remain only the states $D, \overline{D}$ and $(i, \mathbf{0}), (i, \mathbf{1})$ for each i. There is no more need to deal with a De Bruijn sequence. The construction presented in Lemma 7 results from an effort to make this simple variant binary with keeping the size of Q in $\mathcal{O}(m)$. A construction needed to prove the second claim depends on a careful use of swap congruences and appears in the extended version of this paper. The third claim follows directly from Lemma 7 and the last one we get if we then just apply Lemma 3.

3 Deciding about Synchronizability

It is well known that the decision about classical synchronizability of a given automaton (i.e. assuming $S = Q$) is a polynomial time task, even if we also require an explicit reset word on the output. A relatively simple algorithm could be traced back to [3] and since that time a lot of work has been done on various improvements. Besides decreasing the running time of the algorithm there is an effort to decrease the length of reset words produced [16, 21]. It has been proven that it is both NP-hard and coNP-hard to find a *shortest* reset word for given automaton (it is actually DP-complete [14]). Moreover, it remains NP-hard to bound the length of shortest reset words only from above by a given value [4] or approximate its length with a constant factor [2]. Such problems has been studied also with various additional requirements on the automaton, e.g. cyclicity, Eulerian property, commutativity and others, but in most cases also the restricted problem turns up to be hard, see [11, 24].

On the other hand, there have not been done much research in computational complexity of problems concerning synchronization of subsets, although they does not seem to have less chance to emerge in practice. Namely, the first natural problem in this direction is

SUBSYNITY
Input: n-state automaton $A = (Q, X, \delta)$, $S \subseteq Q$
Output: is there some $w \in X^{\star}$ such that $|\delta(S, w)| = 1$?

This problem, in contrast to the similar problem of classical synchronizability, is known to be PSPACE-complete. Note that such hardness is not a consequence of any lower bound of synchronization threshold, because an algorithm need not to produce an explicit reset word.

Theorem 10 ([13, 19])**.** SUBSYNITY *is a PSPACE-complete problem.*

The proofs of the theorem above make use of a result of Kozen [9], which establishes that it is PSPACE-complete to decide if given finite acceptors with a common alphabet accept any common word. This problem is polynomially reduced to SUBSYNITY using the idea of two sink states, which is used also in the automata with prime-length cycles and in Lemma 7. Is it possible to avoid the non-transitivity here? We have proved that the subset synchronization threshold may be exponential even in automata from $\mathcal{AL}_2 \cap \mathcal{TR}$, but this does not imply that there is no trick for tractable decision about their synchronizability. However, the methods we used are general enough to reduce SUBSYNITY to the restricted version:

Theorem 11. SUBSYNITY RESTRICTED TO BINARY TRANSITIVE AUTOMATA *is a PSPACE-complete problem.*

Proof. There is a polynomial reduction from the general problem SUBSYNITY: Perform the construction from Lemma 6 and then the one from Lemma 3. □

Though the results of this paper does not sound very optimistically, there are still many interesting and practical restrictions which could hypothetically make our decision problem tractable or at least decrease the subset synchronization threshold, preferably to a polynomial. Such restrictions, which all have been already studied in terms of classical synchronization, concern monotonic and aperiodic automata, cyclic and one-cluster automata, Eulerian automata and others.

References

[1] Marie-Pierre Béal, Mikhail V. Berlinkov, and Dominique Perrin. A quadratic upper bound on the size of a synchronizing word in one-cluster automata. *Int. J. Found. Comput. Sci.*, 22(2):277–288, 2011. doi:10.1142/S0129054111008039.

[2] Mikhail V. Berlinkov. Approximating the minimum length of synchronizing words is hard. *Theory of Computing Systems*, 54(2):211–223, 2014. doi:10.1007/s00224-013-9511-y.

[3] Ján Černý. Poznámka k homogénnym experimentom s konečnými automatmi. *Matematicko-fyzikálny časopis*, 14(3):208–216, 1964.

[4] David Eppstein. Reset sequences for monotonic automata. *SIAM J. Comput.*, 19(3):500–510, 1990. doi:10.1137/0219033.

[5] Camille Flye Sainte-Marie. Solution to question nr. 48. *L'intermédiaire des Mathématicians*, 1:107–110, 1894.

[6] Mariusz Grech and Andrzej Kisielewicz. The Černý conjecture for automata respecting intervals of a directed graph. *Discrete Mathematics & Theoretical Computer Science*, 15(3):61–72, 2013.

[7] Harald A. Helfgott and Ákos Seress. On the diameter of permutation groups. *Annals of Mathematics, to appear*.

[8] Jakub Kowalski and Marek Szykula. The Černý conjecture for small automata: experimental report. *CoRR*, abs/1301.2092, 2013.

[9] D. Kozen. Lower bounds for natural proof systems. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 254–266, 1977. doi:10.1109/SFCS.1977.16.

[10] D. Lee and Mihalis Yannakakis. Testing finite-state machines: state identification and verification. *Computers, IEEE Transactions on*, 43(3):306–320, 1994. doi:10.1109/12.272431.

[11] Pavel Martyugin. Complexity of problems concerning reset words for cyclic and eulerian automata. In Béatrice Bouchou-Markhoff, Pascal Caron, Jean-Marc Champarnaud, and Denis Maurel, editors, *Implementation and Application of Automata*, volume 6807 of *Lecture Notes in Computer Science*, pages 238–249. Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-22256-6_22.

[12] Pavel V. Martyugin. Careful synchronization of partial automata with restricted alphabets. In Andrei A. Bulatov and Arseny M. Shur, editors, *Computer Science - Theory and Applications*, volume 7913 of *Lecture Notes in Computer Science*, pages 76–87. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-38536-0_7.

[13] B. K. Natarajan. An algorithmic approach to the automated design of parts orienters. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, SFCS '86, pages 132–142, Washington, DC, USA, 1986. IEEE Computer Society. doi:10.1109/SFCS.1986.5.

[14] Jörg Olschewski and Michael Ummels. The complexity of finding reset words in finite automata. In *Proceedings of the 35th international conference on Mathematical foundations of computer science*, MFCS'10, pages 568–579, Berlin, Heidelberg, 2010. Springer-Verlag. doi:10.1007/978-3-642-15155-2_50.

[15] Jean-Eric Pin. On two combinatorial problems arising from automata theory. *Annals of Discrete Mathematics*, 17:535–548, 1983.

[16] Adam Roman. Synchronizing finite automata with short reset words. *Applied Mathematics and Computation*, 209(1):125–136, 2009. doi:10.1016/j.amc.2008.06.019.

[17] Arto Salomaa. Composition sequences for functions over a finite domain. *Theoret. Comput. Sci.*, 292:263–281, 2000. doi:10.1016/S0304-3975(01)00227-4.

[18] Arto Salomaa. A half-century of automata theory. chapter Compositions over a Finite Domain: From Completeness to Synchronizable Automata, pages 131–143. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2001. doi:10.1142/9789812810168_0007.

[19] Sven Sandberg. Homing and synchronizing sequences. In Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, and Alexander Pretschner, editors, *Model-Based Testing of Reactive Systems*, volume 3472 of *Lecture Notes in Computer Science*, pages 5–33. Springer Berlin Heidelberg, 2005. `doi:10.1007/11498490_2`.

[20] Benjamin Steinberg. The Černý conjecture for one-cluster automata with prime length cycle. *Theoret. Comput. Sci.*, 412(39):5487 – 5491, 2011. `doi:10.1016/j.tcs.2011.06.012`.

[21] A. N. Trahtman. An efficient algorithm finds noticeable trends and examples concerning the Cerny conjecture. In *MFCS'06*, pages 789–800, 2006. `doi:10.1007/11821069_68`.

[22] A. N. Trahtman. Modifying the upper bound on the length of minimal synchronizing word. In *FCT*, pages 173–180, 2011. `doi:10.1007/978-3-642-22953-4_15`.

[23] MikhailV. Volkov. Synchronizing automata and the Cerný conjecture. In Carlos Martín-Vide, Friedrich Otto, and Henning Fernau, editors, *Language and Automata Theory and Applications*, volume 5196 of *Lecture Notes in Computer Science*, pages 11–27. Springer Berlin Heidelberg, 2008. `doi:10.1007/978-3-540-88282-4_4`.

[24] Vojtìch Vorel. Complexity of a problem concerning reset words for eulerian binary automata. In Adrian-Horia Dediu, Carlos Martín-Vide, José-Luis Sierra-Rodríguez, and Bianca Truthe, editors, *Language and Automata Theory and Applications*, volume 8370 of *Lecture Notes in Computer Science*, pages 576–587. Springer International Publishing, 2014. `doi:10.1007/978-3-319-04921-2_47`.